ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/МЭК 26300— 2010

Информационная технология

ФОРМАТ OPEN DOCUMENT ДЛЯ ОФИСНЫХ ПРИЛОЖЕНИЙ (OpenDocument) v1.0

ISO/IEC 26300:2006
Information technology — Open Document Format for Office Applications (OpenDocument) v1.0
(IDT)

Издание официальное

Москва
Стандартинформ
2011

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации — ГОСТ Р 1.0—2004 «Стандартизация в Российской Федерации. Основные положения».

Сведения о стандарте

1. ПОДГОТОВЛЕН Евро-Азиатской ассоциацией производителей товаров и услуг в области безопасности (Ассоциация ЕВРААС) на основе аутентичного перевода международного стандарта, указанного в пункте 4, который выполнен ООО «Инфра-Ресурс».

2. ВНЕСЕН Техническим комитетом по стандартизации ТК 22 «Информационные технологии».

3. УТВЕРЖДЕН и ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 21 декабря № 800-ст.

4. Настоящий стандарт идентичен международному стандарту ИСО/МЭК 26300:2006 «Информационная технология. Формат Open Document для офисных приложений (OpenDocument) v1.0» (ISO/IEC 26300:2006 «Information technology — Open Document Format for Office Applications (OpenDocument) v1.0»).

5. ISO/IEC 26300 был подготовлен техническим комитетом OpenDocument организации OASIS как «OASIS Open Document for Office Applications (OpenDocument) v1.0 (second edition)» и выдвинут по процедуре ПДС — публично доступная спецификация (PAS — Publicly Available Specification) совместным техническим комитетом ИСО/МЭК СТК 1 «Информационные технологии» (ISO/IEC JTC 1, Information technology) параллельно с его одобрением органами стандартизации ИСО и МЭК. Содержания ISO/IEC 26300 и второй редакции OASIS OpenDocument v1.0 идентичны.

6. ВВЕДЕН ВПЕРВЫЕ.

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемых информационных указателях «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет.

Содержание

1 Введение....I
1.1 Структура и цель стандарта....I
1.2 Нотация....3
1.3 Пространства имен....3
1.4 Схема Relax-NG....5
1.5 Обработка документа и его соответствие спецификации....6
1.6 Обработка пробельных символов и символов конца строки....7
1.7 Типы MIME и расширения имен файлов....8
2 Структура документа....10
2.1 Корни документа....10
2.1.1 Модели содержимого корневых элементов документа....12
2.1.2 Атрибуты корня документа....13
2.2 Метаданные документа....15
2.2.1 Предопределенные и специфические метаданные....15
2.2.2 Образец метаданных....16
2.3 Элемент основной части документа и типы документов....16
2.3.1 Текстовые документы....17
2.3.2 Графические документы....19
2.3.3 Документы презентаций....20
2.3.4 Документы электронных таблиц....21
2.3.5 Документы диаграмм....22
2.3.6 Документы растровой графики....23
2.4 Параметры настройки приложений....24
2.4.1 Последовательность параметров настройки....24
2.4.2 Основные параметры настройки....25
2.4.3 Индексный доступ к последовательностям....26
2.4.4 Запись в ассоциированный массив....26
2.4.5 Именованный доступ к последовательностям....27
2.4.6 Параметры настройки позиции курсора....27
2.5 Скрипты....28
2.5.1 Скрипт....29
2.6 Объявления начертания шрифта....29
2.7 Стили....30

2.7.1 Расположение стилей....31
2.8 Стили и макеты страницы....33
3 Элементы метаданных....35
3.1 Предопределенные элементы метаданных....35
3.1.1 Генератор....35
3.1.2 Заголовок....36
3.1.3 Описание....36
3.1.4 Тема....36
3.1.5 Ключевые слова....36
3.1.6 Автор первоначального документа....36
3.1.7 Автор....37
3.1.8 Кем распечатан документ....37
3.1.9 Дата и время создания документа....37
3.1.10 Дата и время последней модификации документа....38
3.1.11 Дата и время печати документа....38
3.1.12 Шаблон документа....38
3.1.13 Автоматическая перезагрузка....39
3.1.14 Поведение гиперссылок....40
3.1.15 Язык....42
3.1.16 Циклы редактирования....42
3.1.17 Время редактирования....42
3.1.18 Статистика документа....43
3.2 Метаданные, определяемые пользователем....45
3.3 Специфические метаданные....46
4 Текстовое содержимое....47
4.1 Заголовки, абзацы и основная текстовая структура....47
4.1.1 Заголовки....47
4.1.2 Абзацы....49
4.1.3 Общие атрибуты элементов абзаца....49
4.2 Последовательности страниц....50
4.2.1 Страница....51
4.3 Списки....52
4.3.1 Блок списка....52
4.3.2 Пункт списка....54
4.3.3 Заголовок списка....56
4.3.4 Нумерованные абзацы....56
4.4 Текстовые разделы....57

4.4.1 Атрибуты раздела....58
4.4.2 Источник раздела....60
4.4.3 Источник DDE....62
4.5 Графическое наполнение, привязанное к странице....62
4.6 Отслеживание изменений....62
4.6.1 Отслеженные изменения....63
4.6.2 Измененные области....63
4.6.3 Вставка....64
4.6.4 Удаление....64
4.6.5 Изменение формата....67
4.6.6 Информация об изменениях....67
4.6.7 Метки изменения....67
4.7 Текстовые объявления....68
5 Содержимое элементов абзаца....70
5.1 Основное текстовое содержимое....70
5.1.1 Пробельные символы....70
5.1.2 Мягкие переносы, переносы и неразрывные пробелы....73
5.1.3 Текст с присвоенными атрибутами....73
5.1.4 Гиперссылки....74
5.2 Закладки и перекрестные ссылки....77
5.2.1 Закладки....77
5.2.2 Перекрестные ссылки....78
5.3 Сноски....79
5.3.1 Элемент сноски....80
5.4 Транскрипция....82
5.5 Аннотации к тексту....83
5.6 Индексные метки....83
5.7 Отслеживание изменений и метки изменений....83
5.8 Внедренные графические объекты и текстовые блоки....83
6 Текстовые поля....85
6.1 Общие характеристики элементов поля....85
6.2 Поля документа....87
6.2.1 Поля даты....87
6.2.2 Поля времени....89
6.2.3 Поля номера страницы....91
6.2.4 Текст, указывающий на продолжение страницы....93
6.2.5 Поля отправителя....94

6.2.6 Поля автора....97
6.2.7 Поля главы....98
6.2.8 Поля имени файла....99
6.2.9 Поля имени шаблона документа....100
6.2.10 Поля имен листов....101
6.3 Поля-переменные....101
6.3.1 Объявление простых переменных....103
6.3.2 Задание значения простой переменной....103
6.3.3 Отображение простых переменных....105
6.3.4 Поля ввода простой переменной....106
6.3.5 Определение переменных пользователя....107
6.3.6 Отображение переменных пользователя....108
6.3.7 Поля ввода переменной пользователя....109
6.3.8 Объявление переменных последовательности....110
6.3.9 Использование полей последовательности....112
6.3.10 Поля выражений....114
6.3.11 Поля ввода текста....115
6.4 Поля метаданных....115
6.4.1 Автор первоначального документа....116
6.4.2 Дата создания документа....116
6.4.3 Время создания документа....116
6.4.4 Описание документа....117
6.4.5 Информация о документе, определяемая пользователем....117
6.4.6 Время печати....117
6.4.7 Дата печати....118
6.4.8 Кем распечатан документ....118
6.4.9 Заголовок документа....118
6.4.10 Тема документа....119
6.4.11 Ключевые слова документа....119
6.4.12 Номер редакции документа....119
6.4.13 Продолжительность редактирования документа....119
6.4.14 Время изменения документа....120
6.4.15 Дата изменения документа....120
6.4.16 Автор изменений документа....121
6.4.17 Поля статистики документа....121
6.5 Поля базы данных....121
6.5.1 Источник данных поля базы данных....123

6.5.2 Отображение содержимого базы данных....124
6.5.3 Выборка следующей строки базы данных....125
6.5.4 Выбор номера строки....127
6.5.5 Отображение номера строки....128
6.5.6 Отображение текущей базы данных и таблицы....128
6.6 Прочие поля....129
6.6.1 Поля переменных страницы....129
6.6.2 Поля для заполнения....130
6.6.3 Условные текстовые поля....132
6.6.4 Скрытое текстовое поле....133
6.6.5 Поля ссылок....135
6.6.6 Поля скриптов....138
6.6.7 Поля макросов....139
6.6.8 Скрытые поля абзаца....139
6.6.9 Поля DDE-соединения....141
6.6.10 Поля измерений....141
6.6.11 Поле формулы таблицы....142
6.7 Общие атрибуты полей....143
6.7.1 Переменные значения и типы значения....143
6.7.2 Фиксация....145
6.7.3 Имя переменной....146
6.7.4 Описание 146
6.7.5 Отображение....147
6.7.6 Формула....148
6.7.7 Стиль форматирования....149
6.7.8 Стиль форматирования чисел....149
7 Текстовые индексы....151
7.1 Индексные метки....151
7.1.1 Индексные метки оглавления....152
7.1.2 Индексные метки, определяемые пользователем....153
7.1.3 Алфавитные индексные метки....154
7.1.4 Библиографические индексные метки....157
7.2 Структура индексов....158
7.2.1 Источник индекса....158
7.2.2 Тело индекса....159
7.2.3 Заголовок индекса....159
7.3 Оглавление....160

7.3.1 Источник оглавления....160
7.3.2 Шаблоны записей оглавления....163
7.4 Индексы иллюстраций....164
7.4.1 Индексы источников иллюстраций....165
7.4.2 Шаблон индексных записей иллюстраций....167
7.5 Индексы таблиц....168
7.5.1 Источник индекса таблиц....168
7.5.2 Шаблон индексных записей таблицы....168
7.6 Индексы объектов....169
7.6.1 Источник индекса объектов....169
7.6.2 Шаблон индексных записей объекта....170
7.7 Индексы, определяемые пользователем....171
7.7.1 Источник индекса, определяемого пользователем....171
7.7.2 Шаблоны индексных записей, определяемых пользователем....174
7.8 Алфавитные индексы....175
7.8.1 Источник алфавитного индекса....175
7.8.2 Файл с автоматическими метками....179
7.8.3 Шаблон алфавитных индексных записей....180
7.9 Библиография....181
7.9.1 Источник библиографического индекса....181
7.9.2 Шаблон библиографических записей....181
7.10 Стили источника индекса....182
7.10.1 Стиль источника индекса....183
7.11 Шаблон заголовков индекса....183
7.12 Записи индексного шаблона....183
7.12.1 Информация о главе....184
7.12.2 Текст записи....184
7.12.3 Номер страницы....185
7.12.4 Фиксированная строка....185
7.12.5 Библиографическая информация....185
7.12.6 Табулостоп....187
7.12.7 Начало и конец гиперссылки....188
7.12.8 Пример конфигурации индексной записи....189
8 Таблицы....191
8.1 Общая модель таблиц....191
8.1.1 Элемент таблицы....191
8.1.2 Строка таблицы....195

8.1.3 Ячейки таблицы....198
8.2 Расширенная модель таблиц....204
8.2.1 Описание столбца....204
8.2.2 Заголовки столбцов....206
8.2.3 Группы столбцов....207
8.2.4 Заголовки строк....208
8.2.5 Группы строк....209
8.2.6 Подтаблицы....209
8.3 Расширенные таблицы....213
8.3.1 Ссылки на ячейки таблицы....213
8.3.2 Связанные таблицы....215
8.3.3 Таблицы сценариев....217
8.3.4 Фигуры....221
8.4 Расширенные ячейки таблицы....221
8.4.1 Связанные ячейки таблицы....221
8.4.2 Аннотация ячейки....222
8.4.3 Слежение....222
8.4.4 Операция слежения....223
8.4.5 Диапазон выделения цветом....224
8.5 Содержимое документов электронных таблиц....226
8.5.1 Защита документа....226
8.5.2 Настройки вычислений....226
8.5.3 Проверки содержимого ячеек таблицы....230
8.5.4 Диапазоны надписей....235
8.5.5 Именованные выражения....237
8.6 Диапазоны баз данных....239
8.6.1 Диапазон базы данных....240
8.6.2 SQL источника базы данных....243
8.6.3 Таблица источника базы данных....244
8.6.4 Запрос источника базы данных....245
8.6.5 Сортировка....246
8.6.6 Поле сортировки....248
8.6.7 Правила промежуточных сумм....249
8.6.8 Отсортированные группы промежуточных сумм....250
8.6.9 Правило промежуточной суммы....251
8.6.10 Поле промежуточной суммы....252
8.7 Фильтры....253

8.7.1 Фильтр таблицы....253
8.7.2 Фильтр And (И)....255
8.7.3 Фильтр Or (ИЛИ)....256
8.7.4 Условие фильтра....256
8.8 Сводные таблицы....258
8.8.1 Сводная таблица....259
8.8.2 Источник-диапазон ячеек....265
8.8.3 Источник-служба....266
8.8.4 Поле сводной таблицы....267
8.8.5 Уровень сводной таблицы....270
8.8.6 Промежуточные суммы сводной таблицы....271
8.8.7 Промежуточная сумма сводной таблицы....272
8.8.8 Элементы сводной таблицы....272
8.8.9 Элемент сводной таблицы....273
8.8.10 Информация об отображении сводной таблицы....274
8.8.11 Информация о сортировке сводной таблицы....275
8.8.12 Информация о разметке сводной таблицы....276
8.8.13 Ссылки полей сводной таблицы....278
8.8.14 Группы в сводных таблицах....280
8.8.15 Группа сводной таблицы....283
8.8.16 Элемент группы сводной таблицы....284
8.9 Консолидация....284
8.10 DDE-ссылки....286
8.11 Отслеживание изменений в электронных таблицах....286
8.11.1 Отслеживаемые изменения....287
8.11.2 Вставка....287
8.11.3 Зависимости....289
8.11.4 Зависимость....289
8.11.5 Удаления....290
8.11.6 Удаление содержимого ячейки....290
8.11.7 Удаление изменений....290
8.11.8 Удаление....291
8.11.9 Вырезки....293
8.11.10 Вырезка вставки....293
8.11.11 Вырезка перемещения....294
8.11.12 Перемещение....295
8.11.13 Адрес целевого диапазона, адрес диапазона источника....295

8.11.14 Ячейка отслеживания изменений....297
8.11.15 Изменение содержимого ячейки....299
8.11.16 Адрес ячейки....299
8.11.17 Предыдущее содержимое ячейки....300
8.11.18 Общие атрибуты отслеживания изменений....300
9 Графическое содержимое....302
9.1 Расширенные свойства страниц для графических приложений....302
9.1.1 Мастер-страница тезисов....302
9.1.2 Множества слоев....304
9.1.3 Слой....304
9.1.4 Графические страницы....305
9.1.5 Заметки презентации....309
9.2 Векторные фигуры....310
9.2.1 Прямоугольник....310
9.2.2 Линия....311
9.2.3 Ломаная....312
9.2.4 Многоугольник....313
9.2.5 Правильный многоугольник....314
9.2.6 Траектория....315
9.2.7 Окружность....317
9.2.8 Эллипс....319
9.2.9 Соединительный элемент....320
9.2.10 Подпись....324
9.2.11 Измерение....325
9.2.12 Элемент управления....326
9.2.13 Уменьшенное изображение страницы....327
9.2.14 Группировка....328
9.2.15 Общие атрибуты векторных фигур....328
9.2.16 Общие атрибуты векторных фигур для текстовых документов и документов электронных таблиц....333
9.2.17 Общее содержимое векторных фигур....336
9.2.18 Общие группы атрибутов векторных фигур....337
9.2.19 Точки привязки....337
9.2.20 Датчики событий....339
9.3 Фреймы....339
9.3.1 Текстовый блок....343
9.3.2 Изображение....346

9.3.3 Объекты....348
9.3.4 Апплеты....350
9.3.5 Плагины....352
9.3.6 Параметры....353
9.3.7 Плавающие фреймы....354
9.3.8 Контур....354
9.3.9 Альтернативный текст....356
9.3.10 Гиперссылки....356
9.3.11 Клиентские сенсорные изображения....358
9.4 Трехмерные фигуры....364
9.4.1 Сцена....364
9.4.2 Источник света....368
9.4.3 Куб....369
9.4.4 Сфера....370
9.4.5 Выдавливание....371
9.4.6 Вращение....371
9.5 Специальные фигуры....372
9.5.1 Расширенная геометрия....373
9.5.2 Расширенная геометрия: атрибуты выдавливания....377
9.5.3 Расширенная геометрия: атрибуты траектории....384
9.5.4 Расширенная геометрия: атрибуты траектории текста....389
9.5.5 Расширенная геометрия: уравнение....391
9.5.6 Расширенная геометрия: атрибуты меток-манипуляторов....393
9.6 Фигуры презентации....398
9.6.1 Общие атрибуты фигур презентации....398
9.7 Анимация презентаций....401
9.7.1 Звук....401
9.7.2 Показ фигуры....403
9.7.3 Показ текста....407
9.7.4 Скрытие фигуры....407
9.7.5 Скрытие текста....408
9.7.6 Затемнение....408
9.7.7 Воспроизведение....409
9.7.8 Группы эффектов....409
9.8 Анимация SMIL-презентаций....410
9.8.1 Рекомендуемое применение SMIL....410
9.8.2 Значения атрибутов SMIL-анимации, зависящие от документа....412

9.8.3 Атрибуты анимации SMIL-презентации....415
9.9 События презентации....418
9.10 Текстовые поля презентаций....422
9.10.1 Поле верхнего колонтитула....422
9.10.2 Поле нижнего колонтитула....422
9.10.3 Поле даты и времени....423
9.11 Содержимое документов презентаций....423
9.11.1 Объявления презентации....423
9.11.2 Объявление поля верхнего колонтитула....423
9.11.3 Объявление поля нижнего колонтитула....424
9.11.4 Объявление поля даты и времени....424
9.11.5 Параметры презентации....425
9.11.6 Описание демонстрации....430
10 Содержимое диаграмм....432
10.1 Введение в документы диаграмм....432
10.2 Диаграмма....433
10.3 Заголовок, подзаголовок и нижний колонтитул....437
10.3.1 Заголовок....437
10.3.2 Подзаголовок....438
10.3.3 Нижний колонтитул....439
10.4 Легенда....439
10.5 Область построения....442
10.5.1 Трехмерная область построения....444
10.6 Стена....444
10.7 Основание диаграммы....445
10.8 Оси координат....446
10.8.1 Координатная сетка....448
10.9 Ряды данных....449
10.9.1 Домен....451
10.10 Категории....452
10.11 Точка данных....452
10.12 Среднее значение....453
10.13 Индикатор ошибок....454
10.14 Кривые регрессии....454
10.14.1 Метки биржевой диаграммы....455
11 Содержимое форм....457
11.1 Форма....458

11.1.1 Действие....460
11.1.2 Целевой фрейм....460
11.1.3 Метод....461
11.1.4 Тип кодировки....461
11.1.5 Разрешение удаления....461
11.1.6 Разрешение вставок....462
11.1.7 Разрешение обновлений....462
11.1.8 Применение фильтра....462
11.1.9 Тип команды....462
11.1.10 Команда....463
11.1.11 Источник данных....463
11.1.12 Мастер-поля....464
11.1.13 Поля деталей....464
11.1.14 Избежать обработку команды....464
11.1.15 Фильтр....465
11.1.16 Игнорирование результата....465
11.1.17 Режим навигации....465
11.1.18 Порядок....466
11.1.19 Цикл табуляции....466
11.1.20 Ресурс соединения....467
11.2 Модель XForms....467
11.2.1 Модель XForms (импорт)....468
11.3 Элементы управления....468
11.3.1 Текст....470
11.3.2 Текстовая область....471
11.3.3 Пароль....472
11.3.4 Файл....473
11.3.5 Форматированный текст....473
11.3.6 Число....475
11.3.7 Дата и время....476
11.3.8 Фиксированный текст....478
11.3.9 Поле со списком....479
11.3.10 Список....481
11.3.11 Кнопка....483
11.3.12 Изображение....485
11.3.13 Флажок....486
11.3.14 Радиокнопка....488

11.3.15 Фрейм....489
11.3.16 Фрейм изображения....489
11.3.17 Скрытый элемент....490
11.3.18 Сетка....490
11.3.19 Диапазон значений....492
11.3.20 Базовый элемент управления....494
11.4 Общие атрибуты элементов управления и форм....495
11.4.1 Имя....495
11.4.2 Реализация элемента управления....495
11.4.3 Привязка к XForms....496
11.5 Общие атрибуты элементов управления....496
11.5.1 Тип кнопки....496
11.5.2 Идентификатор элемента управления....497
11.5.3 Текущий выбор....498
11.5.4 Значение и текущее значение....498
11.5.5 Запрещен....500
11.5.6 Выпадающий список....501
11.5.7 Для....501
11.5.8 Изображение....501
11.5.9 Метка....502
11.5.10 Максимальная длина....502
11.5.11 Возможность печати....503
11.5.12 Только для чтения....504
11.5.13 Выбранный....504
11.5.14 Размер....505
11.5.15 Индекс табуляции....505
11.5.16 Остановка табуляции....506
11.5.17 Целевой фрейм....507
11.5.18 Целевой адрес....507
11.5.19 Заголовок....508
11.5.20 Визуальный эффект....509
11.5.21 Относительное расположение изображения....509
11.5.22 Атрибуты связи с базой данных....510
11.6 События....513
11.6.1 События, имеющие эквивалентные типы событий в HTML....513
11.6.2 Типы событий....514
11.7 Свойства....516

11.7.1 Набор свойств....516
11.7.2 Свойство....517
11.7.3 Список свойств....517
12 Общее содержимое....521
12.1 Аннотация....521
12.1.1 Автор....522
12.1.2 Дата и время создания....522
12.1.3 Строка даты и времени создания....522
12.2 Формат чисел....523
12.2.1 Префикс и суффикс....523
12.2.2 Спецификация формата....523
12.2.3 Синхронизация букв в форматах номера....524
12.3 Метаданные отслеживания изменений....525
12.4 Таблицы датчиков событий....525
12.4.1 Датчик событий....526
12.5 Математическое содержимое....528
12.6 DDE-соединения....528
12.6.1 Контейнер описания DDE-соединения....529
12.6.2 Описание DDE-соединения для текстовых полей....529
12.6.3 Объявление DDE-соединений для таблиц....531
13 SMIL-анимации....533
13.1 Основные элементы анимации....533
13.1.1 Анимировать....533
13.1.2 Множество....533
13.1.3 Анимированное движение....534
13.1.4 Анимация цвета....535
13.1.5 Анимация преобразования....536
13.1.6 Фильтр перехода....537
13.2 Атрибуты анимационной модели....538
13.3 Общие атрибуты анимации....539
13.3.1 Целевые атрибуты анимации....539
13.3.2 Функциональные атрибуты анимации....540
13.4 Распределение времени анимации....543
13.4.1 Атрибуты распределения времени анимации....543
13.4.2 Параллельные анимации....547
13.4.3 Последовательные анимации....548
13.4.4 Итеративные анимации....548

13.5 Элементы мультимедиа....550
13.5.1 Звук....550
13.6 Специальные элементы....550
13.6.1 Команда....550
14 Стили....552
14.1 Элемент стиля....553
14.1.1 Соотнесение стилей....558
14.2 Стили по умолчанию....560
14.3 Разметка страницы....561
14.3.1 Стили колонтитулов....562
14.4 Мастер-страницы....562
14.4.1 Колонтитулы....565
14.4.2 Заметки к презентации....568
14.5 Шаблоны таблиц....570
14.5.1 Стили строк и столбцов....571
14.6 Объявление начертания шрифта....573
14.6.1 Дескрипторы шрифта CSS2/SVG....574
14.6.2 Имя....578
14.6.3 Начертание....578
14.6.4 Общее семейство шрифта....578
14.6.5 Шаг шрифта....578
14.6.6 Набор символов шрифта....579
14.7 Стили данных....579
14.7.1 Числовой стиль....579
14.7.2 Валютный стиль....584
14.7.3 Процентный стиль....586
14.7.4 Стиль даты....587
14.7.5 Стиль времени....594
14.7.6 Логический стиль....599
14.7.7 Текстовый стиль....599
14.7.8 Общие элементы стиля данных....601
14.7.9 Общие атрибуты стиля данных....602
14.7.10 Транслитерация....605
14.7.11 Атрибуты общих элементов стиля данных....607
14.8 Стили текста....610
14.8.1 Стили текста....610
14.8.2 Стили абзацев....610

14.8.3 Стили разделов....610
14.8.4 Стиль транскрипции....611
14.9 Расширенные стили текста....611
14.9.1 Конфигурация нумерации строк....611
14.9.2 Элемент конфигурации заметок....615
14.9.3 Конфигурация библиографии....619
14.10 Стили списков....622
14.10.1 Общий атрибут стилей уровня списка....623
14.10.2 Стиль нумерованного списка....624
14.10.3 Стиль маркированного списка....625
14.10.4 Стиль списка, отмечаемого изображениями....627
14.10.5 Пример стиля уровня списка....628
14.11 Стили структур....629
14.11.1 Стиль уровня структуры....629
14.12 Стили таблиц....631
14.12.1 Стили таблиц....631
14.12.2 Стили столбцов таблицы....631
14.12.3 Стили строк таблицы....632
14.12.4 Стили ячеек таблицы....632
14.13 Графические стили....633
14.13.1 Графические стили и стили презентации....633
14.13.2 Стили графических страниц....634
14.14 Элементы расширенных графических стилей....634
14.14.1 Градиент....635
14.14.2 SVG-градиенты....638
14.14.3 Штриховка....642
14.14.4 Заполнение изображением....644
14.14.5 Непрозрачность градиента....645
14.14.6 Маркер....646
14.14.7 Штрих контурной линии....647
14.15 Разметка страницы презентации....649
14.15.1 Поле для заполнения в презентации....650
14.16 Стили диаграмм....651
15 Свойства форматирования....652
15.1 Простые и составные свойства форматирования....653
15.1.1 Простые свойства форматирования....653
15.1.2 Составные свойства форматирования....653

15.1.3 Правила обработки свойств форматирования....654
15.2 Свойства форматирования для разметки страницы....654
15.2.1 Размер страницы....655
15.2.2 Формат номера страницы....655
15.2.3 Лоток для бумаги....656
15.2.4 Ориентация печати....656
15.2.5 Поля....656
15.2.6 Обрамление....657
15.2.7 Толщина линии обрамления....657
15.2.8 Заполнение....657
15.2.9 Тень....658
15.2.10 Фон....658
15.2.11 Колонки....658
15.2.12 Приводка....658
15.2.13 Печать....659
15.2.14 Порядок страниц при печати....660
15.2.15 Нумерация первой страницы....660
15.2.16 Масштаб....660
15.2.17 Выравнивание таблицы....661
15.2.18 Максимальная высота сноски....661
15.2.19 Режим записи....662
15.2.20 Разделитель сноски....662
15.2.21 Сеточная разметка....663
15.2.22 Базовая высота сеточной разметки....664
15.2.23 Высота транскрипции сеточной разметки....664
15.2.24 Строки сеточной разметки....665
15.2.25 Цвет линий сеточной разметки....665
15.2.26 Положение транскрипции сеточной разметки....665
15.2.27 Печать сеточной разметки....665
15.2.28 Отображение разметки сетки....666
15.3 Свойства форматирования верхнего и нижнего колонтитулов....666
15.3.1 Фиксированная и минимальная высота....667
15.3.2 Поля....667
15.3.3 Обрамление....667
15.3.4 Толщина линии обрамления....668
15.3.5 Заполнение....668
15.3.6 Фон....668

15.3.7 Тень....669
15.3.8 Динамический интервал....669
15.4 Свойства форматирования текста....669
15.4.1 Вид шрифта....670
15.4.2 Преобразования текста....670
15.4.3 Цвет....671
15.4.4 Цвет шрифта окна....671
15.4.5 Структура текста....671
15.4.6 Тип линии зачеркивания....672
15.4.7 Стиль линии зачеркивания....672
15.4.8 Протяженность линии зачеркивания....672
15.4.9 Цвет зачеркивания....673
15.4.10 Текст зачеркивания....673
15.4.11 Стиль текста зачеркивания....674
15.4.12 Позиция текста....674
15.4.13 Название шрифта....675
15.4.14 Семейство шрифтов....676
15.4.15 Группа семейства шрифтов....676
15.4.16 Стиль шрифта....677
15.4.17 Шаг шрифта....678
15.4.18 Набор символов шрифта....679
15.4.19 Размер шрифта....679
15.4.20 Масштаб шрифта....680
15.4.21 Тип скрипта....681
15.4.22 Межбуквенный интервал....682
15.4.23 Язык....682
15.4.24 Страна....683
15.4.25 Стиль шрифта....683
15.4.26 Рельефный шрифт....684
15.4.27 Тень шрифта....684
15.4.28 Тип подчеркивания....685
15.4.29 Стиль подчеркивания....685
15.4.30 Протяженность подчеркивания....686
15.4.31 Цвет подчеркивания....686
15.4.32 Насыщенность шрифта....687
15.4.33 Режим подчеркивания слов....687
15.4.34 Режим зачеркивания слов....688

15.4.35 Кернинг букв 688
15.4.36 Мигание текста....688
15.4.37 Цвет фона текста....689
15.4.38 Объединение текста....689
15.4.39 Начальный и конечный символы объединения текста 690
15.4.40 Выделение текста....690
15.4.41 Масштаб текста....691
15.4.42 Угол вращения текста....691
15.4.43 Масштабирование при вращении....691
15.4.44 Расстановка переносов....692
15.4.45 Количество символов до переноса....692
15.4.46 Количество переносимых символов....693
15.4.47 Скрытый или условный текст....693
15.5 Свойства форматирования абзаца....694
15.5.1 Фиксированная высота строки....694
15.5.2 Минимальная высота строки....695
15.5.3 Межстрочный интервал....695
15.5.4 Межстрочный интервал, независимый от шрифта 695
15.5.5 Выравнивание текста....696
15.5.6 Выравнивание текста в последней строке....696
15.5.7 Выравнивание отдельного слова....697
15.5.8 Не разрывать....697
15.5.9 Висячие строки в начале абзаца....697
15.5.10 Висячие строки в конце абзаца....698
15.5.11 Табулостопы....698
15.5.12 Дистанция табулостопов....702
15.5.13 Удержание переноса....702
15.5.14 Максимальное число последовательных переносов....703
15.5.15 Буквица....703
15.5.16 Приводка....705
15.5.17 Левое и правое поля....705
15.5.18 Отступ текста....706
15.5.19 Автоматический отступ текста....707
15.5.20 Верхнее и нижнее поля....707
15.5.21 Поля....708
15.5.22 Разрыв до и разрыв после....708
15.5.23 Фоновый цвет абзаца....709

15.5.24 Фоновое изображение абзаца 709
15.5.25 Обрамление....712
15.5.26 Толщина линии обрамления....713
15.5.27 Заполнение 714
15.5.28 Тень....715
15.5.29 Вместе со следующим....715
15.5.30 Нумерация строк....716
15.5.31 Начальное значение нумерации строк....716
15.5.32 Автоматическое разделение текста....716
15.5.33 Перенос знаков препинания на новую строку....717
15.5.34 Разрыв строки....717
15.5.35 Вертикальное выравнивание....717
15.5.36 Режим записи....718
15.5.37 Режим автоматической записи....719
15.5.38 Привязка к разметке....719
15.5.39 Номер страницы....719
15.5.40 Прозрачность фона....720
15.6 Свойства форматирования транскрипции....720
15.6.1 Позиция транскрипции....720
15.6.2 Выравнивание транскрипции....721
15.7 Свойства форматирования разделов....721
15.7.1 Фон раздела....721
15.7.2 Поля....722
15.7.3 Колонки....722
15.7.4 Спецификация колонок....723
15.7.5 Разделитель колонок....725
15.7.6 Защита....727
15.7.7 Несбалансированные колонки текста....727
15.7.8 Режим записи....727
15.7.9 Настройка примечаний....728
15.8 Свойства форматирования таблиц....728
15.8.1 Ширина таблицы....728
15.8.2 Выравнивание таблицы....729
15.8.3 Левое и правое поля таблицы....730
15.8.4 Верхнее и нижнее поля таблицы....730
15.8.5 Поля таблицы....730
15.8.6 Номер страницы....731

15.8.7 Разрыв до и разрыв после....731
15.8.8 Фон таблицы и фоновое изображение....731
15.8.9 Тень таблицы....731
15.8.10 Вместе со следующим....732
15.8.11 Возможность разрыва между строками....732
15.8.12 Свойство модели обрамления....732
15.8.13 Режим записи....733
15.8.14 Отображение....733
15.9 Свойства форматирования столбцов....733
15.9.1 Ширина столбца....734
15.9.2 Оптимальная ширина столбца таблицы....734
15.9.3 Разрыв до и разрыв после....735
15.10 Свойства форматирования строк таблиц....735
15.10.1 Высота строки....735
15.10.2 Оптимальная высота строки таблицы....736
15.10.3 Фон строки....736
15.10.4 Разрыв до и разрыв после....736
15.10.5 Не разрывать....736
15.11 Свойства форматирования ячеек таблиц....737
15.11.1 Выравнивание по вертикали....737
15.11.2 Источник выравнивания текста....738
15.11.3 Направление....738
15.11.4 Ориентация глифа по вертикали....739
15.11.5 Тень ячейки....739
15.11.6 Фон ячейки....739
15.11.7 Обрамление ячейки....739
15.11.8 Диагональные линии....740
15.11.9 Толщина линии обрамления....740
15.11.10 Отступы от содержимого....741
15.11.11 Перенос по словам....741
15.11.12 Угол поворота....741
15.11.13 Выравнивание при повороте....742
15.11.14 Защита ячейки....742
15.11.15 Печатать содержимое....743
15.11.16 Десятичные разряды....743
15.11.17 Повторять содержимое....743
15.11.18 Подгонять по размеру....744

15.12 Свойства стилей уровней списков....744
15.13 Свойства контурной линии....747
15.13.1 Стиль контурной линии....747
15.13.2 Штрих....748
15.13.3 Составные штрихи....748
15.13.4 Толщина....748
15.13.5 Цвет....749
15.13.6 Начальный маркер....749
15.13.7 Конечный маркер....749
15.13.8 Ширина начального маркера....749
15.13.9 Ширина конечного маркера....750
15.13.10 Центрировать начальный маркер....750
15.13.11 Центрировать конечный маркер....750
15.13.12 Непрозрачность....750
15.13.13 Соединение линий....751
15.14 Свойства заполнения....751
15.14.1 Стиль заполнения....752
15.14.2 Цвет....752
15.14.3 Вспомогательный цвет заполнения....753
15.14.4 Градиент....753
15.14.5 Количество шагов градиента....753
15.14.6 Штриховка....754
15.14.7 Сплошная штриховка....754
15.14.8 Заполняющее изображение....754
15.14.9 Стиль визуализации заполняющего изображения....754
15.14.10 Размер заполняющего изображения....755
15.14.11 Опорная точка плитки заполняющего изображения....755
15.14.12 Смещение плитки заполняющего изображения....756
15.14.13 Непрозрачность и линейная непрозрачность....757
15.14.14 Градиентная непрозрачность....757
15.14.15 Правило заполнения....757
15.14.16 Цвет символов....758
15.15 Свойства анимации текста....758
15.15.1 Анимация....758
15.15.2 Направление анимации....759
15.15.3 Запуск анимации внутри....759
15.15.4 Остановка анимации внутри....760

15.15.5 Повтор анимации....760
15.15.6 Задержка анимации....760
15.15.7 Шаги анимации....760
15.16 Свойства текста и выравнивания текста....761
15.16.1 Автоматически увеличивать ширину и высоту....761
15.16.2 Подгонять по размеру....761
15.16.3 Подгонять по контуру....762
15.16.4 Вертикальное выравнивание области текста....762
15.16.5 Выравнивание области текста по горизонтали....762
15.16.6 Перенос слов....763
15.16.7 Стили списков....763
15.17 Свойства цвета....763
15.17.1 Цветовой режим....764
15.17.2 Инверсия цвета....764
15.17.3 Регулировка яркости....764
15.17.4 Регулировка контрастности....764
15.17.5 Регулировка гаммы....765
15.17.6 Регулировка красного....765
15.17.7 Регулировка зеленого....765
15.17.8 Регулировка синего....765
15.17.9 Регулировка непрозрачности....766
15.18 Свойства тени....766
15.18.1 Тень....766
15.18.2 Смещение....766
15.18.3 Цвет....767
15.18.4 Непрозрачность....767
15.19 Свойства соединительных элементов....767
15.19.1 Отступ начала линии....767
15.19.2 Отступ конца линии....768
15.20 Свойства измерений....768
15.20.1 Расстояние до линии....768
15.20.2 Выступ направляющих....769
15.20.3 Расстояние до направляющих....769
15.20.4 Начальная направляющая....769
15.20.5 Конечная направляющая....770
15.20.6 Расположение....770
15.20.7 Параллельность....770

15.20.8 Выравнивание текста....770
15.20.9 Единица измерения....771
15.20.10 Отображать единицу измерения....771
15.20.11 Десятичные разряды....772
15.21 Свойства примечаний....772
15.21.1 Тип....772
15.21.2 Тип угла....773
15.21.3 Угол....773
15.21.4 Отступ....773
15.21.5 Направление выхода....774
15.21.6 Точка выхода....774
15.21.7 Длина линии....774
15.21.8 Подгонять длину линии....775
15.22 Свойства трехмерной геометрии....775
15.22.1 Горизонтальные сегменты....775
15.22.2 Вертикальные сегменты....775
15.22.3 Скругление граней....776
15.22.4 Режим скругления граней....776
15.22.5 Перспектива....776
15.22.6 Глубина....777
15.22.7 Невидимая поверхность....777
15.22.8 Конечный угол....777
15.22.9 Закрытая фронтальная поверхность....778
15.22.10 Закрытая задняя поверхность....778
15.23 Свойства трехмерного освещения....778
15.23.1 Режим....778
15.23.2 Тип нормали....779
15.23.3 Направление нормалей....779
15.24 Свойства трехмерного текстурирования....779
15.24.1 Режим генерации....780
15.24.2 Тип....780
15.24.3 Фильтр....781
15.24.4 Режим....781
15.25 Свойства трехмерных материалов....781
15.25.1 Цвета....781
15.25.2 Глянцевость....782
15.26 Свойства трехмерных теней....782

15.26.1 Тень....782
15.27 Свойства форматирования фреймов....783
15.27.1 Ширина фрейма....783
15.27.2 Высота фрейма....783
15.27.3 Максимальные ширина и высота....784
15.27.4 Левое и правое поля....784
15.27.5 Верхнее и нижнее поля....785
15.27.6 Поля....785
15.27.7 Печатать содержимое....785
15.27.8 Защита....785
15.27.9 Горизонтальное положение....786
15.27.10 Горизонтальное отношение....787
15.27.11 Вертикальное положение....788
15.27.12 Вертикальное отношение....790
15.27.13 Привязка фрейма....790
15.27.14 Обрамление....791
15.27.15 Толщина линии обрамления....791
15.27.16 Отступы....791
15.27.17 Тень....791
15.27.18 Фон....792
15.27.19 Колонки....792
15.27.20 Возможность редактирования....792
15.27.21 Обтекание....792
15.27.22 Порог динамического обтекания....793
15.27.23 Обтекание только абзацами....793
15.27.24 Контурное обтекание....794
15.27.25 Режим контурного обтекания....794
15.27.26 Сквозное обтекание....795
15.27.27 Следование за текстом....795
15.27.28 Поведение при переполнении....796
15.27.29 Зеркальное отображение....796
15.27.30 Обрезка....797
15.27.31 Влияние обтекания на расположение....797
15.28 Свойства форматирования плавающего фрейма....799
15.28.1 Отображение полосы прокрутки....799
15.28.2 Отображение обрамления....799
15.28.3 Поля....799

15.28.4 Свойства форматирования объекта....800
15.28.5 Видимая область....800
15.28.6 Вид рисунка....801
15.29 Свойства форматирования диаграмм....801
15.29.1 Масштабирование текста....802
15.30 Свойства подтипов диаграмм....802
15.30.1 Трехмерные диаграммы....802
15.30.2 Глубина диаграммы....802
15.30.3 Символ диаграммы....803
15.30.4 Размер символа диаграммы....804
15.30.5 Свойства гистограммы....804
15.30.6 Свойства биржевых диаграмм....805
15.30.7 Свойства линейной диаграммы....805
15.30.8 Свойства круговой диаграммы....806
15.30.9 Линии....806
15.30.10 Сплошные гистограммы....807
15.30.11 Гистограммы с накоплением....807
15.31 Свойства осей диаграмм....807
15.31.1 Форматы связанных данных....808
15.31.2 Видимость....808
15.31.3 Масштабирование....808
15.31.4 Отметки....809
15.31.5 Подписи....810
15.32 Общие свойства диаграмм....811
15.32.1 Текст с накоплением по вертикали....811
15.32.2 Угол поворота....812
15.32.3 Подписи данных....812
15.33 Статистические свойства....813
15.33.1 Среднее значение....813
15.33.2 Категория погрешности....813
15.34 Свойства области построения диаграммы....815
15.34.1 Источник рядов данных....816
15.35 Свойства кривой регрессии....816
15.35.1 Тип регрессии....816
15.36 Атрибуты страницы презентации....817
15.36.1 Тип перехода....817
15.36.2 Стиль перехода....818

15.36.3 Скорость перехода....821
15.36.4 Тип или семейство переходов....821
15.36.5 Подтип перехода....821
15.36.6 Управление переходом....822
15.36.7 Цвет выцветания....822
15.36.8 Длительность показа страницы....822
15.36.9 Видимость страницы....823
15.36.10 Звук....823
15.36.11 Размер фона....823
15.36.12 Видимость объектов фона....824
15.36.13 Видимость фона....824
15.36.14 Отображение заголовков....824
15.36.15 Отображение колонтитулов....824
15.36.16 Отображение номера страницы....825
15.36.17 Отображение даты и времени....825
16 Типы данных и определения в схеме....826
16.1 Типы данных....826
16.2 Другие определения....832
16.3 Суффикс схемы Relax-NG....832
17 Пакеты....833
17.1 Введение....833
17.2 Структура zip-файла....834
17.3 Шифрование....834
17.4 Поток типа MIME....835
17.5 Использование унифицированных идентификаторов ресурса (IRI) в пакетах....836
17.6 Изображение предварительного просмотра....837
17.7 Файл декларации....837
17.7.1 Схема Relax-NG....837
17.7.2 Корневой элемент декларации....838
17.7.3 Включение файла....838
17.7.4 Данные шифрования....840
17.7.5 Алгоритм....841
17.7.6 Источник ключа....842
17.7.7 Суффикс схемы Relax-NG....844
Приложение А. Строгая схема Relax NG....845
Приложение В. Указатели....847
Приложение С. Типы MIME и расширения имен файлов (не нормативное)....849

Приложение D. Набор основных возможностей (не нормативное).....851

Приложение E. Изменения по сравнению с предыдущими версиями (не нормативное)....857

E.1. Изменения, внесенные в «Open Office Specification 1.0 Committee Draft 1».....857

E.2. Изменения, внесенные в «Open Document Format for Office Applications (OpenDocument) 1.0 Committee Draft 2».....857

E.3. Изменения, внесенные в «Open Document Format for Office Applications (OpenDocument) v1.0».....858

Приложение F. Список ассистентов.....860

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационная технология

ФОРМАТ OPEN DOCUMENT ДЛЯ ОФИСНЫХ ПРИЛОЖЕНИЙ (OpenDocument) v1.0

Information technology – Open Document Format for Office Applications (OpenDocument) v1.0

Дата введения — 2011—06—01

1 Введение

1.1 Структура и цель стандарта

В данном документе описана XML-схема для офисных приложений и ее семантика. Рассматриваемая схема охватывает офисные документы, включая текстовые документы, электронные таблицы, диаграммы и графические документы, такие как рисунки и презентации, но не ограничивается этими типами документов.

Рассматриваемая схема обеспечивает высокий уровень информативности, необходимый для редактирования документов. Она описывает структуры XML для офисных документов и достаточно просто изменяется с помощью XSLT или схожих инструментов, основанных на XML.

Раздел 1 содержит введение в формат OpenDocument. Структура документа, соответствующего спецификации OpenDocument, описана в разделе 2. Раздел 3 описывает метаинформацию, которая может содержаться в таких документах. Разделы 4 и 5 описывают текст документа в целом и его абзацы. Текстовые поля описываются в разделе 6, текстовые индексы — в разделе 7.

Раздел 8 описывает таблицы документа формата OpenDocument, раздел 9 — графические объекты, раздел 10 — диаграммы, раздел 11 — формы. Содержимое, общее для всех документов, описывается в разделе 12. Интеграция

Издание официальное

разметки SMIL-анимации в схему OpenDocument описана в разделе 13. В разделе 14 описаны стили, в разделе 15 определены свойства форматирования, которые могут быть использованы в стилях. Типы данных, используемые в схеме OpenDocument, описаны в разделе 16.

Формат OpenDocument поддерживает концепцию пакетов, представленную в разделе 17.

1.2 Нотация

В данной спецификации такие ключевые слова как **«должен»** (**"shall"**), **«не должен»** (**"shall not"**), **«следует»** (**"should"**), **«не следует»** (**"should not"**) и **«может»** (**"may"**) необходимо трактовать как в приложении Н [ISO/IEC Directives], если они даны полужирным шрифтом.

1.3 Пространства имен

В таблице 1 представлены пространства имен, которые определены форматом OpenDocument, и их префиксы, используемые по умолчанию. Более подробную информацию о пространствах имен XML можно получить в спецификации *Пространства имен в XML* [xml-names].

Таблица 1 — Пространства имен XML, определенные схемой OpenDocument

Префикс	***Описание***	***Пространство имен***
office	Для всех общих фрагментов информации, которые не используются в других, более специфичных пространствах имен	urn:oasis:names:tc:opendocument:xmlns:office:1.0
meta	Для элементов и атрибутов, описывающих метаинформацию	urn:oasis:names:tc:opendocument:xmlns:meta:1.0
config	Для элементов и атрибутов, описывающих параметры, специфические для приложений	urn:oasis:names:tc:opendocument:xmlns:config:1.0
text	Для элементов и атрибутов, встречающихся в текстовых документах и текстовых частях документов других типов, таких как содержимое ячейки электронной таблицы	urn:oasis:names:tc:opendocument:xmlns:text:1.0
table	Для элементов и атрибутов, встречающихся в электронных таблицах или в определениях таблиц текстовых документов	urn:oasis:names:tc:opendocument:xmlns:table:1.0
drawing	Для элементов и атрибутов, описывающих графическое содержимое	urn:oasis:names:tc:opendocument:xmlns:drawing:1.0
presentation	Для элементов и атрибутов, описывающих презентации	urn:oasis:names:tc:opendocument:xmlns:presentation:1.0

Префикс	*Описание*	*Пространство имен*
dr3d	Для элементов и атрибутов, описывающих трехмерную графику	urn:oasis:names:tc:opendocument:xmlns: dr3d:1.0
anim	Для элементов и атрибутов, описывающих анимацию	urn:oasis:names:tc:opendocument:xmlns: animation:1.0
chart	Для элементов и атрибутов, описывающих диаграммы	urn:oasis:names:tc:opendocument:xmlns: chart:1.0
form	Для элементов и атрибутов, описывающих формы и элементы управления	urn:oasis:names:tc:opendocument:xmlns: form:1.0
script	Для элементов и атрибутов, представляющих скрипты и события	urn:oasis:names:tc:opendocument:xmlns: script:1.0
style	Для элементов и атрибутов, описывающих стили и модель наследования, используемые в формате OpenDocument, так же как некоторые общие атрибуты форматирования	urn:oasis:names:tc:opendocument:xmlns: style:1.0
number	Для элементов и атрибутов, описывающих стиль данных	urn:oasis:names:tc:opendocument:xmlns: data style:1.0
manifest	Для элементов и атрибутов, содержащихся в пакетных декларациях	urn:oasis:names:tc:opendocument:xmlns: manifest:1.0

В таблице 2 приведен список пространств имен, описанных в формате OpenDocument, которые содержат элементы и атрибуты, чья семантика совместима с элементами и атрибутами других спецификаций.

Таблица 2 — Пространства имен XML, определенные схемой OpenDocument, которые включают элементы и атрибуты, совместимые с другими стандартами

Префикс	*Описание*	*Пространство имен*
fo	Для атрибутов, совместимых с описанными в [XSL]	urn:oasis:names:tc:opendocument:xmlns: xsl-fo-compatible:1.0
svg	Для элементов и атрибутов, совместимых с описанными в [SVG]	urn:oasis:names:tc:opendocument:xmlns: svg-compatible:1.0
smil	Для элементов и атрибутов, совместимых с описанными в [SMIL20]	urn:oasis:names:tc:opendocument:xmlns: smil-compatible:1.0

Таблица 3 содержит список пространств имен, импортированных в формат OpenDocument, и их префиксы по умолчанию.

Таблица 3 — Пространства имен XML, используемые в схеме OpenDocument

Префикс	*Описание*	*Пространство имен*
dc	Пространство имен Дублинского ядра (см. [DCMI])	http://purl.org/dc/elements/1.1/
xlink	Пространство имен XLink (см. [XLink])	http://www.w3.org/1999/xlink
math	Пространство имен MathML (см. [MathML])	http://www.w3.org/1998/Math/MathML
xforms	Пространство имен XForms (см. [XForms])	http://www.w3.org/2002/xforms

1.4 Схема Relax-NG

Нормативная XML-схема для формата OpenDocument содержится внутри данной спецификации. Она может быть получена из документа спецификации путем объединения всех фрагментов схемы, содержащихся в разделах 1—16. Все фрагменты схемы имеют серый фон, а строки пронумерованы.

Язык схемы, применяемый в спецификации — Relax-NG (см. [RNG]). Значения по умолчанию атрибутов определены в спецификации [RNG-Compat], которая используется для обеспечения атрибутов значениями по умолчанию.

Схема, приведенная в этой спецификации, допускает произвольное наполнение в пределах элементов метаинформации, как описано в разделе 1.5. В Приложении А приведена схема, которая ограничивает содержимое в пределах этих элементов для атрибутов и элементов, определенных в данной спецификации.

Префикс для нормативной Relax-NG схемы:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!--
3      OASIS OpenDocument v1.0 (Second Edition)
4      Спецификация комитета №1, 19 Jul 2006
5      Relax-NG Schema
6
7      $Id$
8
9      © 2002-2005 OASIS Open
10     © 1999-2005 Sun Microsystems, Inc.
11 -->
12
13 <grammar
14     xmlns="http://relaxng.org/ns/structure/1.0"
15     xmlns:a="http://relaxng.org/ns/compatibility/annotations/1.0"
16
17     datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes"
18
19     xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0"
```

```
20      xmlns:meta="urn:oasis:names:tc:opendocument:xmlns:meta:1.0"
21      xmlns:config="urn:oasis:names:tc:opendocument:xmlns:config:1.0"
22      xmlns:text="urn:oasis:names:tc:opendocument:xmlns:text:1.0"
23      xmlns:table="urn:oasis:names:tc:opendocument:xmlns:table:1.0"
24      xmlns:draw="urn:oasis:names:tc:opendocument:xmlns:drawing:1.0"
25      xmlns:presentation="urn:oasis:names:tc:opendocument:xmlns:presentation:1.0"
26      xmlns:dr3d="urn:oasis:names:tc:opendocument:xmlns:dr3d:1.0"
27      xmlns:chart="urn:oasis:names:tc:opendocument:xmlns:chart:1.0"
28      xmlns:form="urn:oasis:names:tc:opendocument:xmlns:form:1.0"
29      xmlns:script="urn:oasis:names:tc:opendocument:xmlns:script:1.0"
30      xmlns:style="urn:oasis:names:tc:opendocument:xmlns:style:1.0"
31      xmlns:number="urn:oasis:names:tc:opendocument:xmlns:datastyle:1.0"
32      xmlns:anim="urn:oasis:names:tc:opendocument:xmlns:animation:1.0"
33
34      xmlns:dc="http://purl.org/dc/elements/1.1/"
35      xmlns:xlink="http://www.w3.org/1999/xlink"
36      xmlns:math="http://www.w3.org/1998/Math/MathML"
37      xmlns:xforms="http://www.w3.org/2002/xforms"
38
39      xmlns:fo="urn:oasis:names:tc:opendocument:xmlns:xsl-fo-compatible:1.0"
40      xmlns:svg="urn:oasis:names:tc:opendocument:xmlns:svg-compatible:1.0"
41      xmlns:smil="urn:oasis:names:tc:opendocument:xmlns:smil-compatible:1.0"
42  >
```

1.5 Обработка документа и его соответствие спецификации

Документы, соответствующие спецификации OpenDocument, **могут** содержать элементы и атрибуты, не определенные данной схемой OpenDocument. Такие элементы и атрибуты не должны быть частью пространств имен, которые определены в данной спецификации, и называются внешними элементами и атрибутами.

Приложения, соответствующие спецификации, **должны** либо читать документы, соответствующие схеме, если все внешние элементы и атрибуты удалены до проверки соответствия, либо **должны** записывать документы, соответствующие схеме, если все внешние элементы и атрибуты удалены до проверки соответствия.

Приложения, соответствующие схеме, которые читают и записывают документы, **могут** сохранять внешние элементы и атрибуты.

Кроме того, приложения, соответствующие схеме, должны сохранять метаинформацию и содержание стилей. Это означает:

- Различные элементы `<style:*-properties>` (см. раздел 15) **могут** иметь произвольные атрибуты и **могут** иметь произвольное содержание элементов. Все атрибуты, относящиеся к таким элементам, и элементы, содержащиеся в них, **следует** сохранять (см. раздел 15.1.3).

- Элементы, содержащиеся в элементе `<office:meta>`, **могут** иметь произвольное содержимое, и их **следует** сохранять (см. раздел 2.2.1).

Внешние элементы **могут** иметь атрибут `office:process-content` с возможными значениями `true` или `false`. Если значение атрибута `true`, или если атрибут не существует, содержимое элемента **следует** обрабатывать приложениям, соответствующим данной спецификации. В противном случае, приложениям **не следует** обрабатывать содержимое элементов, а **можно** только сохранять их содержимое. Если содержимое элемента требуется обработать, сам документ **должен** соответствовать схеме OpenDocument при условии замены неизвестных элементов на их содержимое.

Приложения, соответствующие схеме, **должны** читать документы, содержащие инструкции обработки, и данным приложениям **следует** сохранять их.

Не существует правил относительно элементов и атрибутов, которые следует поддерживать приложениям, соответствующим схеме, за исключением того, что приложения не должны использовать внешние элементы и атрибуты для реализации функциональности схемы OpenDocument (см. также Приложение D).

```
43 <define name="office-process-content">
44     <optional>
45         <attribute name="office:process-content" a:defaultValue="true">
46             <ref name="boolean"/>
47         </attribute>
48     </optional>
49 </define>
```

1.6 Обработка пробельных символов и символов конца строки

В соответствии со спецификацией **W3C** XML [XML1.0] необязательные пробельные символы, которые содержатся в элементах, являющимися контейнерами других элементов (содержащими только другие элементы, исключая текст), игнорируются. Данное правило применимо к следующим пробельным символам и символам конца строки (EOL) [UNICODE]:

- ГОРИЗОНТАЛЬНАЯ ТАБУЛЯЦИЯ (0x0009)
- ПЕРЕВОД СТРОКИ (0x000A)
- ВОЗВРАТ КАРЕТКИ (0x000D)
- ПРОБЕЛ (0x0020)

Для любых других элементов пробельные символы сохраняются по умолчанию. Если отдельно не оговорено, никакая другая специальная обработка пробельных символов не осуществляется. Для некоторых элементов возможна другая обработка пробельных символов, например, для элемента абзаца.

Спецификация XML также требует, чтобы любой из четырех пробельных символов, которые содержатся в значении атрибута, преобразовывался в символ ПРОБЕЛА.

Один из следующих символов может быть использован для отображения конца строки:

- ПЕРЕВОД СТРОКИ
- ВОЗВРАТ КАРЕТКИ
- последовательность символов ПЕРЕВОД СТРОКИ и ВОЗВРАТ КАРЕТКИ

В соответствии со спецификацией XML, все возможные варианты конца строки преобразуются в один символ ПЕРЕВОД СТРОКИ.

В соответствии с правилами обработки пробельных символов и символов конца строки, любой символ ВОЗВРАТА КАРЕТКИ, который находится либо в текстовом содержимом элемента, либо в значении атрибута, заменяется на символьную сущность `
`. То же осуществляется с символами ГОРИЗОНТАЛЬНАЯ ТАБУЛЯЦИЯ и ПЕРЕВОД СТРОКИ, если они содержатся в значениях атрибута.

1.7 Типы MIME и расширения имен файлов

Приложение С содержит список типов MIME и расширений имен файлов, используемых для офисных документов, соответствующих данной спецификации, и которые содержатся в пакетах (см. раздел 2.1). Эти типы MIME и расширения файлов либо уже зарегистрированы в соответствии с процедурами, описанными в [RFC2048], либо находятся в процессе регистрации.

Для офисных документов, которые соответствуют данной спецификации, но не содержатся в пакетах, **следует** использовать тип MIME `text/xml`.

Для офисных документов, соответствующих данной спецификации, **следует** использовать только типы MIME и расширения имен файлов, зарегистрированные

в соответствии с [RFC2048]. Типы MIME и расширения имен файлов, перечисленные в Приложении С, **следует** использовать по назначению.

2 Структура документа

В данном разделе описана структура формата OpenDocument. Раздел содержит следующие подразделы:

- корни документа;
- метаданные документа;
- элемент основной части документа и типы документов;
- настройки приложений;
- скрипты;
- объявления начертания шрифта;
- стили;
- стили и макеты страниц.

В формате OpenDocument каждый структурный компонент представлен **элементом** со связанными с ним **атрибутами**. Структура документа в формате OpenDocument применяется ко всем типам документов. Не существует разницы между текстовыми документами, электронными таблицами или графическими документами, различие лишь в их содержимом. Кроме того, все типы документов могут содержать различные стили. Содержимое документа, общее для всех типов документов, может быть использовано для обмена информацией между различными типами документов.

2.1 Корни документа

Корневой элемент документа — первичный элемент документа формата OpenDocument. Он содержит весь документ. Все типы документа, например текстовые документы, электронные таблицы и графические документы, используют одинаковые типы корневых элементов документа.

Формат OpenDocument поддерживает следующие два способа представления документа.

- Как единый XML-документ.
- Как набор нескольких поддокументов внутри пакета (см. раздел 17), каждый из которых содержит часть целого документа. Каждый

поддокумент имеет свой собственный корень и содержит отдельный аспект XML-документа. Например, один поддокумент содержит стили, другой — содержимое документа. Все типы документов, например текстовые документы и электронные таблицы, используют общие определения документов и поддокументов.

Существует четыре типа поддокументов, каждый из которых отличается корневым элементом. Кроме того, единый XML-документ имеет свой собственный корневой элемент, общий для всех пяти поддерживаемых корневых элементов. Корневые элементы приведены в следующей таблице:

Корневой элемент	***Содержимое поддокумента***	***Название поддокумента в пакете***
`<office:document>`	Офисный документ целиком в едином XML-документе	н/д
`<office:document-content>`	Содержимое документа и используемые в нем автоматические стили	content.xml
`<office:document-styles>`	Стили, используемые в содержимом документа, и автоматические стили, используемые в самих стилях	styles.xml
`<office:document-meta>`	Метаинформация документа, такая как автор или время последнего сохранения	meta.xml
`<office:document-settings>`	Настройки, специфические для приложения, такие как размер окна или параметры печати	settings.xml

Определения корневых элементов, описанных в таблице выше, аналогичны определению `<office:document>`, за исключением того, что спецификация дочернего элемента соответственно ограничена.

```
<start>
    <choice>
        <ref name="office-document"/>
        <ref name="office-document-content"/>
        <ref name="office-document-styles"/>
        <ref name="office-document-meta"/>
        <ref name="office-document-settings"/>
    </choice>
</start>
```

2.1.1 Модели содержимого корневых элементов документа

В следующей таблице приведены модели содержимого пяти корневых элементов. Заметим, что `<office:document>` может содержать все поддерживаемые элементы верхнего уровня. Ни один из четырех корневых элементов поддокументов по отдельности не содержит информацию в полном объеме, в то время как в совокупности — содержат.

Корневой элемент	*Метаданные*	*Настройки приложения*	*Скрипты*	*Определения шрифтов*	*Стили*	*Автоматические стили*	*Стили составных документов*	*Основная часть документа*
`<office:document>`	✓	✓	✓	✓	✓	✓	✓	✓
`<office:document-content>`			✓	✓		✓		✓
`<office:document-styles>`				✓	✓	✓	✓	
`<office:document-meta>`	✓							
`<office:document-settings>`		✓						

Корень `<office:document>` содержит документ целиком.

```
<define name="office-document">
    <element name="office:document">
        <ref name="office-document-attrs"/>
        <ref name="office-document-common-attrs"/>
        <ref name="office-meta"/>
        <ref name="office-settings"/>
        <ref name="office-scripts"/>
        <ref name="office-font-face-decls"/>
        <ref name="office-styles"/>
        <ref name="office-automatic-styles"/>
        <ref name="office-master-styles"/>
        <ref name="office-body"/>
    </element>
</define>
```

Корень `<office:document-content>` включает только содержимое документа наряду с требуемыми для него автоматическими стилями.

```
<define name="office-document-content">
    <element name="office:document-content">
        <ref name="office-document-common-attrs"/>
        <ref name="office-scripts"/>
        <ref name="office-font-face-decls"/>
        <ref name="office-automatic-styles"/>
        <ref name="office-body"/>
    </element>
```

```
</define>
```

Корень `<office:document-styles>` содержит все именованные стили документа наряду с необходимыми для них автоматическими стилями.

```
<define name="office-document-styles">
    <element name="office:document-styles">
        <ref name="office-document-common-attrs"/>
        <ref name="office-font-face-decls"/>
        <ref name="office-styles"/>
        <ref name="office-automatic-styles"/>
        <ref name="office-master-styles"/>
    </element>
</define>
```

Корень `<office:document-meta>` содержит метаинформацию документа.

```
<define name="office-document-meta">
    <element name="office:document-meta">
        <ref name="office-document-common-attrs"/>
        <ref name="office-meta"/>
    </element>
</define>
```

Корень `<office:document-settings>` содержит специфические настройки приложения, используемые для обработки документа.

```
<define name="office-document-settings">
    <element name="office:document-settings">
        <ref name="office-document-common-attrs"/>
        <ref name="office-settings"/>
    </element>
</define>
```

2.1.2 Атрибуты корня документа

Версия

Все корневые элементы имеют атрибут `office:version`, который показывает, к какой версии спецификации он относится. Номер версии представлен в формате `revision.version` (`ревизия.версия`). Если файл имеет версию, известную XML-процессору, данный документ может быть проверен на соответствие. В противном случае, проверка на соответствие документа необязательна, но при этом он должен быть правильно сформирован.

```
<define name="office-document-common-attrs" combine="interleave">
    <optional>
        <attribute name="office:version">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Тип MIME

Элемент `<office:document>` имеет атрибут `office:mimetype`, который показывает тип документа (текст, электронная таблица и т. д.). Данный атрибут особенно важен для простых XML-файлов, для которых существует только один способ, которым может быть определен тип документа (в пакете тип MIME также представлен в отдельном файле, см. раздел 17.4). Значениями данного атрибута являются типы MIME, которые используются для пакетного варианта офисных документов (см. раздел 1.7).

```
<define name="office-document-attrs" combine="interleave">
    <attribute name="office:mimetype">
        <ref name="string"/>
    </attribute>
</define>
```

2.2 Метаданные документа

Метаданные — главная информация о документе. В формате OpenDocument все элементы метаданных содержатся в элементе `<office:meta>`, располагаемом обычно в начале документа. Элементы метаданных могут быть опущены или встречаться несколько раз. Обновление множественных вхождений одного типа элементов зависит от приложения.

```
<define name="office-meta">
    <optional>
        <element name="office:meta">
            <ref name="office-meta-content"/>
        </element>
    </optional>
</define>

<define name="office-meta-content">
    <ref name="anyElements"/>
</define>

<define name="office-meta-content-strict">
    <zeroOrMore>
        <ref name="office-meta-data"/>
    </zeroOrMore>
</define>
```

2.2.1 Предопределенные и специфические метаданные

В схеме OpenDocument метаданные состоят из предопределенных элементов метаданных, определяемых пользователем, а также специфических метаданных. Предопределенные элементы метаданных имеют предопределенную семантику. Их **следует** обрабатывать и обновлять редактирующим приложением. На них можно ссылаться из документа, используя подходящие текстовые поля.

Определяемые пользователем метаданные — более общий механизм определения тройки: имя, тип, значение. Поддерживающие приложения могут предоставлять эти значения пользователю, используя предоставляемые типы данных. На метаданные, определяемые пользователем, можно ссылаться из документа, используя подходящие текстовые поля.

Специфические метаданные — любые элементы внутри `<office:meta>`. Так как их семантика не определена в этой спецификации, приложения, соответствующие схеме, не могут обработать или показать такие данные. Приложениям **следует** сохранять эти данные при редактировании документа.

2.2.2 Образец метаданных

Пример — Образец метаданных документа формата OpenDocument:

```
<office:meta>
    <dc:title>Title of the document</dc:title>
    <dc:description>Description/Comment for the document</dc:description>
    <meta:initial-creator>User Name</meta:initial-creator>
    <meta:creation-date>1999-10-18T12:34:56</meta:creation-date>
    <dc:creator>User Name</dc:creator>
    <dc:date>1999-10-19T15:16:17</dc:date>
    <meta:printed-by>User Name</meta:printed-by>
    <meta:print-date>1999-10-20T16:17:18</meta:print-date>
    <dc:subject>Description of the document</dc:subject>
    <meta:editing-duration>PT5H10M10S</meta:editing-duration>
    <meta:keyword>First keyword</meta:keyword>
    <meta:keyword>Second keyword</meta:keyword>
    <meta:keyword>Third keyword</meta:keyword>
    <meta:template xlink:type="simple"

xlink:href="file:///c|/office52/share/template/german/finance/budget.vor"
    xlink:title="Template name"
    meta:date="1999-10-15T10:11:12" />
    <meta:auto-reload
    xlink:type="simple"
    xlink:href="file:///..."
    meta:delay="P60S" />
    <dc:language>de-DE</dc:language>
    <meta:user-defined meta:name="Field 1"
    meta:value-type="string">Value 1</meta:user-defined>
    <meta:user-defined meta:name="Field 2"
    meta:value-type="float">1.234</meta:user-defined>
</office:meta>
```

2.3 Элемент основной части документа и типы документов

Основная часть документа — это элемент, отображающий тип содержимого этого документа. В настоящее время поддерживаются следующие типы документов:

- текстовые документы;
- графические документы;
- документы презентаций;
- документы электронных таблиц;
- документы диаграмм;
- документы растровой графики.

Все типы документов используют одинаковые элементы содержимого, но каждый тип документа накладывает свои ограничения на использование различных элементов и их комбинации. В содержимом документа обычно выделяются пролог и эпилог, которые содержат дополнительную информацию для

специфических типов документов, такую как данные формы или объявление переменных.

```
<define name="office-body">
    <element name="office:body">
        <ref name="office-body-content"/>
    </element>
</define>
```

2.3.1 Текстовые документы

Содержимое текстовых документов в основном состоит из последовательности, содержащей любое число абзацев, таблиц, индексов, текстовых фреймов, текстовых разделов и графических элементов. В дополнение к этому, текстовый документ может содержать формы, отслеживание изменений и объявления переменных. Каждый из них определяется в прологе, и на них можно ссылаться из содержимого документа.

```
<define name="office-body-content" combine="choice">
    <element name="office:text">
        <ref name="office-text-attlist"/>
        <ref name="office-text-content-prelude"/>
        <zeroOrMore>
            <ref name="office-text-content-main"/>
        </zeroOrMore>
        <ref name="office-text-content-epilogue"/>
    </element>
</define>
```

Модель содержимого текстового документа

Пролог текстового документа содержит данные формы документа, отслеживание изменений и объявления переменных. Для того чтобы разрешить офисным приложениям использовать в текстовых документах функциональные возможности, присущие электронным таблицам, они также могут содержать элементы, реализующие расширенные табличные свойства (см. раздел 2.3.4).

```
<define name="office-text-content-prelude">
    <ref name="office-forms"/>
    <ref name="text-tracked-changes"/>
    <ref name="text-decls"/>
    <ref name="table-decls"/>
</define>
```

Содержимое основного документа состоит из любой последовательности элементов текстового содержимого, которое включает абзацы (и заголовки), текстовые разделы (и индексы), таблицы, объекты векторной графики. В качестве

альтернативы текстовый документ может содержать отдельную последовательность страниц.

Необязательно текстовый документ должен содержать абзац. Текстовый документ может состоять из единственной последовательности фреймов.

```
<define name="office-text-content-main">
    <choice>
        <zeroOrMore>
            <ref name="text-content"/>
        </zeroOrMore>
        <group>
            <ref name="text-page-sequence"/>
            <zeroOrMore>
                <choice>
                    <ref name="draw-a"/>
                    <ref name="shape"/>
                </choice>
            </zeroOrMore>
        </group>
    </choice>
</define>

<define name="text-content">
    <choice>
        <ref name="text-h"/>
        <ref name="text-p"/>
        <ref name="text-list"/>
        <ref name="text-numbered-paragraph"/>
        <ref name="table-table"/>
        <ref name="draw-a"/>
        <ref name="text-section"/>
        <ref name="text-table-of-content"/>
        <ref name="text-illustration-index"/>
        <ref name="text-table-index"/>
        <ref name="text-object-index"/>
        <ref name="text-user-index"/>
        <ref name="text-alphabetical-index"/>
        <ref name="text-bibliography"/>
        <ref name="shape"/>
        <ref name="change-marks"/>
    </choice>
</define>
```

Для текстовых документов нет специфических элементов эпилога, но эпилог может содержать элементы, которые реализуют расширенные табличные свойства (см. раздел 2.3.4).

```
<define name="office-text-content-epilogue">
    <ref name="table-functions"/>
</define>
```

Составные текстовые документы

Существует общий вариант использования больших документов, редактируемых в разных сущностях, основанный на том, что составной документ

содержит несколько связанных поддокументов. Это может быть реализовано при использовании связанных разделов текста (см. раздел 4.4). С целью облегчения редактирующему приложению адаптации пользовательского интерфейса для лучшей поддержки представления составного документа с составляющими частями (в отличие от документа с произвольно связанными частями) может быть использован флаг `text:global`. Если он установлен в значение `true`, то это сообщает приложению, что связанные разделы документа несут семантику «часть — целое». Фактическое XML-представление раздела при этом не изменяется.

```
<define name="office-text-attlist" combine="interleave">
    <optional>
        <attribute name="text:global" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

2.3.2 Графические документы

Содержимое графических документов состоит из последовательности страниц изображений.

```
<define name="office-body-content" combine="choice">
    <element name="office:drawing">
        <ref name="office-drawing-attlist"/>
        <ref name="office-drawing-content-prelude"/>
        <ref name="office-drawing-content-main"/>
        <ref name="office-drawing-content-epilogue"/>
    </element>
</define>

<define name="office-drawing-attlist">
    <empty/>
</define>
```

Модель содержимого графического элемента

Пролог графического документа может содержать только текстовые объявления. Для того чтобы разрешить офисным приложениям использовать в графических документах функциональные возможности, присущие электронным таблицам, они также могут содержать элементы, реализующие расширенные табличные свойства (см. раздел 2.3.4).

```
<define name="office-drawing-content-prelude">
    <ref name="text-decls"/>
    <ref name="table-decls"/>
</define>
```

Содержимое главного документа состоит из последовательности страниц изображений.

```
<define name="office-drawing-content-main">
    <zeroOrMore>
        <ref name="draw-page"/>
    </zeroOrMore>
</define>
```

Для графических документов нет специфических элементов эпилога, но эпилог может содержать элементы, которые реализуют расширенные табличные свойства (см. раздел 2.3.4).

```
<define name="office-drawing-content-epilogue">
    <ref name="table-functions"/>
</define>
```

2.3.3 Документы презентаций

Содержимое документов презентаций состоит из последовательности страниц изображений.

```
<define name="office-body-content" combine="choice">
    <element name="office:presentation">
        <ref name="office-presentation-attlist"/>
        <ref name="office-presentation-content-prelude"/>
        <ref name="office-presentation-content-main"/>
        <ref name="office-presentation-content-epilogue"/>
    </element>
</define>

<define name="office-presentation-attlist">
    <empty/>
</define>
```

Модель содержимого документа презентации

Пролог документа презентации идентичен графическому документу, но может содержать некоторые дополнительные определения (см. раздел 2.3.2).

```
<define name="office-presentation-content-prelude">
    <ref name="text-decls"/>
    <ref name="table-decls"/>
    <ref name="presentation-decls"/>
</define>
```

Содержимое главного документа состоит из последовательности страниц изображений.

```
<define name="office-presentation-content-main">
    <zeroOrMore>
        <ref name="draw-page"/>
    </zeroOrMore>
</define>
```

Эпилог документов презентаций может содержать настройки презентации. Дополнительно он может включать элементы, которые реализуют расширенные табличные свойства (см. раздел 2.3.4).

```
<define name="office-presentation-content-epilogue">
    <ref name="presentation-settings"/>
    <ref name="table-functions"/>
</define>
```

2.3.4 Документы электронных таблиц

Содержимое документов электронных таблиц в основном состоит из последовательности таблиц. Дополнительно документы электронных таблиц могут содержать формы, информацию о произведенных изменениях и различные виды объявлений, которые упрощают использование электронных таблиц и их анализ. Каждый из них содержится либо в прологе документа, либо в эпилоге.

```
<define name="office-body-content" combine="choice">
    <element name="office:spreadsheet">
        <ref name="office-spreadsheet-attlist"/>
        <ref name="office-spreadsheet-content-prelude"/>
        <ref name="office-spreadsheet-content-main"/>
        <ref name="office-spreadsheet-content-epilogue"/>
    </element>
</define>
```

Модель содержимого документа электронных таблиц

Пролог документа электронных таблиц содержит данные формы документа, информацию о произведенных изменениях, настройки вычислений для формул, правила проверки содержимого ячеек и объявления меток диапазонов.

```
<define name="office-spreadsheet-content-prelude">
    <optional>
        <ref name="table-tracked-changes"/>
    </optional>
    <ref name="text-decls"/>
    <ref name="table-decls"/>
</define>

<define name="table-decls">
    <optional>
        <ref name="table-calculation-settings"/>
    </optional>
    <optional>
        <ref name="table-content-validations"/>
    </optional>
    <optional>
        <ref name="table-label-ranges"/>
    </optional>
</define>
```

Главный документ — это список таблиц.

```
<define name="office-spreadsheet-content-main">
    <zeroOrMore>
        <ref name="table-table"/>
    </zeroOrMore>
</define>
```

Эпилог документов электронных таблиц содержит объявления для именованных выражений, диапазонов баз данных, сводных таблиц, операций по консолидации и DDE-ссылок.

```
<define name="office-spreadsheet-content-epilogue">
    <ref name="table-functions"/>
</define>

<define name="table-functions">
    <optional>
        <ref name="table-named-expressions"/>
    </optional>
    <optional>
        <ref name="table-database-ranges"/>
    </optional>
    <optional>
        <ref name="table-data-pilot-tables"/>
    </optional>
    <optional>
        <ref name="table-consolidation"/>
    </optional>
    <optional>
        <ref name="table-dde-links"/>
    </optional>
</define>
```

2.3.5 Документы диаграмм

Содержимое документов диаграмм в основном состоит из элементов диаграмм.

```
<define name="office-body-content" combine="choice">
    <element name="office:chart">
        <ref name="office-chart-attlist"/>
        <ref name="office-chart-content-prelude"/>
        <ref name="office-chart-content-main"/>
        <ref name="office-chart-content-epilogue"/>
    </element>
</define>

<define name="office-chart-attlist">
    <empty/>
</define>
```

Модель содержимого документа диаграмм

Для того, чтобы позволить офисным приложениям использовать для таблиц, которые могут содержаться в диаграммах, функциональные возможности,

присущие электронным таблицам, пролог документа диаграмм может содержать элементы, реализующие расширенные табличные свойства (см. раздел 2.3.4).

```
<define name="office-chart-content-prelude">
    <ref name="text-decls"/>
    <ref name="table-decls"/>
</define>
```

Главный документ представляет собой единственный элемент диаграммы.

```
<define name="office-chart-content-main">
    <ref name="chart-chart"/>
</define>
```

Для документов диаграмм не существует специальных элементов, описываемых в эпилоге. Однако эпилог может содержать элементы, которые реализуют расширенные табличные свойства (см. раздел 2.3.4).

```
<define name="office-chart-content-epilogue">
    <ref name="table-functions"/>
</define>
```

2.3.6 Документы растровой графики

Содержимое документа растровой графики представляет собой только элемент фрейма. Элемент фрейма должен содержать единственный элемент изображения.

```
<define name="office-body-content" combine="choice">
    <element name="office:image">
        <ref name="office-image-attlist"/>
        <ref name="office-image-content-prelude"/>
        <ref name="office-image-content-main"/>
        <ref name="office-image-content-epilogue"/>
    </element>
</define>

<define name="office-image-attlist">
    <empty/>
</define>
```

Модель содержимого документа растровой графики

Пролог документа растровой графики пуст.

```
<define name="office-image-content-prelude">
    <empty/>
</define>
```

Содержимое главного документа состоит только из одного фрейма.

```
<define name="office-image-content-main">
    <ref name="draw-frame"/>
</define>
```

Для документов растровой графики не существует специальных элементов, описываемых в эпилоге.

```
<define name="office-image-content-epilogue">
    <empty/>
</define>
```

2.4 Параметры настройки приложений

Параметры настройки приложений содержатся в элементе `<office:settings>`.

```
<define name="office-settings">
    <optional>
        <element name="office:settings">
            <oneOrMore>
                <ref name="config-config-item-set"/>
            </oneOrMore>
        </element>
    </optional>
</define>
```

Параметры настройки офисных приложений можно разделить на несколько категорий, каждая из которых представлена элементом `<config:config-item-set>`. Например, могут существовать две категории:

- параметры документа, например принтер, используемый по умолчанию;
- параметры просмотра, например масштаб.

2.4.1 Последовательность параметров настройки

Элемент `<config:config-item-set>` является контейнером для всех типов параметров настройки. Параметры настройки могут содержаться в элементе в любом порядке.

```
<define name="config-config-item-set">
    <element name="config:config-item-set">
        <ref name="config-config-item-set-attlist"/>
        <ref name="config-items"/>
    </element>
</define>

<define name="config-items">
    <oneOrMore>
        <choice>
            <ref name="config-config-item"/>
            <ref name="config-config-item-set"/>
            <ref name="config-config-item-map-named"/>
            <ref name="config-config-item-map-indexed"/>
        </choice>
    </oneOrMore>
</define>
```

Имя параметров настройки

Атрибут `config:name` определяет имя контейнера параметров настройки. Для элементов верхнего уровня `<config:config-item-set>` существуют элементы, непосредственно дочерние элемента `<office:settings>`, имени которых должен предшествовать префикс пространства имен, определяющий приложение, к которому относятся параметры настройки.

```
<define name="config-config-item-set-attlist" combine="interleave">
    <attribute name="config:name">
        <ref name="string"/>
    </attribute>
</define>
```

Пример —

```
<office:settings>
    <config:config-item-set xmlns:ooo="http://www.openoffice.org/...";
                            config:name="ooo:view-settings">
        <config:config-item config:name="ViewAreaTop"
                            config:type="int">0</config:config-item>
    </config:config-item-set>
</office:settings>
```

2.4.2 Основные параметры настройки

Элемент `<config:config-item>` содержит все основные параметры настройки. Значение параметра хранится в элементе.

```
<define name="config-config-item">
    <element name="config:config-item">
        <ref name="config-config-item-attlist"/>
        <text/>
    </element>
</define>
```

Имя параметра настройки

Атрибут `config:name` устанавливает имя параметра настройки.

```
<define name="config-config-item-attlist" combine="interleave">
    <attribute name="config:name">
        <ref name="string"/>
    </attribute>
</define>
```

Тип параметра настройки

Атрибут `config:type` устанавливает тип данных параметра настройки.

```
<define name="config-config-item-attlist" combine="interleave">
    <attribute name="config:type">
        <choice>
```

```
            <value>boolean</value>
            <value>short</value>
            <value>int</value>
            <value>long</value>
            <value>double</value>
            <value>string</value>
            <value>datetime</value>
            <value>base64Binary</value>
        </choice>
    </attribute>
</define>
```

2.4.3 Индексный доступ к последовательностям

Элемент `<config:config-item-map-indexed>` является контейнером для последовательностей. Порядок определяет индекс элементов.

```
<define name="config-config-item-map-indexed">
    <element name="config:config-item-map-indexed">
        <ref name="config-config-item-map-indexed-attlist"/>
        <oneOrMore>
            <ref name="config-config-item-map-entry"/>
        </oneOrMore>
    </element>
</define>
```

Имя параметров настройки

Атрибут `config:name` устанавливает имя последовательности параметров настройки.

```
<define name="config-config-item-map-indexed-attlist" combine="interleave">
    <attribute name="config:name">
        <ref name="string"/>
    </attribute>
</define>
```

2.4.4 Запись в ассоциированный массив

Элемент `<config:config-item-map-entry>` представляет запись в проиндексированной или именованной последовательности параметров настройки. Он является контейнером для всех типов элементов параметров настройки.

```
<define name="config-config-item-map-entry">
    <element name="config:config-item-map-entry">
        <ref name="config-config-item-map-entry-attlist"/>
        <ref name="config-items"/>
    </element>
</define>
```

Имя параметров настройки

Атрибут `config:name` устанавливает имя последовательности параметров настройки.

```
<define name="config-config-item-map-entry-attlist" combine="interleave">
    <optional>
        <attribute name="config:name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

2.4.5 Именованный доступ к последовательностям

Элемент `<config:config-item-map-named>` является контейнером для последовательностей, где каждый параметр настройки в последовательности определяется по имени.

```
<define name="config-config-item-map-named">
    <element name="config:config-item-map-named">
        <ref name="config-config-item-map-named-attlist"/>
        <oneOrMore>
            <ref name="config-config-item-map-entry"/>
        </oneOrMore>
    </element>
</define>
```

Имя параметров настройки

Атрибут `config:name` устанавливает имя последовательности параметров настройки.

```
<define name="config-config-item-map-named-attlist" combine="interleave">
    <attribute name="config:name">
        <ref name="string"/>
    </attribute>
</define>
```

2.4.6 Параметры настройки позиции курсора

Позиция текстового курсора во время сохранения документа является общим параметром настройки визуального отображения для приложений, осуществляющих редактирование. Для WYSIWYG-приложений это, как правило, только позиция в абзаце. Для приложений, которые обеспечивают просмотр документа на основе XML, позиция курсора также может быть между произвольными элементами, даже внутри тегов.

Для отображения позиции текстового курсора в документе **следует** использовать инструкцию обработки `PITarget opendocument` (см. § 2.6 в [XML1.0]). Имя инструкции обработки позиции курсора, `cursor-position`, **должно** следовать `PITarget opendocument`. Инструкция обработки может иметь произвольные атрибуты, специфические для приложения, например для связи позиции курсора с определенным визуальным отображением документа, где сами виды отображения определяются параметрами настройки, специфичными для приложения. Синтаксис для таких атрибутов **должен** быть идентичен синтаксису атрибутов в начальных XML-тегах.

Там, где позиция курсора не важна для восстановления визуального отображения документа, приложения могут использовать произвольные параметры настройки, специфические для документа, в дополнение к инструкции обработки позиции курсора. Также они могут использовать произвольные параметры настройки, специфические для документа, если позиция курсора не является позицией курсора ввода, а, например, выделение графических объектов.

Пример — Инструкция обработки позиции курсора:

```
<text:p>This is<?opendocument cursor-position view-id="view1"?> an
example.</text:p>
```

2.5 Скрипты

Документ может содержать несколько скриптов на разных языках. Каждый скрипт представлен элементом `<office:script>`. Все такие элементы скриптов содержатся в единственном элементе `<office:scripts>`.

Понятие скрипт не накладывает ограничения на язык скрипта или объектную модель. Скрипт может работать с объектной моделью документа (Document Object Model (DOM)) в формате OpenDocument или со специфичными функциями API приложения.

Скрипты не могут изменять документ во время его загрузки. Однако некоторые события вызываются немедленно после загрузки документа.

В дополнение к элементам `<office:script>`, элемент `<office:scripts>` может также содержать элемент `<office:event-listeners>`, который содержит события, связанные с самим документом. Примером являются события, вызываемые во время открытия или закрытия

документа. См. раздел 12.4 для получения более подробной информации об элементе `<office:event-listeners>`.

```
<define name="office-scripts">
    <optional>
        <element name="office:scripts">
            <zeroOrMore>
                <ref name="office-script"/>
            </zeroOrMore>
            <optional>
                <ref name="office-event-listeners"/>
            </optional>
        </element>
    </optional>
</define>
```

2.5.1 Скрипт

Элемент `<office:script>` содержит специфичное для языка скрипта содержимое. В большинстве случаев элемент содержит исходный код скрипта, но также он может содержать скомпилированную версию скрипта или ссылку на некоторый внешний код скрипта.

```
<define name="office-script">
    <element name="office:script">
        <ref name="office-script-attlist"/>
        <mixed>
            <ref name="anyElements"/>
        </mixed>
    </element>
</define>
```

Язык скрипта

Атрибут `script:language` определяет язык скрипта по его названию. Так как названия языков скрипта специфичны для приложения, названию должен предшествовать префикс пространства имен.

```
<define name="office-script-attlist">
    <attribute name="script:language">
        <ref name="string"/>
    </attribute>
</define>
```

2.6 Объявления начертания шрифта

Документ в формате OpenDocument может содержать объявления начертания шрифта. Объявление начертания шрифта предусматривает информацию о шрифтах, используемых автором документа, так что эти шрифты

или шрифты, очень близкие к ним, могут быть использованы на других системах (см. раздел 14.6).

```
<define name="office-font-face-decls">
    <optional>
        <element name="office:font-face-decls">
            <zeroOrMore>
                <ref name="style-font-face"/>
            </zeroOrMore>
        </element>
    </optional>
</define>
```

2.7 Стили

Формат OpenDocument поддерживает три типа **стилей**.

- **Общие стили**

  Большинство офисных приложений поддерживает стили в пределах своего пользовательского интерфейса. В рамках данной спецификации XML-представления таких стилей упоминаются как стили. Когда требуется дифференциация с другими стилями, они упоминаются как общие стили. Термин *общий* показывает, что этот тип стиля воспринимается пользователем офисного приложения как стиль.

- **Автоматические стили**

  Автоматические стили содержат свойства форматирования, которые в пользовательском интерфейсе визуального отображения документа связаны с объектом, таким как абзац. Термин *автоматический* показывает, что стиль формируется автоматически. Другими словами, свойства форматирования, которые непосредственно связаны со специфичным объектом, представлены автоматическим стилем. Таким образом достигается разделение содержимого и дизайна.

- **Мастер-стили**

  Мастер-стиль — это общий стиль, который содержит информацию о форматировании и дополнительное содержимое, отображаемое с содержимым документа при применении стиля. Примером мастер-стиля являются мастер-страницы. Мастер-страницы могут быть использованы в графических приложениях. В этом случае дополнительное содержимое — векторные фигуры, отображаются в качестве фона графической страницы. Мастер-страницы также могут быть использованы в текстовых

документах. В этом случае дополнительное содержимое — это верхние и нижние колонтитулы. Отметьте, пожалуйста, что содержимое, содержащееся в мастер-стилях, — дополнительное содержимое, которое влияет на отображение документа, но не изменяет содержимого документа.

Все типы стилей являются частью документа, что соответствует заинтересованности пользователей офисных приложений. Они представляют выходной аппаратно-независимый макет страницы и информацию о форматировании, необходимые пользователю для создания или редактирования документа. Предполагается, что автор документа хочет сохранить информацию о форматировании и макете страницы во время загрузки или отображения документа на любом устройстве, так как это общепринятая практика для документов, созданных текстовыми процессорами.

Этот тип стиля отличается от таблиц стилей [CSS2] или [XSLT], которые используются для отображения документа. Дополнительная таблица стилей для CSS, XSLT и т. д. требуется для отображения документа в формате OpenDocument на определенном устройстве. Эта таблица стилей должна учитывать стили, уже существующие в документе, так же как и требования совместимости с устройствами вывода. В идеальном варианте таблица стилей зависит только от устройств вывода.

Более подробную информацию о стилях см. в разделе 14.

2.7.1 Расположение стилей

Общие и автоматические стили имеют одинаковое XML-представление, но они содержатся в двух разных элементах-контейнерах:

- `<office:styles>` для общих стилей;
- `<office:automatic-styles>` для автоматических стилей.

Мастер-стили содержатся в своих собственных элементах-контейнерах:

- `<office:master-styles>`

```
<define name="office-styles">
    <optional>
        <element name="office:styles">
            <interleave>
                <ref name="styles"/>
                <zeroOrMore>
                    <ref name="style-default-style"/>
```

```
                </zeroOrMore>
                <optional>
                    <ref name="text-outline-style"/>
                </optional>
                <zeroOrMore>
                    <ref name="text-notes-configuration"/>
                </zeroOrMore>
                <optional>
                    <ref name="text-bibliography-configuration"/>
                </optional>
                <optional>
                    <ref name="text-linenumbering-configuration"/>
                </optional>
                <zeroOrMore>
                    <ref name="draw-gradient"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="svg-linearGradient"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="svg-radialGradient"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="draw-hatch"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="draw-fill-image"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="draw-marker"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="draw-stroke-dash"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="draw-opacity"/>
                </zeroOrMore>
                <zeroOrMore>
                    <ref name="style-presentation-page-layout"/>
                </zeroOrMore>
            </interleave>
        </element>
    </optional>
</define>
<define name="office-automatic-styles">
    <optional>
        <element name="office:automatic-styles">
            <interleave>
                <ref name="styles"/>
                <zeroOrMore>
                    <ref name="style-page-layout"/>
                </zeroOrMore>
            </interleave>
        </element>
    </optional>
</define>
<define name="office-master-styles">
    <optional>
        <element name="office:master-styles">
            <interleave>
                <zeroOrMore>
                    <ref name="style-master-page"/>
                </zeroOrMore>
```

```
                <optional>
                    <ref name="style-handout-master"/>
                </optional>
                <optional>
                    <ref name="draw-layer-set"/>
                </optional>
            </interleave>
        </element>
    </optional>
</define>

<define name="styles">
    <interleave>
        <zeroOrMore>
            <ref name="style-style"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="text-list-style"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="number-number-style"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="number-currency-style"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="number-percentage-style"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="number-date-style"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="number-time-style"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="number-boolean-style"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="number-text-style"/>
        </zeroOrMore>
    </interleave>
</define>
```

Следующие примеры иллюстрируют разные типы стилей OpenDocument.

Пример — Стили OpenDocument:

```
<office:document ...>
    <office:styles>
        ...
    </office:styles>
    <office:automatic-styles>
        ...
    </office:automatic-styles>
    <office:master-styles>
        ...
    </office:master-styles>
</office:document>
```

2.8 Стили и макеты страницы

Стили и макеты страниц в документе определяются:

- макетами страниц;
- мастер-страницами.

Макет страницы описывает физические свойства или геометрию страницы, например, размер страницы, поля, высоту верхних и нижних колонтитулов.

Мастер-страница — это шаблон для страниц в документе. Он содержит ссылку на макет страницы, который определяет физические параметры страницы, а также может содержать статическое содержимое, отображаемое на всех страницах документа, которые используют мастер-страницу. Примером статического содержимого являются колонтитулы или фоновое изображение.

Если текстовый документ или электронная таблица отображаются в режиме макета страницы, то создаются экземпляры мастер-страницы для создания последовательности страниц, содержащих содержимое документа. При создании экземпляра мастер-страницы создается пустая страница со свойствами мастер-страницы и статическим содержимом мастер-страницы. После этого основная часть документа заполняется содержимым. Если несколько страниц в документе используют одинаковую мастер-страницу, то экземпляр может быть создан несколько раз в документе.

В текстовом документе или электронной таблице мастер-страница может быть связана со стилями абзацев или таблиц через использование атрибута `style:master-page-name`. Каждый раз, когда стиль абзаца или таблицы будет использован для текста, перед абзацем или таблицей будет вставлен разрыв страницы. Страница, которая начинается с позиции разрыва страницы, использует указанную мастер-страницу.

В графических документах и презентациях мастер-страницы могут быть связаны с графической страницей через использование атрибута `style:parent-style-name`.

Примечание — Методология организации страниц в формате OpenDocument значительно отличается от методологии, принятой в [XSL]. В XSL колонтитулы содержатся внутри последовательности страниц, которые также содержат содержимое документа. В формате OpenDocument колонтитулы содержатся в стилях страницы. Таким образом, содержимое колонтитулов может быть изменено или пропущено без последствий для содержимого документа.

Макеты страниц описаны в разделе 14.3, мастер-страницы — в разделе 14.4.

3 Элементы метаданных

Элементы метаданных в большой степени заимствованы из стандарта на метаданные, разработанного проектом Dublin Core Metadata Initiative (http://www.dublincore.org). Элементы метаданных взяты напрямую из Дублинского ядра с использованием их префикса пространства имен (см. раздел 1.3).

3.1 Предопределенные элементы метаданных

Существует набор предопределенных метаданных, которые приложениям следует обрабатывать и обновлять. Элементы метаданных могут быть пропущены или встречаться неоднократно. Как обновлять множественные вхождения одинаковых элементов определяется приложением.

3.1.1 Генератор

Элемент `<meta:generator>` содержит строку, которая идентифицирует приложение или средство, которое использовалось для создания или последний модификации XML-документа. Данную строку **следует** записывать в соответствии с определением user-agent в протоколе HTTP, описанным в разделе 14.43 [RFC2616]. Данной строкой **следует** давать представления о разных версиях использованной программы, указывая, в том числе, номер сборки или номер заплатки.

Приложение, соответствующее спецификации, **может** использовать строку генератора для обхода ошибок, которые существуют или существовали в обозначенном приложении, но оно **не должно** сознательно реализовывать различное поведение приложения в зависимости от содержания данной строки.

Если приложение, создавшее документ, не может предоставить идентификационную строку, то этот элемент не используется. Если приложение, в котором данный документ модифицируется, не может записать в строку свой уникальный идентификатор, оно **не должно** использовать идентификатор приложения, создавшего документа.

```
<define name="office-meta-data" combine="choice">
    <element name="meta:generator">
```

```
        <ref name="string"/>
    </element>
</define>
```

3.1.2 Заголовок

Элемент `<dc:title>` определяет заголовок документа.

```
<define name="office-meta-data" combine="choice">
    <element name="dc:title">
        <ref name="string"/>
    </element>
</define>
```

3.1.3 Описание

Элемент `<dc:description>` содержит краткое описание документа.

```
<define name="office-meta-data" combine="choice">
    <element name="dc:description">
        <ref name="string"/>
    </element>
</define>
```

3.1.4 Тема

Элемент `<dc:subject>` определяет тему документа.

```
<define name="office-meta-data" combine="choice">
    <element name="dc:subject">
        <ref name="string"/>
    </element>
</define>
```

3.1.5 Ключевые слова

Элемент `<meta:keyword>` содержит ключевые слова, свойственные документу. Метаданные могут содержать любое число элементов `<meta:keyword>`. Каждый элемент описывает одно ключевое слово.

```
<define name="office-meta-data" combine="choice">
    <element name="meta:keyword">
        <ref name="string"/>
    </element>
</define>
```

3.1.6 Автор первоначального документа

Элемент `<meta:initial-creator>` определяет имя автора, изначально создавшего документ.

```
<define name="office-meta-data" combine="choice">
```

```
    <element name="meta:initial-creator">
        <ref name="string"/>
    </element>
</define>
```

3.1.7 Автор

Элемент `<dc:creator>` определяет имя автора последней модификации документа. Данный элемент был выбран для совместимости с Дублинским ядром, но данное определение отличается от использованного в Дублинском ядре, которое определяет создателя (автора) как «Сущность, изначально ответственную за создание содержимого ресурса». В терминологии OpenDocument, последний редактировавший документ автор в первую очередь ответственен за его содержимое.

```
<define name="office-meta-data" combine="choice">
    <ref name="dc-creator"/>
</define>
<define name="dc-creator">
    <element name="dc:creator">
        <ref name="string"/>
    </element>
</define>
```

3.1.8 Кем распечатан документ

Элемент `<meta:printed-by>` определяет имя исполнителя, который последним распечатал документ.

```
<define name="office-meta-data" combine="choice">
    <element name="meta:printed-by">
        <ref name="string"/>
    </element>
</define>
```

3.1.9 Дата и время создания документа

Элемент `<meta:creation-date>` определяет дату и время изначального создания документа.

Для соответствия [xmlschema-2] используется следующий формат даты и времени: YYYY-MM-DDThh:mm:ss.

```
<define name="office-meta-data" combine="choice">
    <element name="meta:creation-date">
        <ref name="dateTime"/>
    </element>
</define>
```

3.1.10 Дата и время последней модификации документа

Элемент `<dc:date>` определяет дату и время последнего изменения документа.

Для соответствия [xmlschema-2] используется следующий формат даты и времени: YYYY-MM-DDThh:mm:ss.

Имя данного элемента было выбрано для совместимости с Дублинским ядром.

```
<define name="office-meta-data" combine="choice">
    <ref name="dc-date"/>
</define>
<define name="dc-date">
    <element name="dc:date">
        <ref name="dateTime"/>
    </element>
</define>
```

3.1.11 Дата и время печати документа

Элемент `<meta:print-date>` определяет дату и время последней печати документа.

Для соответствия [xmlschema-2] используется следующий формат даты и времени: YYYY-MM-DDThh:mm:ss.

```
<define name="office-meta-data" combine="choice">
    <element name="meta:print-date">
        <ref name="dateTime"/>
    </element>
</define>
```

3.1.12 Шаблон документа

Элемент `<meta:template>` содержит URL на шаблон, который использовался при создании документа. URL оформляется как XLink.

Элемент подчиняется правилам, описанным в спецификации XLink (см. [XLink]).

С элементом `<meta:template>` могут быть связаны следующие атрибуты:

- местоположение шаблона;
- имя шаблона;
- дата и время последней модификации шаблона.

Местоположение шаблона

Атрибут `xlink:href` определяет местоположение шаблона документа.

Имя шаблона

Атрибут `xlink:title` определяет имя шаблона документа.

Дата и время модификации шаблона

Атрибут `meta:date` определяет дату и время последней модификации шаблона до момента его использования для создания данного документа.

Для соответствия [xmlschema-2] используется следующий формат даты и времени: YYYY-MM-DDThh:mm:ss.

```
<define name="office-meta-data" combine="choice">
    <element name="meta:template">
        <attribute name="xlink:href">
            <ref name="anyURI"/>
        </attribute>
        <optional>
            <attribute name="xlink:type" a:defaultValue="simple">
                <value>simple</value>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:actuate" a:defaultValue="onRequest">
                <value>onRequest</value>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:title">
                <ref name="string"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="meta:date">
                <ref name="dateTime"/>
            </attribute>
        </optional>
    </element>
</define>
```

3.1.13 Автоматическая перезагрузка

Элемент `<meta:auto-reload>` определяет, будет ли перезагружен или заменен документ по истечению установленного периода времени другим документом.

С элементом `<meta:auto-reload>` могут быть связаны следующие атрибуты:

- URL перезагрузки;
- задержка перед перезагрузкой.

URL перезагрузки

Если загруженный документ подлежит замене на другой по истечении указанного периода времени, элемент `<meta:auto-reload>` представляет собой XLink. Атрибут `xlink:href` определяет URL заменяющего документа.

Задержка перед перезагрузкой

Атрибут `meta:delay` определяет задержку перед перезагрузкой.

Для соответствия типу данных длительности в [xmlschema-2], используется следующий формат значений данного атрибута `PnYnMnDTnHnMnS`. См. § 3.2.6 [xmlschema-2] для получения более детальной информации о данном формате.

```
<define name="office-meta-data" combine="choice">
    <element name="meta:auto-reload">
        <optional>
            <attribute name="xlink:type" a:defaultValue="simple">
                <value>simple</value>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:show" a:defaultValue="replace">
                <value>replace</value>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:actuate" a:defaultValue="onLoad">
                <value>onLoad</value>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:href">
                <ref name="anyURI"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="meta:delay">
                <ref name="duration"/>
            </attribute>
        </optional>
    </element>
</define>
```

3.1.14 Поведение гиперссылок

Элемент `<meta:hyperlink-behaviour>` определяет поведение по умолчанию для гиперссылок в документе.

С элементом `<meta:hyperlink-behaviour>` может быть связан только один атрибут:

- целевой фрейм.

Целевой фрейм

Атрибут `meta:target-frame-name` определяет имя целевого фрейма, в котором по умолчанию будет отображаться документ, на который указывает гиперссылка.

Данный атрибут может принимать одно из следующих значений:

- `_self` — документ по ссылке заменяет содержимое текущего фрейма;
- `_blank` — документ по ссылке открывается в новом фрейме;
- `_parent` — документ по ссылке открывается в родительском по отношению к текущему документу фрейме;
- `_top` — документ по ссылке отображается в самом верхнем фрейме, это фрейм, который содержит текущей фрейм в качестве дочернего, но который не содержится внутри никакого другого фрейма;
- имя фрейма — документ по ссылке отображается в заданном фрейме, если фрейм с таким именем отсутствует, будет создан новый фрейм с заданным именем.

Для соответствия спецификации XLink добавлен дополнительный атрибут `xlink:show` для элемента `<meta:hyperlink-behaviour>`. Если значением атрибута `meta:target-frame-name` является `_blank`, то значение атрибута `xlink:show` будет `new`. Если значением атрибута `meta:target-frame-name` является одно из других возможных, то значение атрибута `xlink:show` будет `replace`.

```
<define name="office-meta-data" combine="choice">
    <element name="meta:hyperlink-behaviour">
        <optional>
            <attribute name="office:target-frame-name">
                <ref name="targetFrameName"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:show">
                <choice>
                    <value>new</value>
                    <value>replace</value>
                </choice>
            </attribute>
```

```
        </optional>
    </element>
</define>
```

3.1.15 Язык

Элемент `<dc:language>` определяет язык по умолчанию для документа.

Метод определения данного элемента такой же, как и при определении языкового тега, описанного в [RFC3066]. Возможное значение состоит из двух или трех букв языкового кода, взятого из стандарта ISO 639, затем может следовать дефис (-) и два символа кода страны, взятых из стандарта ISO 3166.

```
<define name="office-meta-data" combine="choice">
    <element name="dc:language">
        <ref name="language"/>
    </element>
</define>
```

3.1.16 Циклы редактирования

Элемент `<meta:editing-cycles>` определяет число циклов редактирования, которого достиг документ.

Значение данного элемента увеличивается каждый раз, когда документ сохраняется. Элемент содержит число циклов редактирования в виде текста.

```
<define name="office-meta-data" combine="choice">
    <element name="meta:editing-cycles">
        <ref name="nonNegativeInteger"/>
    </element>
</define>
```

3.1.17 Время редактирования

Элемент `<meta:editing-duration>` определяет общее время, затраченное на редактирование документа.

Оно представляется в типе данных длительности [xmlschema-2], который выглядит следующим образом: `PnYnMnDTnHnMnS`. См. § 3.2.6 в [xmlschema-2] для получения более детальной информации по формату записи длительности.

```
<define name="office-meta-data" combine="choice">
    <element name="meta:editing-duration">
        <ref name="duration"/>
    </element>
</define>
```

3.1.18 Статистика документа

Элемент `<meta:document-statistic>` определяет статистику документа, например число страниц, число слов и др. Статистика записывается в виде атрибутов элемента `<meta:document-statistic>`, статистика, экспортированная в документ, зависит от типа документа и от приложения, в котором он создавался:

Тип документа	*Статистические атрибуты документа*
Текстовый	meta:page-count meta:table-count meta:draw-count meta:image-count meta:ole-object-count meta:paragraph-count meta:word-count meta:character-count meta:row-count meta:frame-count meta:sentence-count meta:syllable-count meta:non-whitespace-character-count
Электронная таблица	meta:page-count meta:table-count meta:image-count meta:cell-count meta:object-count
Графический	meta:page-count meta:image-count meta:object-count

```
<define name="office-meta-data" combine="choice">
    <element name="meta:document-statistic">
        <optional>
            <attribute name="meta:page-count">
                <ref name="nonNegativeInteger"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="meta:table-count">
                <ref name="nonNegativeInteger"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="meta:draw-count">
                <ref name="nonNegativeInteger"/>
```

```
751             </attribute>
752         </optional>
753         <optional>
754             <attribute name="meta:image-count">
755                 <ref name="nonNegativeInteger"/>
756             </attribute>
757         </optional>
758         <optional>
759             <attribute name="meta:ole-object-count">
760                 <ref name="nonNegativeInteger"/>
761             </attribute>
762         </optional>
763         <optional>
764             <attribute name="meta:paragraph-count">
765                 <ref name="nonNegativeInteger"/>
766             </attribute>
767         </optional>
768         <optional>
769             <attribute name="meta:word-count">
770                 <ref name="nonNegativeInteger"/>
771             </attribute>
772         </optional>
773         <optional>
774             <attribute name="meta:character-count">
775                 <ref name="nonNegativeInteger"/>
776             </attribute>
777         </optional>
778         <optional>
779             <attribute name="frame-count">
780                 <ref name="nonNegativeInteger"/>
781             </attribute>
782         </optional>
783         <optional>
784             <attribute name="sentence-count">
785                 <ref name="nonNegativeInteger"/>
786             </attribute>
787         </optional>
788         <optional>
789             <attribute name="syllable-count">
790                 <ref name="nonNegativeInteger"/>
791             </attribute>
792         </optional>
793         <optional>
794             <attribute name="non-whitespace-character-count">
795                 <ref name="nonNegativeInteger"/>
796             </attribute>
797         </optional>
798         <optional>
799             <attribute name="meta:row-count">
800                 <ref name="nonNegativeInteger"/>
801             </attribute>
802         </optional>
803         <optional>
804             <attribute name="meta:cell-count">
805                 <ref name="nonNegativeInteger"/>
806             </antribute>
807         </optional>
808         <optional>
809             <attribute name="meta:object-count">
810                 <ref name="nonNegativeInteger"/>
811             </attribute>
812         </optional>
813     </element>
```

```
</define>
```

3.2 Метаданные, определяемые пользователем

Элемент `<meta:user-defined>` определяет дополнительные метаданные документа, определяемые пользователем. Каждый из таких элементов может содержать одну часть метаданных, определяемых пользователем. Элемент состоит из:

- Атрибута `meta:name`, который определяет имя элемента метаданных.
- Необязательного атрибута `meta:value-type`, определяющего тип элемента метаданных. Допустимы метатипы `float`, `date`, `time`, `boolean` и `string` (см. также раздел 6.7.1).
- Значения элементов метаданных задаются в формате, описанном в разделе 6.7.1, как значения атрибутов `office:value` для различных типов данных.

Типом по умолчанию элементов метаданных является строка (string).

```
<define name="office-meta-data" combine="choice">
    <element name="meta:user-defined">
        <attribute name="meta:name">
            <ref name="string"/>
        </attribute>
        <choice>
            <group>
                <attribute name="meta:value-type">
                    <value>float</value>
                </attribute>
                <ref name="double"/>
            </group>
            <group>
                <attribute name="meta:value-type">
                    <value>date</value>
                </attribute>
                <ref name="dateOrDateTime"/>
            </group>
            <group>
                <attribute name="meta:value-type">
                    <value>time</value>
                </attribute>
                <ref name="duration"/>
            </group>
            <group>
                <attribute name="meta:value-type">
                    <value>boolean</value>
                </attribute>
                <ref name="boolean"/>
            </group>
            <group>
```

```
                <attribute name="meta:value-type">
                    <value>string</value>
                </attribute>
                <ref name="string"/>
            </group>
            <text/>
        </choice>
    </element>
</define>
```

3.3 Специфические метаданные

Помимо определяемых пользователем элементов метаданных приложениям следует также сохранять любое дополнительное содержимое, найденное внутри элемента `<office:meta>`. Так как не существует специфической семантики для подобного внешнего содержимого, приложению не обязательно обрабатывать данную информацию, ее лишь следует сохранять при редактировании документа.

4 Текстовое содержимое

4.1 Заголовки, абзацы и основная текстовая структура

В этом разделе описаны элементы XML и атрибуты, которые используются для представления заголовков и абзацев в текстовом документе.

Элементы `<text:h>` и `<text:p>` представляют заголовки и абзацы, соответственно, вместе упоминаются как **элементы абзаца**. Все текстовое содержимое в файле OpenDocument должно содержаться в любом из этих элементов.

4.1.1 Заголовки

Заголовки определяют структуру разделов в документе. Раздел или подраздел начинается с заголовка и продолжается до следующего заголовка такого же или более высокого уровня.

```
<define name="text-h">
    <element name="text:h">
        <ref name="heading-attrs"/>
        <ref name="paragraph-attrs"/>
        <optional>
            <ref name="text-number"/>
        </optional>
        <zeroOrMore>
            <ref name="paragraph-content"/>
        </zeroOrMore>
    </element>
</define>
```

Уровень заголовка

Атрибут `text:outline-level` связан с элементом заголовка и определяет уровень заголовка, начиная с 1. Уровень заголовка без данного атрибута принимается равным 1.

```
<define name="heading-attrs" combine="interleave">
    <attribute name="text:outline-level">
        <ref name="positiveInteger"/>
    </attribute>
</define>
```

Нумерация заголовков

Нумерация заголовка может быть изменена за счет использования дополнительных атрибутов, подобных пунктам списка (см. раздел 4.3.2). Нумерация заголовков может быть запущена сначала установкой значения `true` для атрибута `text:restart-numbering`.

```
<define name="heading-attrs" combine="interleave">
    <optional>
        <attribute name="text:restart-numbering" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Стартовое значение

Атрибут `text:start-value` может быть использован для возобновления нумерации заголовков на текущем уровне при помощи установки нового значения нумерации.

```
<define name="heading-attrs" combine="interleave">
    <optional>
        <attribute name="text:start-value">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Подавление нумерации заголовка

Иногда необходим специальный заголовок, который не должен быть пронумерован. Это соответствует ненумерованным заголовкам в списках (см. раздел 4.3). Чтобы достичь этого, можно использовать дополнительный атрибут `text:is-list-header`. Если его значение `true`, то данный заголовок не будет пронумерован, даже если будет явно указан `list-style`.

```
<define name="heading-attrs" combine="interleave">
    <optional>
        <attribute name="text:is-list-header" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Форматированная нумерация заголовков

Если заголовок пронумерован, текст форматированного номера может быть добавлен в элемент `<text:number>`. Этот текст может использовать приложение,

не поддерживающее нумерацию заголовков, но он будет проигнорирован, если такая поддержка присутствует.

```
<define name="text-number">
    <element name="text:number">
        <ref name="string"/>
    </element>
</define>
```

4.1.2 Абзацы

Абзацы — основные единицы текста.

```
<define name="text-p">
    <element name="text:p">
        <ref name="paragraph-attrs"/>
        <zeroOrMore>
            <ref name="paragraph-content"/>
        </zeroOrMore>
    </element>
</define>
```

4.1.3 Общие атрибуты элементов абзаца

Элементы абзаца имеют следующие атрибуты: `text:style-name`, `text:class-names` и `text:cond-style-name`. Эти атрибуты должны ссылаться на стиль абзаца.

Атрибут `text:style-name` указывает на стиль абзаца, в то время как атрибут `text:cond-style-name` ссылается на условный стиль, то есть, стиль который содержит указатели на другие стили (см. раздел 14.1.1). Если к абзацу применен условный стиль, то атрибут `text:style-name` содержит имя стиля, который стал результатом условной оценки, в то время как само имя условного стиля является значением атрибута `text:cond-style-name`. Такая структура XML упрощает преобразование [XSLT], поскольку XSLT должен воспользоваться только условным стилем, если атрибуты форматирования корректны. Указанный стиль может быть общим или автоматическим.

Атрибут `text:class-names` содержит список имен стилей в абзаце, разделенных пробелами. Указанные стили применяются в порядке, в котором они содержатся в списке. Если указаны атрибуты `text:style-name` и `text:class-names`, то стиль, на который ссылается атрибут `text:style-name`, будет в `text:class-names` как первый стиль в списке. Если условный стиль определен совместно с атрибутом `style:class-names`, но без атрибута `text:style-name`,

тогда первый стиль в списке стилей используется как значение пропущенного атрибута `text:style-name`.

Соответствующие спецификации приложения должны поддерживать атрибут `text:class-names`, а также сохранять их в процессе редактирования.

```
<define name="paragraph-attrs">
    <optional>
        <attribute name="text:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:class-names">
            <ref name="styleNameRefs"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:cond-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Пример — Стили и условные стили:

```
<text:p text:style-name="Heading 1">
"Heading 1" is not a conditional style.
</text:p>
<text:p text:style-name="Numbering 1" text:cond-style-name="Text body">
"Text body" is a conditional style. If it is contained in a numbered
paragraph, it maps to "Numbering 1". This is assumed in this example.
</text:p>
```

Абзац может иметь идентификатор, который может быть использован для ссылки на абзац из других элементов.

```
<define name="paragraph-attrs" combine="interleave">
    <optional>
        <ref name="text-id"/>
    </optional>
</define>
```

4.2 Последовательности страниц

Элемент последовательности страниц `<text:page-sequence>` определяет последовательность мастер-страниц, которые представлены в том же порядке, в каком на них имеются ссылки в последовательности страниц. Если текстовый документ содержит последовательность страниц, то он будет состоять из точно такого количества страниц, какое определено. Документ с последовательностью страниц не имеет основного потока текста, состоящего из заголовков и абзацев, как в случае, если документ не содержит последовательности страниц. Для

документов с последовательностью страниц текстовое наполнение заключается внутри текстовых блоков. Единственно допустимым содержимым другого характера являются графические объекты.

Пример — Последовательность страниц:

```
<style:automatic-style>
    <style:page-layout name="pm1">
        <!-- portrait page -->
    </style:page-layout>
    <style:page-layout name="pm2">
        <!-- landscape page -->
    </style:page-layout>
</style:automatic-style>
...
<style:master-styles>
    <style:master-page name="portrait" style:page-layout-name="pm1"/>
    <style:master-page name="landscape" style:page-layout-name="pm2"/>
</style:master-styles>
...
<office:body>
    <text:page-sequence>
        <text:page text:master-page-name="portrait"/>
        <text:page text:master-page-name="portrait"/>
        <text:page text:master-page-name="landscape"/>
        <text:page text:master-page-name="landscape"/>
        <text:page text:master-page-name="portrait"/>
    </text:page-sequence>
    <draw:frame ...>
        <draw:text-box ...>
            <text:p>Example text.</text:p>
            ...
        </draw:text-box>
    </draw:frame>
</office:body>
```

```
<define name="text-page-sequence">
    <element name="text:page-sequence">
        <oneOrMore>
            <ref name="text-page"/>
        </oneOrMore>
    </element>
</define>
```

4.2.1 Страница

Элемент `<text:page>` описывает отдельную страницу из последовательности страниц.

```
<define name="text-page">
    <element name="text:page">
        <ref name="text-page-attlist"/>
        <empty/>
    </element>
</define>
```

Имя мастер-страницы

Атрибут `text:master-page-name` определяет имя мастер-страницы, которой он присваивается.

```
<define name="text-page-attlist">
    <attribute name="text:master-page-name">
        <ref name="styleNameRef"/>
    </attribute>
</define>
```

4.3 Списки

Формат OpenDocument поддерживает структуру списков, подобную описанной в [HTML4]. Список является элементом уровня абзаца, который содержит необязательный заголовок списка, следующего за последовательностью пунктов списка. Заголовок списка, а также каждый пункт списка содержат последовательность абзацев или элементов списка. Списки могут быть вложенными.

Списки могут быть нумерованными. Нумерация возможна с произвольного номера, который может быть присвоен в каждом пункте списка. Нумерация списка может также продолжать нумерацию другого списка, что позволяет пользователю объединять несколько списков в один прерывающийся список. Отметим, что отображение нумерации зависит от использованного стиля списка.

В дополнении к этой структурной информации, списки могут иметь стили, связанные с ними, которые содержат информацию о разметке, такую как:

- тип метки пункта списка, такой как маркер или номер;
- ширина метки пункта списка и отступ;
- символ маркера или изображения (если есть);
- формат номера для нумерованных списков (если есть);
- отступ абзаца для пунктов списка.

4.3.1 Блок списка

Списки описываются элементом `<text:list>`. Он содержит необязательный заголовок списка со следующим за ним произвольным числом пунктов списка.

Каждый список имеет свой *уровень (list level)*, который определяется вложенностью элементов `<text:list>`. Если список не находится внутри другого

списка, то его уровень равен 1. Если список находится внутри другого списка, его уровень равен уровню списка, в котором он содержится, увеличенному на единицу. Если список содержится внутри ячейки таблицы или текстового блока, его уровень равен единице даже в том случае, когда таблица или текстовый блок сами являются вложенными в другой список.

С элементом списка могут быть связаны следующие атрибуты:

- имя стиля;
- продолжение нумерации.

```
<define name="text-list">
    <element name="text:list">
        <ref name="text-list-attr"/>
        <optional>
            <ref name="text-list-header"/>
        </optional>
        <zeroOrMore>
            <ref name="text-list-item"/>
        </zeroOrMore>
    </element>
</define>
```

Имя стиля

Необязательный атрибут `text:style-name` определяет имя стиля списка, который применяется для списка.

Если данный атрибут пропущен, следовательно стиль списка не определен, применяется одно из следующих действий.

- Если список содержится внутри другого, стилем списка по умолчанию является стиль внешнего списка.
- Если для внешнего списка не определен стиль списка, но список содержит абзацы, у которых описан стиль абзаца с определенным стилем списка, этот стиль списка используется для любого из этих абзацев.
- Для всех остальных абзацев применяется стиль списка по умолчанию.

Для определения свойств форматирования списка принимается во внимание уровень списка и имя стиля списка (см. раздел 14.10 для получения дополнительной информации о свойствах форматирования списка).

```
<define name="text-list-attr" combine="interleave">
    <optional>
        <attribute name="text:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Продолжение нумерации

По умолчанию первый пункт списка начинается с номера, определенного в стиле списка. Атрибут продолжения нумерации может использоваться для продолжения нумерации от предшествующего списка.

Атрибут может использоваться с элементом `<text:list>` и может принимать значения `true` и `false`.

Если значение атрибута равно `true` и стиль нумерации предшествующего списка такой же, как в текущем списке, то номер первого пункта списка в текущем списке равен номеру последнего пункта предыдущего списка, увеличенному на единицу.

```
964 <define name="text-list-attr" combine="interleave">
965     <optional>
966         <attribute name="text:continue-numbering">
967             <ref name="boolean"/>
968         </attribute>
969     </optional>
970 </define>
```

4.3.2 Пункт списка

Пункты списка содержат текстовое содержимое списка. Элемент `<text:list-item>` может содержать абзацы или списки. Пункт списка не может содержать заголовки или таблицы.

```
971 <define name="text-list-item">
972     <element name="text:list-item">
973         <ref name="text-list-item-attr"/>
974         <ref name="text-list-item-content"/>
975     </element>
976 </define>
977 <define name="text-list-item-content">
978     <optional>
979         <ref name="text-number"/>
980     </optional>
981     <zeroOrMore>
982         <choice>
983             <ref name="text-p"/>
984             <ref name="text-h"/>
985             <ref name="text-list"/>
986         </choice>
987     </zeroOrMore>
988 </define>
```

Первой строке в пункте списка предшествует маркер или номер, в зависимости от стиля списка, связанного с данным списком. Если пункт списка непосредственно начинает другой список и не содержит никакого текста, маркер или номер не отображается.

С элементом `<text:list-item>` может быть связан только следующий атрибут:

- начальное значение.

Начальное значение

Нумерация текущего списка может быть начата заново с указанного номера. Атрибут `text:start-value` используется для определения начального номера списка.

Этот атрибут может быть применен только к пунктам с нумерованным типом списка. Он начинает нумерацию списка заново в текущем пункте.

```
<define name="text-list-item-attr" combine="interleave">
    <optional>
        <attribute name="text:start-value">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Форматированный номер

Если к пункту списка применяется нумерация, то текст форматированного номера может быть включен в элемент `<text:number>`. Этот текст может быть использован приложениями, которые не поддерживают нумерацию, но он не будет рассматриваться приложением, которое нумерацию поддерживает (см. также раздел 4.1.1).

Пример — Списки и вложенные списки:

```
<text:list text:style-name="List 1">
    <text:list-item>
    <text:p>This is the first list item</text:p>
    <text:p>This is a continuation of the first list item.</text:p>
    </text:list-item>
    <text:list-item>
    <text:p>This is the second list item.
            It contains a sub list.</text:p>
    <text:list>
        <text:list-item><text:p>This is a sub list item.</text:p>
        <text:list-item><text:p>This is a sub list item.</text:p>
        <text:list-item><text:p>This is a sub list item.</text:p>
    </text:list>
    </text:list-item>
    <text:list-item>
    <text:p>This is the third list item</text:p>
    </text:list-item>
</text:list>
```

4.3.3 Заголовок списка

Заголовок списка — это специальный вид пункта списка. Он содержит один или несколько абзацев, которые отображаются перед списком. Абзацы отображаются как пункты списка, но без предшествующего номера или маркера. Заголовок списка представляется элементом заголовок списка.

```
<define name="text-list-header">
    <element name="text:list-header">
        <ref name="text-list-item-content"/>
    </element>
</define>
```

4.3.4 Нумерованные абзацы

В некоторых случаях желательно определять список не как структурный элемент, состоящий из нескольких пунктов списка, а определять его на уровне целых абзацев, установив, нумерован ли абзац и на каком уровне. Для обеспечения этого используется элемент `<text:numbered-paragraph>`, который позволяет осуществлять нумерацию отдельных абзацев, как если бы они являлись частью списка на определенном уровне.

Нумерованные абзацы могут использовать те же самые свойства сквозной нумерации, что и пункты списков, таким образом формируется похожий, альтернативный способ определения списков. Список, представленный в `<text:list>`, может быть преобразован в список, представленный в `<text:numbered-paragraph>`, и наоборот.

```
<define name="text-numbered-paragraph">
    <element name="text:numbered-paragraph">
        <ref name="text-numbered-paragraph-attr"/>
        <optional>
            <ref name="text-number"/>
        </optional>
        <choice>
            <ref name="text-p"/>
            <ref name="text-h"/>
        </choice>
    </element>
</define>
```

Нумерованному абзацу может быть назначен уровень списка. Нумерованный абзац эквивалентен вложенному списку представленного уровня, содержащего один пункт списка с одним абзацем. Если уровень не назначен, нумерованный абзац интерпретируется как находящийся на первом уровне.

```
<define name="text-numbered-paragraph-attr" combine="interleave">
```

```
    <optional>
        <attribute name="text:level" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Так как нумерованный абзац совмещает функциональность списка (возможно вложенного) с одним пунктом списка, он также может использовать атрибуты этого элемента.

```
<define name="text-numbered-paragraph-attr" combine="interleave">
    <ref name="text-list-attr"/>
</define>
<define name="text-numbered-paragraph-attr" combine="interleave">
    <ref name="text-list-item-attr"/>
</define>
```

Текст форматированного номера может быть включен в элемент `<text:number>`. Этот текст может быть использован приложением, которое не поддерживает нумерацию, но он не будет использоваться, если нумерация поддерживается приложением (см. также раздел 4.1.1).

4.4 Текстовые разделы

Текстовый раздел — это именованная область текстового содержимого уровня абзаца. Раздел начинается и заканчивается на границах абзаца и может содержать любое количество абзацев.

В формате OpenDocument есть два способа применения разделов: для назначения определенных свойств оформления области текста; .для группировки текста, который получается автоматически из некоторого внешнего источника данных.

В дополнении разделы могут содержать обычное текстовое содержимое или текст может содержаться в другом файле, связанном с разделом. Раздел также может иметь защиту от записи или быть скрытым.

Раздел может иметь параметры текстовых колонок, цвета фона или образца и настройку сносок. Параметры настройки формируют стиль раздела, который представлен в элементе `<style:style>` (см. раздел 14.8.3).

Свойства форматирования для разделов объясняются в разделе 15.7.

Разделы поддерживают два способа связи с внешним содержимым. Если раздел связан с другим документом, связь может быть организована одним из следующих способов.

- Источник, определяемый XLink, задается элементом `text:section-source`.
- Динамический обмен данными (Dynamic Data Exchange, DDE) задается элементом `office:dde-source`.

Ссылочная информация на внешнее содержимое располагается в первом потомке элемента раздела. Раздел, который ссылается на внешнее содержимое, содержит полное представление источника данных, так что приложения нуждаются в понимании информации о связях, только если они предполагают обновить содержимое раздела.

```
<define name="text-section">
    <element name="text:section">
        <ref name="text-section-attr"/>
        <choice>
            <ref name="text-section-source"/>
            <ref name="text-section-source-dde"/>
            <empty/>
        </choice>
        <zeroOrMore>
            <ref name="text-content"/>
        </zeroOrMore>
    </element>
</define>
```

Примечание — Пункты списка не могут содержать разделов. Таким образом, списки могут только полностью содержаться внутри элементов раздела. Если решено достичь эффекта наложения списков и разделов или разделы содержатся внутри списков, то списки должны быть разбиты на несколько списков, каждый из которых находился бы полностью внутри раздела. Разбивая список, атрибуты сквозной нумерации следует установить такими, чтобы отображение и поведение их совпадало с исходным списком, не прерванным разделами.

4.4.1 Атрибуты раздела

Текстовые индексы, описанные в разделе 7, можно считать специальным видом текстового раздела, поскольку они разделяют такую же самую общую структуру, а также определенные атрибуты. Они собраны в следующем определении:

```
<define name="text-section-attr" combine="interleave">
    <ref name="sectionAttr"/>
```

```
</define>
```

Остальные атрибуты в этом разделе относятся к элементу `<text:section>`.

Стиль раздела

Атрибут `text:style-name` ссылается на стиль раздела.

```
<define name="sectionAttr" combine="interleave">
    <optional>
        <attribute name="text:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Имя раздела

Каждый раздел должен иметь уникальное имя, которое определяет его однозначно. Атрибут `text:name` содержит имя раздела.

```
<define name="sectionAttr" combine="interleave">
    <attribute name="text:name">
        <ref name="string"/>
    </attribute>
</define>
```

Защищенные разделы

Раздел может быть защищен, это означает, что пользователь не может редактировать его. Атрибут `text:protected` показывает, защищен ли раздел. Пользовательский интерфейс должен реализовывать защиту, если атрибут установлен.

```
<define name="sectionAttr" combine="interleave">
    <optional>
        <attribute name="text:protected">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Средствами пользовательского интерфейса можно сбросить флаг защиты, если раздел не защищен паролем. В этом случае пользователь должен знать пароль, чтобы сбросить флаг защиты. Атрибут `text:protection-key` определяет пароль защиты раздела. Для предотвращения сохранения пароля непосредственно в XML-файле, сохраняется только хеш-значение пароля.

```
<define name="sectionAttr" combine="interleave">
    <optional>
```

```
        <attribute name="text:protection-key">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Скрытые и условные разделы

Разделы могут быть скрыты при выполнении определенного условия или скрыты безусловно.

Атрибут `text:display` определяет, скрыт раздел или нет. Этот атрибут может принимать следующие значения:

- `true` — раздел отображается (значение по умолчанию);
- `none` — раздел скрыт безусловно;
- `condition` — раздел скрыт по условию, определенному в атрибуте `text:condition`.

Атрибут `text:condition` определяет условие, по которому раздел становится скрытым. Условие представляется в виде строки. Если значение атрибута `text:display` равно `condition`, атрибут `text:condition` должен быть определен.

```
<define name="text-section-attr" combine="interleave">
    <choice>
        <attribute name="text:display">
            <choice>
                <value>true</value>
                <value>none</value>
            </choice>
        </attribute>
        <group>
            <attribute name="text:display">
                <value>condition</value>
            </attribute>
            <attribute name="text:condition">
                <ref name="string"/>
            </attribute>
        </group>
        <empty/>
    </choice>
</define>
```

4.4.2 Источник раздела

Элемент `<text:section-source>` указывает на то, что включенный раздел является связанным. Если этот элемент используется, то он должен быть первым элементом в элементе `<text:section>`.

```
<define name="text-section-source">
```

```
    <element name="text:section-source">
        <ref name="text-section-source-attr"/>
    </element>
</define>
```

С элементом `<text:section-source>` могут быть связаны следующие атрибуты:

- URL источника раздела;
- имя связанного раздела;
- имя фильтра.

URL источника раздела

Этот атрибут определяет документ или раздел, с которым связан данный раздел. Имя целевого раздела устанавливается локальной частью URL, следующей после знака диез. Подразумевается использование атрибута `xlink:href`, потому что элементы `<text:section-source>` могут также ссылаться на внутренние разделы.

```
<define name="text-section-source-attr" combine="interleave">
    <optional>
        <attribute name="xlink:href">
            <ref name="anyURI"/>
        </attribute>
        <optional>
            <attribute name="xlink:type" a:defaultValue="simple">
                <value>simple</value>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:show" a:defaultValue="embed">
                <value>embed</value>
            </attribute>
        </optional>
    </optional>
</define>
```

Имя связанного раздела

Если целевой объект ссылки является разделом документа, атрибут `text:section` содержит имя целевого раздела. Если атрибут не задан, целевым объектом ссылки является документ целиком.

```
<define name="text-section-source-attr" combine="interleave">
    <optional>
        <attribute name="text:section-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Имя фильтра

Атрибут `text:filter-name` определяет, какой фильтр будет использован при импорте целевого объекта ссылки. Значение этого атрибута обуславливается реализацией.

```
<define name="text-section-source-attr" combine="interleave">
    <optional>
        <attribute name="text:filter-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

4.4.3 Источник DDE

Если разделы связаны по DDE, то информация об их присоединении представлена в элементах `<office:dde-source>`. Он содержит атрибуты, которые определяют приложение, тему и пункт DDE-соединения. Отметим это, потому что раздел содержит XML-интерпретацию содержимого DDE-ссылки, эта информация нужна только в том случае, если желательно обновление данных по DDE.

```
<define name="text-section-source-dde">
    <ref name="office-dde-source"/>
</define>
```

4.5 Графическое наполнение, привязанное к странице

В пределах текстовых документов изображения, вложенные объекты и другие графические объекты представляются на уровне абзаца, если они привязаны к странице, а не к абзацу или символьной позиции внутри абзаца (см. раздел 9.2 для получения детальной информации о графических объектах и раздел 9.2.16 для информации по их привязке).

4.6 Отслеживание изменений

В этом разделе описано, как могут быть представлены изменения в документе.

4.6.1 Отслеженные изменения

Все отслеженные изменения в текстовом документе сохраняются в списке. Список содержит элементы для каждого произошедшего изменения документа. Если элемент `<text:tracked-changes>` отсутствует, значит отслеживание изменений выключено.

```
<define name="text-tracked-changes">
    <optional>
        <element name="text:tracked-changes">
            <ref name="text-tracked-changes-attr"/>
            <zeroOrMore>
                <ref name="text-changed-region"/>
            </zeroOrMore>
        </element>
    </optional>
</define>
```

Слежение за изменениями

Этот атрибут определяет, должны ли пользовательские приложения отслеживать и записывать изменения в данном документе.

```
<define name="text-tracked-changes-attr" combine="interleave">
    <optional>
        <attribute name="text:track-changes" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

4.6.2 Измененные области

Для каждой измененной области документа есть только одна запись в списке отслеженных изменений. Эта запись содержит список всех изменений, которые были сделаны в этой области. Начало и конец данной области обозначены с помощью элементов начала и конца, которые описаны ниже.

```
<define name="text-changed-region">
    <element name="text:changed-region">
        <ref name="text-changed-region-attr"/>
        <ref name="text-changed-region-content"/>
    </element>
</define>
```

Идентификатор изменения

Каждый элемент имеет идентификатор. Элементы, обозначающие начало и конец области, используют этот идентификатор определения области, к которой они относятся.

```
<define name="text-changed-region-attr" combine="interleave">
    <attribute name="text:id">
        <ref name="ID"/>
    </attribute>
</define>
```

4.6.3 Вставка

Элемент `<text:insertion>` содержит информацию, необходимую для определения вставки в содержимое. Содержимое может быть частью текста внутри абзаца, целым абзацем или целой таблицей. Вставленное содержимое — непосредственно часть текстового документа, отмеченная элементами начала и конца изменения.

```
<define name="text-changed-region-content" combine="choice">
    <element name="text:insertion">
        <ref name="office-change-info"/>
    </element>
</define>
```

Пример – Вставка текста:

```
<text:tracked-changes>
    <text:changed-region text:id="c001">
        <text:insertion>
            <office:change-info>
                <dc:creator>Michael Brauer</dc:creator>
                <dc:date>1999-05-18T12:56:04</dc:date>
            </office:change-info>
        </text:insertion>
    </text:changed-region>
</text:tracked-changes>

<text:p>
    This is the original text<text:change-start text:change-id="c001"/>,
    but this has been added</text:change-end text:change-id="c001"/>.
</text:p>
```

4.6.4 Удаление

Элемент `<text:deletion>` содержит содержимое, которое было удалено при включенном слежении за изменениями. Позиция, где текст был удален, обозначена с помощью элемента позиции изменения.

Если часть абзаца была удалена, то удаленный текст содержится в этом элементе как элемент абзаца. Если удаленный текст был повторно вставлен в документ, абзац объединяется с абзацем где было удаление.

```
<define name="text-changed-region-content" combine="choice">
    <element name="text:deletion">
        <ref name="office-change-info"/>
        <zeroOrMore>
```

```
            <ref name="text-content"/>
        </zeroOrMore>
    </element>
</define>
```

Пример – Удаление текста:

```
<text:tracked-changes>
    <text:changed-region text:id="c002">
        <text:deletion>
            <office:change-info>
                <dc:creator>Michael Brauer</dc:creator>
                <dc:date>1999-05-18T12:56:04</dc:date>
            </office:change-info>
            <text:p>, but this has been deleted</text:p>
        </text:deletion>
    </text:changed-region>
</text:tracked-changes>

<text:p>
    This is the original text<text:change text:region-id="c002"/>.
</text:p>
```

Этот пример показывает следующее:

- текст, который был удален = `, but this has been deleted`, содержится в элементе `<text:p>` внутри элемента `<text:deletion>`;
- текущий текст = `This is the original text` содержится в элементе `<text:p>` в конце примера;
- до удаления текст был следующим = `This is the original text, but this has been deleted.`

Отметим, что удаленный текст, как и весь текст в формате OpenDocument, содержится в элементе абзаца. Для восстановления изначального текста, абзац объединяется с его окружением. Другими словами, удаленный фрагмент, состоящий из одного слова, будет представлен в виде абзаца, содержащего одно слово.

Для восстановления текста до момента удаления сделайте следующее:

- если метка изменения находится внутри абзаца, вставьте текстовое содержимое элемента `<text:deletion>`, как будто начальный `<text:p>` и конечный `</text:p>` теги пропущены;
- если метка изменения находится внутри заголовка, поступайте как описано выше, но приспосабливая вставляемые теги под особенности их нового прототипа;
- иначе просто скопируйте содержимое элемента `<text:deletion>` в место метки изменения.

Пример –

Даны следующие изменения:

```
<text:changed-region text:id="example">
    <text:deletion>
        <office:change-info>...</office:change-info>
        <text:p>Hello</text:p>
        <text:p>World!</text:p>
    </text:deletion>
</text:changed-region>
```

Первый (и самый обычный) случай – если метка изменения находится внутри обычного абзаца:

```
<text:p>abc<text:change text:id="example/>def</text:p>
```

Для восстановления изначального текста, два элемента `<text:p>` копируются взамен метки изменения, за исключением тех случаев, когда теги начала и конца отсутствуют:

```
<text:p>abcHello</text:p>
<text:p>World!def</text:p>
```

Второй случай – метка изменения встречается внутри заголовка, следует такая же процедура, только скопированные теги адаптируются, чтобы быть уверенными что мы все еще имеем правильно сформированный XML:

```
<text:h>abc<text:change text:id="example/>def</text:h>
```

становится:

```
<text:h>abcHello</text:h>
<text:h>World!def</text:h>
```

Третий случай – изменения встречаются вне абзаца. При этом удаленный текст просто копируется:

```
<text:p>abcdef</text:p>
<text:change text:id="example/>
<text:p>ghijkl</text:p>
```

становится:

```
<text:p>abcdef</text:p>
<text:h>Hello</text:h>
<text:h>World!</text:h>
<text:p>ghijkl</text:p>
```

Если бы в первых двух случаях удаленный фрагмент содержал целый абзац, то дополнительные пустые абзацы должны были бы быть вставлены в элемент `<text:deletion>` для достижения желаемого результата.

Изменение от

```
<text:p>abc</text:p>
<text:h>Hello</text:h>
```

```
<text:h>World!</text:h>
<text:p>def</text:p>
```

до

```
<text:p>abc<text:change text:id="example/>def</text:p>
```

будет восстановлено так:

```
<text:changed-region text:id="example">
    <text:deletion>
        <office:change-info>...</office:change-info>
        <text:p/>
        <text:p>Hello</text:p>
        <text:p>World!</text:p>
        <text:p/>
    </text:deletion>
</text:changed-region>
```

4.6.5 Изменение формата

Элемент изменения формата отражает любые изменения в атрибутах форматирования. Область, в которой произошли изменения, отмечается элементами начала и конца изменения.

```
<define name="text-changed-region-content" combine="choice">
    <element name="text:format-change">
        <ref name="office-change-info"/>
    </element>
</define>
```

Примечание – элемент изменения формата не содержит фактических изменений.

4.6.6 Информация об изменениях

Информация об изменениях содержит метаинформацию о том, кто сделал изменения и когда. Она также используется для электронных таблиц, в соответствии с описанием в разделе 12.3 (Метаданные отслеживания изменений).

4.6.7 Метки изменения

Три элемента отмечают начало и конец области изменений.

- Элемент начала изменений – `<text:change-start>`.

  Этот элемент отмечает начало области с содержимым, где текст был вставлен или был изменен формат.

- Элемент окончания изменений – `<text:change-end>`.

  Этот элемент отмечает конец области с содержимым, где текст был вставлен или был изменен формат.

- Элемент позиции изменений – `<text:change>`.

  Это элемент отмечает позицию в пустой области, где текст был удален.

Все три элемента имеют атрибуты, определяющие идентификатор области, к которой они относятся.

```
<define name="change-marks">
    <choice>
        <element name="text:change">
            <ref name="change-mark-attr"/>
        </element>
        <element name="text:change-start">
            <ref name="change-mark-attr"/>
        </element>
        <element name="text:change-end">
            <ref name="change-mark-attr"/>
        </element>
    </choice>
</define>
<define name="change-mark-attr">
    <attribute name="text:change-id">
        <ref name="IDREF"/>
    </attribute>
</define>
```

4.7 Текстовые объявления

Некоторые текстовые элементы нуждаются в предварительном объявлении прежде, чем они могут быть использованы. Например, переменные поля требуют, чтобы используемые переменные были объявлены в начале документа. Эти объявления собраны в начале текстового документа. Такие объявления необязательны. Детальное описание таких объявлений доступно в соответствующем разделе.

Поддерживаются следующие текстовые объявления:

- объявление переменной — эти объявления используются для переменных полей (см. раздел 6.3.1);
- объявление поля пользователя — эти объявления используются для полей определяемых пользователем (см. раздел 6.3.5);
- объявление последовательности — эти объявления используются для последовательных полей (см. раздел 6.3.8);
- DDE-ссылки — эти объявления используются для DDE-полей и DDE-разделов (см. разделы 6.6.9 и 4.4.3, соответственно);
- файл с автоматическими метками — это объявление используется для формирования алфавитных указателей (см. раздел 7.8.2).

```
<define name="text-decls">
    <optional>
        <element name="text:variable-decls">
            <zeroOrMore>
                <ref name="text-variable-decl"/>
            </zeroOrMore>
        </element>
    </optional>
    <optional>
        <element name="text:sequence-decls">
            <zeroOrMore>
                <ref name="text-sequence-decl"/>
            </zeroOrMore>
        </element>
    </optional>
    <optional>
        <element name="text:user-field-decls">
            <zeroOrMore>
                <ref name="text-user-field-decl"/>
            </zeroOrMore>
        </element>
    </optional>
    <optional>
        <element name="text:dde-connection-decls">
            <zeroOrMore>
                <ref name="text-dde-connection-decl"/>
            </zeroOrMore>
        </element>
    </optional>
    <optional>
        <ref name="text-alphabetical-index-auto-mark-file"/>
    </optional>
</define>
```

5 Содержимое элементов абзаца

5.1 Основное текстовое содержимое

Потомки элементов абзаца составляют текстовое содержимое любого документа. Весь текст в элементах абзаца или их потомках — текстовое содержимое, за некоторыми исключениями, описанными ниже. Это значительно упрощает преобразование в другие форматы, так как при преобразовании могут игнорироваться любые дочерние элементы элемента абзаца и обрабатываться только их текстовое содержимое, при этом получается правильное отображение текстового содержимого.

Элементами текстового содержимого, не содержащими внутренних текстовых дочерних элементов, являются:

- Обычные и концевые сноски (см. раздел 5.3). Обычные и концевые сноски содержат текстовое содержимое, но обычно отображаются вне основного текста, например в конце страницы или документа.
- Транскрипции (см. раздел 5.4). Тексты транскрипции обычно отображаются над или под основным текстом.
- Аннотации (см. раздел 5.5). Аннотации обычно не отображаются.

```
<define name="paragraph-content" combine="choice">
    <text/>
</define>
```

5.1.1 Пробельные символы

Если элемент абзаца или любой его дочерний элемент содержит пробельные символы, то они объединяются; другими словами, они обрабатываются тем же путем, как их обрабатывает [HTML4]. Следующие символы кодовой таблицы [UNICODE] приводятся к символу ПРОБЕЛ:

- ГОРИЗОНТАЛЬНАЯ ТАБУЛЯЦИЯ (0x0009)
- ВОЗВРАТ КАРЕТКИ (0x000D)
- ПЕРЕВОД СТРОКИ (0x000A)
- ПРОБЕЛ (0x0020)

Кроме того, эти символы игнорируются, если предыдущий символ является пробельным. Предыдущий символ может находится в таком же элементе, в родительском элементе или предыдущем элементе того же уровня при условии, что он находится в том же элементе абзаца. В этом случае пробельные символы обрабатываются так, как описано выше.

Обработка пробельных символов имеет место в пределах следующих элементов:

- `<text:p>`
- `<text:h>`
- `<text:span>`
- `<text:a>`
- `<text:ref-point>`
- `<text:ref-point-start>`
- `<text:ref-point-end>`
- `<text:bookmark>`
- `<text:bookmark-start>`
- `<text:bookmark-end>`

Примечание — В [XSL] обработку пробельных символов текста абзаца можно активировать за счет добавления атрибута `fo:white-space="collapse"` элементу `<fo:block>`, который соответствует элементу абзаца.

Символ пробела

Обычно подряд идущие пробельные символы в абзаце сокращаются. Поэтому в XML есть специальный элемент используемый для отображения символов ПРОБЕЛ (0x0020) кодовой таблицы [UNICODE].

Этот элемент использует необязательный атрибут `text:c` для определения числа символов ПРОБЕЛ, которое представляет элемент. Если атрибут `text:c` пропущен, то считается что символ ПРОБЕЛ один.

Элемент требуется для представления второго и всех последующих символов ПРОБЕЛ в последовательности символов ПРОБЕЛ. Не считается ошибкой, если символ, предшествующий элементу, будет не пробельный символ, но хорошей практикой считается использование данного элемента для второго и

всех последующих символов ПРОБЕЛ в последовательности. Таким образом, приложение распознает один пробельный символ без распознания данного элемента.

```
<define name="paragraph-content" combine="choice">
    <element name="text:s">
        <optional>
            <attribute name="text:c">
                <ref name="nonNegativeInteger"/>
            </attribute>
        </optional>
    </element>
</define>
```

Символ табуляции

Элемент `<text:tab>` представляет символ табуляции в кодовой таблице [UNICODE] ГОРИЗОНТАЛЬНАЯ ТАБУЛЯЦИЯ (0x0009) в заголовках и абзацах. Элемент `<text:tab>` резервирует пространство от текущей позиции до следующего табулостопа, как определено в стиле абзаца.

```
<define name="paragraph-content" combine="choice">
    <element name="text:tab">
        <ref name="text-tab-attr"/>
    </element>
</define>
```

Для определения того, на какой табулостоп переместится символ табуляции, необходима информация о разметке. Чтобы не ориентированные на разметку приложения могли проще определять эту информацию, приложения могут формировать атрибут `text:tab-ref`, указывающий на связь символа табуляции с табулостопом в текущем стиле абзаца. Он содержит количество табулостопов на которые ссылается символ табуляции. Нулевая позиция имеет особое значение и указывает на начальный отступ абзаца.

```
<define name="text-tab-attr">
    <optional>
        <attribute name="text:tab-ref">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Примечание — Атрибут `text:tab-ref` является только подсказкой для помощи не ориентированных на разметку приложениям с целью определения связи символов табуляции с табулостопом. Ориентированные на разметку приложения должны определить позиции табуляции, исключительно основываясь на информации о стилях.

Разрывы строки

Атрибут `<text:line-break>` определяет разрыв строки в заголовке или абзаце.

```
<define name="paragraph-content" combine="choice">
    <element name="text:line-break">
        <empty/>
    </element>
</define>
```

5.1.2 Мягкие переносы, переносы и неразрывные пробелы

Мягкие переносы, переносы и неразрывные пробелы в кодовой таблице [UNICODE] представлены следующими символами:

Символ в кодовой таблице [UNICODE]	*Представление*
SOFT HYPHEN (00AD)	Мягкий перенос
NON-BREAKING HYPHEN (2011)	Неразрывный дефис
NO-BREAK SPACE (00A0)	Неразрывный пробел

5.1.3 Текст с присвоенными атрибутами

Элемент `<text:span>` представляет части текста, которым присвоены атрибуты, с использованием определенного текстового стиля или класса. Содержимым этого элемента является текст, использующий текстовый стиль.

Имя текстового стиля или текстового класса записываются в атрибуты `text:style-name` или `text:class-names`, соответственно, прикрепленные к элементу `<text:span>`. Эти атрибуты должны ссылаться на текстовые стили или классы.

Атрибут `text:style-name` описывает один текстовый стиль. Атрибут `text:class-names` содержит список имен текстовых стилей, разделенных пробелами. Указанные текстовые стили применяются в той последовательности, в которой они содержатся в списке. Если представлены атрибуты `text:style-name` и `text:class-names`, стиль, на который ссылается атрибут `text:style-name`, обрабатывается как первый стиль в списке `text:class-names`. Соответствующему спецификации приложению следует поддерживать атрибут `text:class-names`, а также сохранять его при редактировании.

Элементы `<text:span>` могут быть вложенными.

Пробельные символы, содержащиеся в этом элементе, объединяются.

```
<define name="paragraph-content" combine="choice">
    <element name="text:span">
        <optional>
            <attribute name="text:style-name">
                <ref name="styleNameRef"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="text:class-names">
                <ref name="styleNameRefs"/>
            </attribute>
        </optional>
        <zeroOrMore>
            <ref name="paragraph-content"/>
        </zeroOrMore>
    </element>
</define>
```

Пример – Текстовый стиль в документах OpenDocument:

```
<text:p>
    The last word of this sentence is
    <text:span text:style-name="emphasize">emphasized</text:span>.
</text:p>
```

5.1.4 Гиперссылки

Гиперссылки в текстовом документе описываются элементом `<text:a>`.

Этот элемент также содержит элемент таблицы событий `<office:event-listeners>`, который содержит события, связанные с гиперссылками. См. раздел 12.4 для получения дополнительной информации об элементе таблицы событий.

```
<define name="paragraph-content" combine="choice">
    <element name="text:a">
        <ref name="text-a-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="paragraph-content"/>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<text:a>` могут быть связаны следующие атрибуты:

- имя;
- адрес ссылки;
- целевой фрейм;
- стили текста.

Имя

Гиперссылка может иметь имя, но оно не обязательно. Атрибут `office:name` определяет имя гиперссылки, если таковое присутствует. Это имя может служить целью для других гиперссылок.

```
<define name="text-a-attlist" combine="interleave">
    <optional>
        <attribute name="office:name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Адрес ссылки

Атрибут `xlink:href` определяет URL цели ссылки.

```
<define name="text-a-attlist" combine="interleave">
    <attribute name="xlink:href">
        <ref name="anyURI"/>
    </attribute>
    <optional>
        <attribute name="xlink:type" a:defaultValue="simple">
            <value>simple</value>
        </attribute>
    </optional>
    <optional>
        <attribute name="xlink:actuate" a:defaultValue="onRequest">
            <value>onRequest</value>
        </attribute>
    </optional>
</define>
```

Целевой фрейм

Атрибут `office:target-frame-name` определяет целевой фрейм ссылки. Этот атрибут может принимать следующие значения:

- `_self` — документ по ссылке заменяет содержимое текущего фрейма;
- `_blank` — документ по ссылке открывается в новом фрейме;
- `_parent` — документ по ссылке открывается в родительском по отношению к текущему документу фрейме;
- `_top` — документ по ссылке отображается в самом верхнем фрейме, это фрейм, который содержит текущей фрейм в качестве дочернего, но который не содержится внутри никакого другого фрейма;

- имя фрейма — документ по ссылке отображается в заданном фрейме. Если фрейм с таким именем отсутствует, будет создан новый фрейм с заданным именем.

Для соответствия спецификации XLink добавлен дополнительный атрибут `xlink:show` элементу `<text:a>`. Если значением атрибута является `_blank`, то значение атрибута `xlink:show` будет `new`. Если значением атрибута является одно из других возможных, то значение атрибута `xlink:show` будет `replace` (см. [XLink]).

```
<define name="text-a-attlist" combine="interleave">
    <optional>
        <attribute name="office:target-frame-name">
            <ref name="targetFrameName"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="xlink:show">
            <choice>
                <value>new</value>
                <value>replace</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Стили текста

Каждая ссылка имеет два стиля текста.

- Если ссылка не посещалась, то стиль текста, определенный атрибутом `text:style-name`, применяется к тексту гиперссылки.
- Если ссылка посещалась, то стиль текста, определенный атрибутом `text:visited-style-name`, применяется к тексту гиперссылки.

```
<define name="text-a-attlist" combine="interleave">
    <optional>
        <attribute name="text:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:visited-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

5.2 Закладки и перекрестные ссылки

5.2.1 Закладки

Закладки могут помечать текстовую позицию или диапазон текста. Текстовый диапазон может начинаться в любой текстовой позиции и заканчиваться в другой текстовой позиции. В частности, закладка может начинаться в середине одного абзаца, а заканчиваться в середине другого. Элемент XML, используемый для представления закладки, изменяется в зависимости от ее типа следующим образом:

- `<text:bookmark>` — для пометки одной текстовой позиции;
- `<text:bookmark-start>` — для пометки начала текстового диапазона;
- `<text:bookmark-end>` — для пометки конца текстового диапазона.

Для каждого элемента `<text:bookmark-start>` должен быть элемент `<text:bookmark-end>` в том же текстовом потоке, использующем тот же атрибут `text:name`, и наоборот. Элемент `<text:bookmark-start>` должен предшествовать элементу `<text:bookmark-end>`.

```
<define name="paragraph-content" combine="choice">
    <choice>
        <element name="text:bookmark">
            <attribute name="text:name">
                <ref name="string"/>
            </attribute>
        </element>
        <element name="text:bookmark-start">
            <attribute name="text:name">
                <ref name="string"/>
            </attribute>
        </element>
        <element name="text:bookmark-end">
            <attribute name="text:name">
                <ref name="string"/>
            </attribute>
        </element>
    </choice>
</define>
```

Пример – Закладки:

```
<text:p>
<text:bookmark text:name="Mark 1"/>There is a text mark in front of this
paragraph.
<text:bookmark-start text:name="Mark 2"/>In front of this paragraph there
is
the start of a bookmark.
</text:p>
<text:p>
This bookmark ends
```

```
<text:bookmark-end text:name="Mark 2"/>
amid this sentence.
</text:p>
```

5.2.2 Перекрестные ссылки

Представление перекрестных ссылок основано на XML-представлении закладок.

Есть два типа перекрестных ссылок.

- Перекрестная ссылка на точку, которая помечает специфическую позицию в тексте и определяется одним атрибутом `<text:reference-mark>`.
- Перекрестная ссылка на диапазон, которая помечает диапазон символов в тексте и определяется двумя элементами: `<text:reference-mark-start>` для пометки начала диапазона и `<text:reference-mark-end>` для пометки конца диапазона.

Каждая перекрестная ссылка идентифицируется по ее имени, которое должно быть уникальным. В перекрестных ссылках на диапазон начальный и конечный элементы должны использовать одинаковое имя перекрестной ссылки.

Перекрестная ссылка на точку

Элемент `<text:reference-mark>` определяет перекрестную ссылку на точку.

```
<define name="paragraph-content" combine="choice">
    <element name="text:reference-mark">
        <attribute name="text:name">
            <ref name="string"/>
        </attribute>
    </element>
</define>
```

Перекрестная ссылка на диапазон

Элементы `<text:reference-mark-start>` и `<text:reference-mark-end>` определяют перекрестную ссылку на диапазон.

```
<define name="paragraph-content" combine="choice">
    <choice>
        <element name="text:reference-mark-start">
            <attribute name="text:name">
                <ref name="string"/>
            </attribute>
        </element>
```

```
        <element name="text:reference-mark-end">
            <attribute name="text:name">
                <ref name="string"/>
            </attribute>
        </element>
    </choice>
</define>
```

В схеме OpenDocument три элемента используются для представления перекрестных ссылок вместо одного, потому что перекрестные ссылки, представленные как один элемент XML:

- не поддерживают перекрывающиеся перекрестные ссылки;
- плохо взаимодействует с другими элементами.

Рассмотрим следующий пример.

Пример – Перекрывающиеся перекрестные ссылки на диапазон:

```
<text:p>
    <text:reference-mark-start text:name="first"/>This is an
    <text:reference-mark-start text:name="second"/>example of a sentence
    <text:reference-mark-end text:name="first"/>with overlapping
references.
    <text:reference-mark-end text:name="second"/>
</text:p>
```

Пример показывает две перекрестные ссылки, которые покрывают следующий текст:

перекрестная ссылка «first»	“This is an example of a sentence”
перекрестная ссылка «second»	“example of a sentence with overlapping references.”

Эта перекрывающая структура не может быть представлена с использованием одного элемента перекрестной ссылки, чтобы содержать тот текст, на который ссылаются. Точно также перекрестные ссылки, охватывающие несколько абзацев, создают такую же ситуацию, как и два перекрывающих XML-элемента. Получается, что в пределах текста, на который организована ссылка, есть метки начала или конца, но не обе.

5.3 Сноски

Сноски состоят из элементов `<text:note>`, которые встречаются в тексте на той позиции, к которой сноска привязана. То, как сноски нумеруются и

отображаются, определяется элементом `<text:notes-configuration>`, который встречается в разделе `<office:styles>`.

5.3.1 Элемент сноски

Элемент сноски определяется как текстовые примечания, которые добавляются в определенные позиции текста. Общую реализацию концепции обычных и концевых сносок можно встретить в большинстве текстовых процессоров. Сноска содержит элемент цитирования и элементы основной части, которые формируют содержимое сносок.

В документах OpenDocument сноски представлены в виде, подобном обычным сноскам в [XSL]. В XSL первый потомок элемента сноски содержит цитату в виде элемента `<fo:inline>`. Схема OpenDocument использует такую же структуру, но вводится элемент `<text:note-citation>`. Вторая часть содержит основную часть сноски, так же как и в XSL.

В дополнение, в OpenDocument есть особенный элемент `<text:notes-configuration>`. Чтобы в конфигурации сноски достичь такого же эффекта как и в XSL, каждый элемент цитаты сноски должен быть отформатирован соответственно.

```
<define name="paragraph-content" combine="choice">
    <element name="text:note">
        <ref name="text-note-class"/>
        <optional>
            <attribute name="text:id">
                <ref name="string"/>
            </attribute>
        </optional>
        <element name="text:note-citation">
            <optional>
                <attribute name="text:label">
                    <ref name="string"/>
                </attribute>
            </optional>
            <text/>
        </element>
        <element name="text:note-body">
            <zeroOrMore>
                <ref name="text-content"/>
            </zeroOrMore>
        </element>
    </element>
</define>
```

Классы сносок

Каждая сноска принадлежит классу, который определяет особенности ее отображения. Поддерживаются два класса сносок: обычные и концевые.

```
<define name="text-note-class">
    <attribute name="text:note-class">
        <choice>
            <value>footnote</value>
            <value>endnote</value>
        </choice>
    </attribute>
</define>
```

Ссылочный идентификатор обычной сноски

Ссылочный идентификатор обычной сноски используется ссылкой на обычную сноску для идентификации обычной сноски, на которую ссылаются.

Цитатный элемент сноски

Элемент `<text:note-citation>` содержит отформатированный цитатный элемент сноски, равно как и отформатированный номер или строку.

Маркирование сноски

Цитатные элементы сноски могут быть маркированы или нумерованы. Если они нумеруются, то номер выбирается и форматируется автоматически, в соответствии с элементом конфигурации сноски. Если они маркированы, то пользователь должен задать маркер для каждой сноски, вставленной в документ. Маркер хранится в атрибуте `text:label` элемента `<text:note-citation>`.

Основная часть сноски

Элемент `<text:note-body>` содержит актуальное содержимое обычной сноски. У него нет никаких атрибутов.

Схема разрешает включение сносок в основную часть сноски. Хотя это может быть применимо для будущих типов сносок, но это не применимо для обычных и концевых сносок. Приложения, соответствующие спецификации, могут поддерживать, а могут и не поддерживать такие вложенные сноски.

Пример – Обычная сноска:

```
<text:p>
```

```
    This paragraph contains a footnote
    <text:note text:note-class="footnote" text:id="ftn001">
        <text:note-citation>1</text:note-citation>
        <text:note-body>
            <text:p>
                This footnote has a generated sequence number
            </text:p>
        </text:note-body>
    </text:note>
    .
</text:p>
<text:p>
    This paragraph contains a footnote
    <text:note text:note-class="footnote" text:id="ftn002">
        <text:note-citation text:label="*">*</text:note-citation>
        <text:note-body>
            <text:p>
                 This footnote has a fixed citation
            </text:p>
        </text:note-body>
    </text:note>
    , too
</text:p>
```

5.4 Транскрипция

Транскрипция — это дополнительный текст, который отображается выше или ниже основного текста. Назначение транскрипции — пояснять основной текст или давать дополнительную информацию о его произношении.

Есть два элемента, которые могут содержаться в элементе `<text:ruby>`:

- основа транскрипции;
- текст транскрипции.

Элемент `<text:ruby-base>` содержит текст, который комментируется. Он содержит любой элемент содержимого абзаца, например текстовые диапазоны. Атрибут `text:style-name` элемента ссылается на стиль транскрипции, который определяет форматирующие атрибуты транскрипции (см. раздел 14.8.4).

Элемент `<text:ruby-text >` содержит текст комментария. Он может содержать только простой текст. Атрибут `text:style-name` элемента ссылается на текстовый стиль, который определяет форматирующие атрибуты, используемые для текста.

```
1400 <define name="paragraph-content" combine="choice">
1401     <element name="text:ruby">
1402         <optional>
1403             <attribute name="text:style-name">
1404                 <ref name="styleNameRef"/>
1405             </attribute>
1406         </optional>
1407         <element name="text:ruby-base">
```

```
            <ref name="paragraph-content"/>
        </element>
        <element name="text:ruby-text">
            <optional>
                <attribute name="text:style-name">
                    <ref name="styleNameRef"/>
                </attribute>
            </optional>
            <text/>
        </element>
    </element>
</define>
```

5.5 Аннотации к тексту

Формат OpenDocument позволяет добавлять аннотацию к элементам абзаца. См. раздел 12.1 для получения детальной информации по аннотациям.

```
<define name="paragraph-content" combine="choice">
    <ref name="office-annotation"/>
</define>
```

5.6 Индексные метки

Индексные метки используются для пометки фрагментов с целью включения в текстовые индексы. Они имею схожую структуру с закладками и перекрестными ссылками. Детально они описаны в разделе 7.1 вместе с текстовыми индексами.

5.7 Отслеживание изменений и метки изменений

Абзацы также могут содержать метки отслеживания изменений. Они уже описаны в разделе об отслеживании изменений (см. раздел 4.6) и упоминаются в данном разделе для полноты.

```
<define name="paragraph-content" combine="choice">
    <ref name="change-marks"/>
</define>
```

5.8 Внедренные графические объекты и текстовые блоки

В пределах текстовых документов изображения, вставленные объекты и другие графические объекты могут быть привязаны к абзацу, к символу или как символ. Если они привязаны к абзацу, то отображаются в пределах абзаца в произвольном месте. Если они привязаны к символу или как символ, то они отображаются в пределах абзаца в определенной позиции, к которой они

прикреплены. См. раздел 9.2 для получения детальной информации о графических объектах и раздел 9.2.16 об их привязке.

```
<define name="paragraph-content" combine="choice">
    <choice>
        <ref name="shape"/>
        <ref name="draw-a"/>
    </choice>
</define>
```

6 Текстовые поля

Текстовые документы OpenDocument или текстовое содержимое OpenDocument, вставленное в другие типы документов, может иметь изменяемые элементы текста, называемые полями. Существует несколько различных типов полей, каждый из которых представляет определенный тип изменяемого элемента текста. Поля обычно используются для определения:

- **Номера страницы**

  Поле номера страницы отображает номер страницы, на которой оно находится. Это поле полезно для нижних колонтитулов. На каждой из страниц, на которой показан нижний колонтитул, поле присваивает себе номер текущей страницы так, что все страницы оказываются корректно пронумерованными.

- **Даты создания**

  Поле даты создания отображает дату создания текущего документа. Это поле полезно использовать в шаблонах документов. Каждый документ, созданный с использованием шаблона, содержит дату его создания.

- **Диапазона чисел**

  Поле диапазона чисел позволяет пользователю нумеровать определенные элементы, например изображения или таблицы. Поле диапазона чисел отображает его собственную позицию относительно другого поля диапазона чисел для того же диапазона. Следовательно, если изображение и связанное с ним поле диапазона чисел перемещаются внутри документа, поля автоматически обновляются, чтобы отразить новый порядок.

В данном разделе описано, как поля представляются в формате файла OpenDocument.

6.1 Общие характеристики элементов поля

Каждый тип поля представлен соответствующим типом элемента. Поле в документе закодировано как одиночный элемент подходящего типа. Содержимым

элемента является текстовое представление значения текущего поля, соответствующее тому, как оно должно быть отображено или напечатано. Следовательно, игнорирование всех элементов поля и отображение только текстового содержания элементов обеспечивает приближенную, состоящую только из текста версию документа.

Значение поля обычно хранится в атрибуте. Необходимо хранить значение для того, чтобы представление поля могло заново вычисляться в случае необходимости, например, если пользователь решает изменить стиль форматирования поля. Также необходимо хранение стиля представления содержимого элемента, чтобы способствовать легкой обработке XML-документа. Например, если полная обработка поля невозможна или нежелательна, приложение может проигнорировать поле и использовать в этой ситуации только содержимое. Для строковых значений, если значение идентично его представлению, значение атрибута пропускается во избежание дублирования хранимой информации.

Для полей, которые могут хранить содержимое различных типов, например числа, строки или даты, тип значения хранится в дополнение к самому значению. Значение и атрибуты типа значения объясняются в разделе 6.7.1. Если необходимо больше информации для восстановления поля, она хранится в дополнительных атрибутах.

Ниже приведены наиболее используемые атрибуты элементов поля.

- **Фиксированные поля**

  Многие поля имеют разновидность, где содержимое не изменяется после того, как задано начальное значение. Эти поля отмечаются атрибутом `text:fixed`. См. раздел 6.7.2 для более подробной информации по этому атрибуту.

- **Стиль форматирования**

  Несколько типов полей, особенно те, которые представляют числа, дату или данные времени, содержат стиль форматирования. В формате OpenDocument, этот стиль форматирования представлен атрибутом `style:data-style-name`. Так как пользователь может изменять стиль представления для полей, приложения должны быть способны вычислять заново новое представление содержимого поля в любой момент

времени. См. раздел 6.7.7 для более подробной информации по этому атрибуту.

6.2 Поля документа

Поля OpenDocument могут отображать информацию о текущем документе или о специфичной части текущего документа, такую как автор, номер текущей страницы или дата создания документа. Эти поля имеют общее название — поля документа.

Поля документа часто фиксированы. Поле может быть отмечено как фиксированное, чтобы указать, что содержимое становится неизменяемым, а не пересчитывается при редактировании документа. Например, поле даты показывает текущую дату. Если поле даты было отмечено как фиксированное, значение поля становится неизменяемым в течение последующих редактирований и всегда отражает дату, когда поле было вставлено в документ. Если поле не было отмечено как фиксированное, то его значение изменяется, когда документ редактируется. Также поле автора может отображать автора оригинала или последнего автора документа в зависимости от того, отмечено поле как фиксированное или нет.

Группа полей документа включает:

- поля даты и времени;
- поля номеров страниц;
- поля автора и отправителя;
- поля главы;
- поля имени файла;
- поля шаблона документа.

6.2.1 Поля даты

Поля даты отображают текущую дату. Дата может быть скорректирована для отображения даты, отличной от текущей. Например, дата может изменяться в документе, который был отредактирован поздно ночью так, чтобы отображать дату следующего дня или на несколько дней позже.

Этот элемент содержит представление значения поля даты в зависимости от установленного стиля даты. Дата по умолчанию — текущая дата. Значение этого

элемента может быть задано неизменяемым при использовании атрибута `text:fixed`, описанного в разделе 6.7.2.

```
<define name="paragraph-content" combine="choice">
    <element name="text:date">
        <ref name="text-date-attlist"/>
        <text/>
    </element>
</define>
```

С элементом `<text:date>` могут быть связаны следующие атрибуты:

- значение даты;
- корректировка даты;
- фиксирование (см. раздел 6.7.2);
- стиль форматирования (см. раздел 6.7.7), который должен быть стилем данных типа дата (см. раздел 14.7 для получения более подробной информации).

```
<define name="text-date-attlist" combine="interleave">
    <interleave>
        <ref name="common-field-fixed-attlist"/>
        <ref name="common-field-data-style-name-attlist"/>
    </interleave>
</define>
```

Значение даты

Атрибут `text:date-value` определяет конкретное значение даты. Например, если поле даты отмечено как фиксированное, этот атрибут может использоваться для определения даты, когда поле было отмечено как фиксированное. Этот атрибут может также быть использован, чтобы определить будущую дату. Некоторые приложения поддерживают значение даты и времени в дополнение к значению, состоящему только из даты.

Значение даты должно соответствовать форматам даты, описанным в § 3.2.7 и § 3.2.9 в [xmlschema-2]. Если не задано никакого значения, то устанавливается текущая дата, даже если поле было отмечено как фиксированное.

```
<define name="text-date-attlist" combine="interleave">
    <optional>
        <attribute name="text:date-value">
            <ref name="dateOrDateTime"/>
        </attribute>
    </optional>
</define>
```

Корректировка даты

Значение поля даты может быть скорректировано на определенный промежуток времени, который определен с использованием атрибута `text:date-adjust`. Если промежуток времени отрицательный, он вычитается из значения поля даты, получается дата более ранняя, чем текущая.

Значение этого атрибута должно соответствовать формату промежутка времени, описанному в § 3.2.6 в [xmlschema-2]. Значение может следовать за необязательным знаком минус, чтобы показать отрицательную продолжительность времени.

```
<define name="text-date-attlist" combine="interleave">
    <optional>
        <attribute name="text:date-adjust">
            <ref name="duration"/>
        </attribute>
    </optional>
</define>
```

6.2.2 Поля времени

Поля времени отображают текущее время. Они подобны полям даты, описанным в разделе 6.2.1, и поддерживают те же атрибуты, за исключением того, что для полей времени они называются атрибутами `text:time-value` и `text:time-adjust`.

Этот элемент содержит представление значения поля времени, зависящего от установленного стиля даты. Время по умолчанию — текущее время. Значение данного элемента может быть задано неизменяемым при использовании атрибута `text:fixed`, описанного в разделе 6.7.2.

```
<define name="paragraph-content" combine="choice">
    <element name="text:time">
        <ref name="text-time-attlist"/>
        <text/>
    </element>
</define>
```

С элементом `<text:time>` могут быть связаны следующие атрибуты:

- значение времени;
- корректировка времени;
- фиксирование (см. раздел 6.7.2);

- стиль форматирования (см. раздел 6.7.7), который Стдолжен быть стилем данных типа время (см. раздел 14.7 для получения более подробной информации).

```
<define name="text-time-attlist" combine="interleave">
    <interleave>
        <ref name="common-field-fixed-attlist"/>
        <ref name="common-field-data-style-name-attlist"/>
    </interleave>
</define>
```

Значение времени

Атрибут `text:time-value` отмечает время последнего редактирования документа.

Некоторые приложения поддерживают значение даты и времени в дополнение к значению, состоящему только из даты.

Значение данного атрибута должно соответствовать типам данных «dateTime» или «time», описанным в § 3.2.7 и § 3.2.8 в [xmlschema-2]. Если не задано никакого значения, то устанавливается значение текущего времени, даже если поле было отмечено как фиксированное.

```
<define name="text-time-attlist" combine="interleave">
    <optional>
        <attribute name="text:time-value">
            <ref name="timeOrDateTime"/>
        </attribute>
    </optional>
</define>
```

Корректировка времени

Значение поля времени может быть скорректировано к определенному промежутку времени, который определен с использованием атрибута `text:time-adjust`.

Значение этого атрибута должно соответствовать формату промежутка времени, описанному в § 3.2.6 в [xmlschema-2]. Значение может следовать за необязательным знаком минус, чтобы показать отрицательную продолжительность времени. Положительные значения корректируют время к более позднему значению времени, в то время как отрицательные значения корректируют время к более раннему значению времени. Продолжительность времени усекается до целого значения минут.

```
<define name="text-time-attlist" combine="interleave">
    <optional>
        <attribute name="text:time-adjust">
            <ref name="duration"/>
        </attribute>
    </optional>
</define>
```

Пример – Атрибуты корректировки времени и вызываемый ими эффект:

Если установлен атрибут `text:time-adjust="PTM15"`, поле времени отображает время, которое на 15 минут больше фактического значения времени, определяемого значением поля времени.

Если установлен атрибут `text:time-adjust="-PTH1"`, поле времени отображает время, которое на час меньше фактического значения времени, определяемого значением поля времени.

6.2.3 Поля номера страницы

Поля номера страницы отображают номер текущей страницы. Эти поля особенно полезны в колонтитулах. Например, если поле номера страницы вставлено в нижний колонтитул, номер текущей страницы отображается на каждой странице, на которой появляется нижний колонтитул.

С элементом `<text:page-number>` могут быть связаны могут следующие атрибуты:

- корректировка номеров страниц;
- отображение номера предыдущей или следующей страницы;
- фиксирование (см. раздел 6.7.2);
- стиль форматирования (см. раздел 6.7.8).

  Номера страниц могут быть отформатированы в соответствии с форматом чисел, описанным в разделе 12.2. Если стиль номера не определен, номера страниц форматируются в соответствии со стилем чисел, определенным в стиле текущей страницы.

```
<define name="paragraph-content" combine="choice">
    <element name="text:page-number">
        <ref name="text-page-number-attlist"/>
        <text/>
    </element>
</define>
<define name="text-page-number-attlist" combine="interleave">
    <interleave>
```

```
        <ref name="common-field-num-format-attlist"/>
        <ref name="common-field-fixed-attlist"/>
    </interleave>
</define>
```

Примечание — для отображения общего числа страниц в документе используйте поле `<text:page-count/>`, описанное в разделе 6.4.17.

Корректировка номеров страниц

Значение поля номера страницы может быть скорректировано на определенное число, что позволяет отображать номера следующих или предыдущих страниц. Корректировка осуществляется с использованием атрибута `text:page-adjust`. Когда используется этот атрибут, приложение:

- добавляет значение атрибута к номеру текущей страницы;
- проверяет, существует ли результирующая страница;
- если страница существует, отображается номер этой страницы;
- если страницы не существует, значение поля номера страницы остается пустым и никакого значения не отображается.

```
<define name="text-page-number-attlist" combine="interleave">
    <optional>
        <attribute name="text:page-adjust">
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

Отображение номера предыдущей или следующей страницы

Атрибут `text:select-page` используется для отображения номера предыдущей или следующей страницы, а не номер текущей страницы.

```
<define name="text-page-number-attlist" combine="interleave">
    <optional>
        <attribute name="text:select-page">
            <choice>
                <value>previous</value>
                <value>current</value>
                <value>next</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Примечание — Для отображения номера текущей страницы на всех страницах за исключением первой или последней, используйте комбинацию атрибутов `text:select-page` и `text:page-adjust`.

Пример – Отображение номера текущей страницы на всех страницах, за исключением первой:

```
<text:page-number text:select-page="previous"
                  text:page-adjust="1"
                   style:num-format="1"/>
```

6.2.4 Текст, указывающий на продолжение страницы

В некоторых публикациях в нижней части страницы в дополнение к номеру страницы печатается текст, указывающий на продолжение. Для добавления текста, указывающего на продолжение, используется элемент `<text:page-continuation>`.

```
1514 <define name="paragraph-content" combine="choice">
1515     <element name="text:page-continuation">
1516         <ref name="text-page-continuation-attlist"/>
1517         <text/>
1518     </element>
1519 </define>
```

С элементом `<text:page-continuation>` могут быть связаны следующие атрибуты:

- предыдущая или следующая страница;
- строковое значение.

Предыдущая или следующая страница

Этот атрибут указывает на необходимость проверить наличие предыдущей или следующей страницы, и если страница существует, печатается текст, указывающий на продолжение.

```
1520 <define name="text-page-continuation-attlist" combine="interleave">
1521     <attribute name="text:select-page">
1522         <choice>
1523             <value>previous</value>
1524             <value>next</value>
1525         </choice>
1526     </attribute>
1527 </define>
```

Строковое значение

Этот атрибут определяет отображение текста, указывающего на продолжение. Если атрибут опущен, используется содержимое элемента.

```
1528 <define name="text-page-continuation-attlist" combine="interleave">
1529     <optional>
1530         <attribute name="text:string-value">
```

```
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

6.2.5 Поля отправителя

Есть несколько полей, которые содержат информацию об отправителе текущего документа, например, имя и адрес электронной почты. Информация об отправителе берется из диалогового окна информации о пользователе OpenDocument. Если поле отправителя отмечено как фиксированное использованием атрибута `text:fixed`, информация о первоначальном отправителе в полях отправителя остается неизменной (см. раздел 6.7.2). В противном случае информация обновляется каждый раз, когда редактируется файл, и изменение значений данных полей происходит всегда, когда документ редактируется другим пользователем.

Имя

Этот элемент представляет имя отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-firstname">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Фамилия

Этот элемент представляет фамилию отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-lastname">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Инициалы

Этот элемент представляет инициалы отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-initials">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Титул

Этот элемент представляет титул отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-title">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Должность

Этот элемент представляет должность отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-position">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Адрес электронной почты

Этот элемент представляет адрес электронной почты отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-email">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Личный телефонный номер

Этот элемент представляет номер личного телефона отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-phone-private">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Номер факса

Этот элемент представляет номер факса отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-fax">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Название компании

Этот элемент представляет название компании, в которой работает отправитель.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-company">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Номер рабочего телефона

Этот элемент представляет номер рабочего телефона отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-phone-work">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Улица

Этот элемент представляет название улицы отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-street">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Город

Этот элемент представляет название города отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-city">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Почтовый индекс

Этот элемент представляет почтовый индекс отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-postal-code">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Страна

Этот элемент представляет страну отправителя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-country">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Штат или регион

Этот элемент представляет название штата или региона, в котором проживает отправитель, если применимо.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sender-state-or-province">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

6.2.6 Поля автора

Есть два элемента, доступных для отображения автора документа. Первый элемент отображает полное имя автора, второй — его инициалы.

Значение полей автора может быть зафиксировано использованием атрибута `text:fixed`. Отметка в поле автора как фиксированного оставляет неизменным начальное содержимое поля. В противном случае, содержимое поля изменяется каждый раз, когда документ обновляется, чтобы отразить последнего автора документа.

Имя автора

Этот элемент представляет полное имя автора.

```
<define name="paragraph-content" combine="choice">
    <element name="text:author-name">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Инициалы автора

Этот элемент представляет инициалы автора.

```
<define name="paragraph-content" combine="choice">
```

```
    <element name="text:author-initials">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

6.2.7 Поля главы

Поля главы отображают одно из следующего:

- название текущей главы;
- номер текущей главы;
- одновременно название и номер текущей главы.

Если поле главы расположено внутри верхнего или нижнего колонтитула, оно отображает название или номер текущей главы на каждой странице.

```
<define name="paragraph-content" combine="choice">
    <element name="text:chapter">
        <ref name="text-chapter-attlist"/>
        <text/>
    </element>
</define>
```

С элементом `<text:chapter>` могут быть связаны следующие атрибуты:

- отображение;
- уровень структуры.

Отображение

Атрибут `text:display` определяет информацию, которую поле главы должно отобразить.

```
<define name="text-chapter-attlist" combine="interleave">
    <attribute name="text:display">
        <choice>
            <value>name</value>
            <value>number</value>
            <value>number-and-name</value>
            <value>plain-number-and-name</value>
            <value>plain-number</value>
        </choice>
    </attribute>
</define>
```

Пример –

Если номер текущей главы 2.4, заголовок главы «Работа с таблицами», возможные варианты отображения и результаты следующие:

Значение атрибута `text:display`	*Отображаемое содержимое поля*
`number`	[2.4]
`name`	Работа с таблицами
`number-and-name`	[2.4] Работа с таблицами
`plain-number`	2.4
`plain-number-and-name`	2.4 Работа с таблицами

Уровень структуры

Этот атрибут применяется, чтобы определить используемый уровень структуры. Поле главы отображает номер или название главы вплоть до определенного уровня структуры.

```
<define name="text-chapter-attlist" combine="interleave">
    <attribute name="text:outline-level">
        <ref name="nonNegativeInteger"/>
    </attribute>
</define>
```

6.2.8 Поля имени файла

Поля имени файла отображают имя файла, который в настоящее время редактируется.

С элементом `<text:file-name>` могут быть связаны следующие атрибуты:

- отображение;
- фиксирование.

```
<define name="paragraph-content" combine="choice">
    <element name="text:file-name">
        <ref name="text-file-name-attlist"/>
        <text/>
    </element>
</define>
```

Отображение

Атрибут `text:display` определяет, как отображать имя файла. Следующие варианты позволяют отобразить:

- полное имя файла, включая расширение и путь к нему;
- только путь к файлу;
- только имя файла;
- имя файла и расширение.

Имя файла должно быть записано в формате IRI потому, что формат IRI используется для получения файла или из-за того, что приложение внутренне использует формат IRI и поэтому конвертирует даже специфические для системы пути в формат IRI. Если указанные факторы являются причиной и если путь, имя или расширение файла не могут быть вычислены из IRI, тогда IRI должен отображаться неизмененным.

```
<define name="text-file-name-attlist" combine="interleave">
    <optional>
        <attribute name="text:display">
            <choice>
                <value>full</value>
                <value>path</value>
                <value>name</value>
                <value>name-and-extension</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Фиксированные поля имени файла

Если поле имени файла фиксировано, его значение не изменяется, когда файл редактируется.

```
<define name="text-file-name-attlist" combine="interleave">
    <ref name="common-field-fixed-attlist"/>
</define>
```

6.2.9 Поля имени шаблона документа

Поле имени шаблона документа отображает информацию о используемом шаблоне документа, такую как имя шаблона или имя файла.

С элементом `<text:template-name>` может быть связан атрибут:

- отображение.

```
<define name="paragraph-content" combine="choice">
    <element name="text:template-name">
        <ref name="text-template-name-attlist"/>
        <text/>
    </element>
</define>
```

Отображение

Этот атрибут определяет, какую информацию о шаблоне документа отображать. Доступны следующие варианты отображения:

- полное имя файла, включающее путь и расширение;

- только путь к файлу;
- только имя файла;
- имя файла и расширение;
- заголовок;
- область шаблона документа.

Последние два значения могут быть использованы для диалогов шаблона. Они являются расширенным набором значений отображения для элемента `<text:file-name>`.

```
<define name="text-template-name-attlist">
    <optional>
        <attribute name="text:display">
            <choice>
                <value>full</value>
                <value>path</value>
                <value>name</value>
                <value>name-and-extension</value>
                <value>area</value>
                <value>title</value>
            </choice>
        </attribute>
    </optional>
</define>
```

6.2.10 Поля имен листов

Для документов электронных таблиц поля имен листов отображают имя листа, который в настоящий момент редактируется.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sheet-name">
        <text/>
    </element>
</define>
```

6.3 Поля-переменные

Текстовые документы OpenDocument могут содержать переменные, которые обрабатываются или отображаются с помощью полей-переменных. Переменной называется пара имя/значение. Имя переменной используется во всем документе для обозначения конкретной переменной, и потому имя переменной не может быть заново использовано для определения других типов переменных. Большинство полей-переменных поддерживают различные типы значений, таких как числа, даты, строки и др. В формате файла OpenDocument переменная должна быть объявлена в начале документа.

Имеется три типа переменных.

- **Простые переменные**

  Простые переменные, обычно называемые просто переменными, могут принимать различные значения в различных позициях на всем протяжении документа. Простые переменные могут быть установлены с использованием либо полей установки, либо полей ввода. Поля установки содержат выражение, которое используется для вычисления нового значения переменной. Поля ввода запрашивают у пользователя новое значение. Простые переменные могут быть использованы для отображения различного текста в повторяющихся элементах, таких как верхние или нижние колонтитулы.

- **Переменные пользователя**

  Переменные пользователя имеют одно и то же значение на протяжении всего документа. Если переменная пользователя установлена в любом месте в пределах документа, все поля в документе, которые отображают переменную пользователя, имеют то же самое значение. В пользовательском интерфейсе офисного приложения переменная пользователя может быть установлена в любом экземпляре поля пользователя или с использованием входных полей переменной пользователя. В формате файла OpenDocument значение переменной пользователя может быть установлено только после объявления переменной.

- **Переменные последовательности**

  Переменные последовательности используются для нумерации определенных объектов в текстовом документе OpenDocument, например изображений или таблиц.

Выражения и поля ввода текста также являются полями-переменными, но они не связаны с какими-либо конкретными переменными. Однако, поскольку по функциональности они тесно связаны с полями-переменными, то они также описаны в данном разделе руководства.

Переменные должны быть объявлены прежде, чем они могут быть использованы. Объявления переменных собраны в элементах-контейнерах для

переменных определенного типа. Код OpenDocument для объявления переменных описан в разделах 6.3.1, 6.3.5 и 6.3.8.

6.3.1 Объявление простых переменных

Простые переменные объявляются с использованием элементов `<text:variable-decl>`. Объявление устанавливает имя и тип значения переменной.

Для определения имени и типа значения простой переменной с элементом `<text:variable-decl>` могут быть связаны следующие атрибуты:

- `text:name`

  Имя переменной должно быть уникально. Оно не может использоваться для переменной любого другого типа (см. раздел 6.7.3 для получения информации по использованию этого атрибута).

- `office:value-type`

  См. раздел 6.7.1 для получения информации по использованию этого атрибута.

```
<define name="text-variable-decl">
    <element name="text:variable-decl">
        <ref name="common-field-name-attlist"/>
        <ref name="common-value-type-attlist"/>
    </element>
</define>
```

6.3.2 Задание значения простой переменной

Значения простых переменных могут быть заданы с использованием элементов-установщиков переменных. Эти элементы содержат представление значения переменной, которое может быть пустым, если атрибут `text:display` установлен в значение `none`.

С элементом `<text:variable-set>` могут быть связаны следующие атрибуты:

- `text:name`

  Этот атрибут определяет имя устанавливаемой переменной. Оно должно соответствовать имени переменной, которая уже была объявлена (см. раздел 6.7.3 для получения информации по использованию этого атрибута).

- text:formula

  Этот атрибут содержит формулу для вычисления значения поля-переменной. Если формула совпадает с содержимым элемента поля, этот атрибут может быть опущен (см. раздел 6.7.6 для получения информации по использованию этого атрибута).

- office:value-type и соответствующий атрибут значения (см. раздел 6.7.1 для получения информации по использованию этих атрибутов).

  Примечание — Простая переменная не должна иметь различные типы значения в различных местах документа. Тем не менее, реализация может допускать использование различных типов значения для различных экземпляров той же самой переменной. В случае типов числовых значений float, percentage и currency значение автоматически преобразуется в другой тип значения. Для типов значений, которые имеют внутреннее представление в виде чисел, такие как date, time и boolean значения интерпретируются как числа соответствующего типа. Если переменная используется как для строкового, так и для не строкового типов, то поведение не определено, следовательно, подобная практика не рекомендована.

- text:display

  Этот атрибут может использоваться, чтобы определить необходимо или нет отображать значение элемента <text:variable-set>. Если атрибут text:display установлен в value, значение переменной отображается. Если атрибут установлен в none, значение не отображается (см. раздел 6.7.5 для получения информации по использованию этого атрибута).

- style:data-style-name

  Этот атрибут определяет стиль данных для использования, чтобы задать формат числовой, логической переменной или переменной типа дата/время. Если стиль данных не определен, то используется стандартный стиль данных (см. раздел 6.7.7 для получения информации по использованию этого атрибута).

```
<define name="paragraph-content" combine="choice">
    <element name="text:variable-set">
        <interleave>
            <ref name="common-field-name-attlist"/>
            <ref name="common-field-formula-attlist"/>
            <ref name="common-value-and-type-attlist"/>
            <ref name="common-field-display-value-none-attlist"/>
```

```
            <ref name="common-field-data-style-name-attlist"/>
        </interleave>
        <text/>
    </element>
</define>
```

6.3.3 Отображение простых переменных

Элемент `<text:variable-get>` читает и отображает значение простой переменной. Значение этого элемента является значением последнего предшествовавшего элемента `<text:variable-set>` с таким же атрибутом `text:name`. Этот элемент определяет, как представлять значение переменной, в зависимости от выбранного стиля форматирования.

С элементом `<text:variable-get>` могут быть связаны следующие атрибуты:

- `text:name`

  Этот атрибут определяет имя отображаемой переменной. Имя должно соответствовать имени предшествовавшего элемента `<text:variable-decl>` (см. раздел 6.7.3 для получения информации по использованию этого атрибута).

- `text:display`

  Этот элемент может быть использован для определения того, что отображать: формулу для простой переменной или вычисленное значение переменной (см. раздел 6.7.5 для получения информации по использованию этого атрибута).

- `style:data-style-name`

  Этот атрибут определяет используемый стиль данных для форматирования числовой, логической переменной или переменной типа дата/время. Если стиль данных не определен, используется стандартный стиль данных (см. раздел 6.7.7 для получения информации по использованию этого атрибута).

```
<define name="paragraph-content" combine="choice">
    <element name="text:variable-get">
        <interleave>
            <ref name="common-field-name-attlist"/>
            <ref name="common-field-display-value-formula-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
        </interleave>
        <text/>
    </element>
```

```
</define>
```

6.3.4 Поля ввода простой переменной

Как альтернатива установке значений простых переменных с использованием формул в элементах-установщиках переменных, значения переменных могут запрашиваться у пользователя. Для этого используется элемент `<text:variable-input>`, содержащий представление значения переменной в соответствии с выбранным стилем форматирования. Представление может быть пустым, если атрибут `text:display` установлен в `none`.

С элементом `<text:variable-input>` могут быть связаны следующие атрибуты:

- `text:name`

  Этот атрибут определяет имя отображаемой переменной. Оно должно совпадать с именем ранее объявленной переменной (см. раздел 6.7.3 для получения информации по использованию этого атрибута).

- `text:description`

  Этот необязательный атрибут содержит короткое сообщение, которое представляется пользователям во время запроса ввода. Сообщение должно давать пользователям достаточную информацию о переменной или об использовании значения внутри документа, чтобы позволить им выбрать подходящее значение (см. раздел 6.7.4 для получения информации по использованию этого атрибута).

- `office:value-type` и соответствующий атрибут значения (см. раздел 6.7.1 для получения информации по использованию этого атрибута).

- `text:display`

  Этот атрибут может быть использован для определения отображать или спрятать значение переменной в поле ввода переменной (см. раздел 6.7.5 для получения информации по использованию этого атрибута).

- `style:data-style-name`

  Этот атрибут определяет стиль данных, используемый для форматирования числовой, логической переменной или переменной типа дата/время. Если стиль данных не определен, используется стандартный

стиль данных (см. раздел 6.7.7 для получения информации по использованию этого атрибута).

```
<define name="paragraph-content" combine="choice">
    <element name="text:variable-input">
        <interleave>
            <ref name="common-field-name-attlist"/>
            <ref name="common-field-description-attlist"/>
            <ref name="common-value-type-attlist"/>
            <ref name="common-field-display-value-none-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
        </interleave>
        <text/>
    </element>
</define>
```

6.3.5 Определение переменных пользователя

Переменные пользователя содержат значения, которые отображаются с помощью соответствующих полей. В отличие от простых переменных, переменные пользователя имеют одно и то же значение на всем протяжении документа. По этой причине значение переменной пользователя хранится в самом объявлении переменной.

С элементом `<text:user-field-decl>` могут быть связаны следующие атрибуты:

- `text:name`

  Этот атрибут определяет имя объявляемой переменной, которое должно быть уникально. Оно также не может использоваться для переменных любого другого типа, включая простые переменные и переменные последовательности (см. раздел 6.7.3 для получения информации по использованию этого атрибута).

- `text:formula`

  Этот атрибут содержит формулу для вычисления значения поля переменной пользователя. Если формула совпадает с содержимым элемента поля, этот атрибут может быть опущен (см. раздел 6.7.6 для получения информации по использованию этого атрибута).

- `office:value-type` и соответствующий атрибут значения (см. раздел 6.7.1 для получения информации по использованию этого атрибута).

```
<define name="text-user-field-decl">
    <element name="text:user-field-decl">
        <ref name="common-field-name-attlist"/>
        <optional>
```

```
            <ref name="common-field-formula-attlist"/>
        </optional>
        <ref name="common-value-and-type-attlist"/>
    </element>
</define>
```

6.3.6 Отображение переменных пользователя

Содержимое переменных пользователя может быть отображено с использованием элементов `<text:user-field-get>`.

С элементом `<text:user-field-get>` могут быть связаны следующие атрибуты:

- `text:name`

  Этот атрибут определяет имя отображаемой переменной, которе должно полностью соответствовать имени предшествующего элемента `<text:user-field-decl>` (см. раздел 6.7.3 для получения информации по использованию этого атрибута).

- `text:display`

  Этот атрибут может быть использован для определения одной из следующих возможностей:

  — отображение формулы, используемой для вычисления значения переменной пользователя;

  — отображение значения переменной пользователя;

  — спрятать поля переменной пользователя.

  См. раздел 6.7.5 для получения информации по использованию этого атрибута.

  Примечание — Так как пользовательские интерфейсы офисных приложений обычно позволяют редактировать переменные пользователя щелчком мыши на поле пользователя, то чтобы иметь возможность простого доступа к конкретной переменной пользователя, в качестве привязки может быть использован скрытый элемент `<text:user-field-get>`.

- `style:data-style-name`

  Этот атрибут определяет стиль данных для использования при форматировании числовой, логической переменной или переменной типа дата/время. Если стиль данных не определен, то используется

стандартный стиль данных (см. раздел 6.7.7 для получения информации по использованию этого атрибута).

```
<define name="paragraph-content" combine="choice">
    <element name="text:user-field-get">
        <interleave>
            <ref name="common-field-name-attlist"/>
            <ref name="common-field-display-value-formula-none-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
        </interleave>
        <text/>
    </element>
</define>
```

6.3.7 Поля ввода переменной пользователя

Альтернативным способом установки значений переменных пользователя является использование полей ввода, подобных полям ввода простых переменных. Переменная пользователя может быть установлена этим способом при использовании элемента `<text:user-field-input>`. С того момента как значение изменяемого поля пользователя сохранено в элементе `<text:user-field-decl>`, элемент `<text:user-field-input>` не содержит атрибуты значения и типа значения из поля `<text:variable-input>`.

Представление может оказаться пустым, если атрибут `text:display` установлен в `none`.

С элементом `<text:user-field-input>` могут быть связаны следующие атрибуты:

- `text:name`

  Этот атрибут определяет имя устанавливаемой переменной. Оно должно совпадать с именем ранее объявленной переменной (см. раздел 6.7.3 для получения информации по использованию этого атрибута).

- `text:description`

  Этот необязательный атрибут содержит короткое сообщение, которое представляется пользователям во время запроса ввода. Сообщение должно давать пользователям достаточную информацию о переменной или об использовании значения внутри документа, чтобы позволить им выбрать подходящее значение (см. раздел 6.7.4 для получения информации по использованию этого атрибута).

- `style:data-style-name`

Этот атрибут определяет стиль данных, используемый для форматирования числовой, логической переменной или переменной типа дата/время. Если стиль данных не определен, используется стандартный стиль данных (см. раздел 6.7.7 для получения информации по использованию этого атрибута).

```
<define name="paragraph-content" combine="choice">
    <element name="text:user-field-input">
        <interleave>
            <ref name="common-field-name-attlist"/>
            <ref name="common-field-description-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
        </interleave>
        <text/>
    </element>
</define>
```

6.3.8 Объявление переменных последовательности

Переменные последовательности используются для нумерации объектов внутри текстового документа OpenDocument. Переменные последовательности наиболее часто используются для последовательной нумерации. Тем не менее, формулы выражения могут быть включены в поля последовательности для поддержки более расширенных последовательностей (см. раздел 6.3.9 для получения более подробной информации о полях последовательности и их использовании).

Переменные последовательности объявляются с использованием элемента `<text:sequence-decl>`.

Для упрощения специфичной для главы нумерации, к переменной последовательности могут быть присоединены атрибуты для определения уровня главы и символа-разделителя. С элементом `<text:sequence-decl>` могут быть связаны следующие атрибуты:

- `text:name`

  Этот атрибут определяет имя объявляемой переменной, которое должно быть уникально. Оно также не может быть использовано для переменных любого другого типа, включая простые переменные и переменные пользователя (см. раздел 6.7.3 для получения информации по этому атрибуту).

- `text:display-outline-level`

См. ниже раздел *Уровень структуры* для получения информации по этому атрибуту.

- `text:separation-character`

  См. ниже раздел *Символ-разделитель* для получения информации по этому атрибуту.

```
<define name="text-sequence-decl">
    <element name="text:sequence-decl">
        <ref name="text-sequence-decl-attlist"/>
    </element>
</define>
<define name="text-sequence-decl-attlist" combine="interleave">
    <ref name="common-field-name-attlist"/>
</define>
```

Уровень структуры

Последовательности могут быть пронумерованы по главам. Для использования этой особенности применяется атрибут `text:display-outline-level`, чтобы определить уровень структуры, задающий, какие главы ссылаются на специфичную для главы нумерацию. Все главы с определенного и ниже уровня структуры сбрасывают значение последовательности в ноль — значение по умолчанию. Кроме того, номер последней главы с определенного или ниже уровня структуры добавляется в начало номера последовательности. Выбор нуля в качестве уровня структуры приводит к прямому порядку следования всех элементов последовательности для этой переменной последовательности.

```
<define name="text-sequence-decl-attlist" combine="interleave">
    <attribute name="text:display-outline-level">
        <ref name="nonNegativeInteger"/>
    </attribute>
</define>
```

Символ-разделитель

Если последовательности нумеруются по главам, то этот атрибут используется для выбора символа, который будет отделять номер главы от номера последовательности.

Если значение атрибута `text:display-outline-level` отлично от нуля, символ-разделитель может быть определен. Символ-разделитель по умолчанию — «`.`». В противном случае, если значение `text:display-outline-level` равно нулю, этот атрибут может быть опущен.

```
<define name="text-sequence-decl-attlist" combine="interleave">
    <optional>
        <attribute name="text:separation-character">
            <ref name="character"/>
        </attribute>
    </optional>
</define>
```

Пример – Переменная последовательности:

Переменная последовательности 3.7.36#5 со значением равным 5 объявляется с использованием:

Атрибут	*Значение*
text:display-outline-level	3
text:separation-character	#

6.3.9 Использование полей последовательности

С того момента как переменная последовательности объявлена, она может использоваться в полях последовательности всюду в документе. Большинство полей последовательности просто увеличивают и отображают переменную последовательности. Однако поля последовательности также могут принимать новое начальное значение в любом заданном месте документа. Это начальное значение вычисляется с использованием формулы, которая находится в поле последовательности. Если поле последовательности добавлено без начального значения, офисное приложение автоматически вставляет выражение типа variable+1.

Поля последовательности чаще всего используются для простых счетных последовательностей. Однако способность поддерживать произвольные выражения позволяет задавать более сложные последовательности. Для того чтобы сформировать последовательность четных чисел, все элементы последовательности для этой конкретной переменной должны содержать формулу, увеличивающую значение на два, например variable+2. Последовательность с начальным значением равным 1 и всеми последующими элементами, использующими формулу variable*2, приводит ко всем степеням 2. Так как различные элементы последовательности для одной и той же переменной последовательности могут содержать различные формулы, возможно построение сложных последовательностей.

С элементом `<text:sequence>` могут быть связаны следующие атрибуты:

- `text:name`

  Этот атрибут определяет имя переменной, которую данное поле отображает. Оно должно совпадать с именем переменной последовательности, которая ранее определена (см. раздел 6.7.3 для получения информации по использованию этого атрибута).

- `text:formula`

  Этот необязательный атрибут содержит формулу для вычисления значения поля последовательности. Если этот атрибут опущен, используется выражение, соответствующее содержимому элемента (см. раздел 6.7.6 для получения информации по использованию этого атрибута).

- `style:num-format` и `style:num-letter-sync`

  Эти атрибуты определяют использование стиля нумерации. Если стиль нумерации не определен, стиль нумерации наследуется из стиля страницы (см. раздел 6.7.8 для получения более подробной информации по использованию этого атрибута).

- `text:ref-name`

  См. ниже раздел *Имя ссылки* для получения информации по этому атрибуту.

```
<define name="paragraph-content" combine="choice">
    <element name="text:sequence">
        <interleave>
            <ref name="common-field-name-attlist"/>
            <ref name="common-field-formula-attlist"/>
            <ref name="common-field-num-format-attlist"/>
            <ref name="text-sequence-ref-name"/>
        </interleave>
        <text/>
    </element>
</define>
```

Имя ссылки

Поля последовательности могут быть целью ссылок, в соответствии с представлением используемых полей ссылки (см. раздел 6.6.5 для получения более подробной информации о полях ссылки). Чтобы позволить полю ссылки указывать на определенное поле последовательности, поле последовательности

должно иметь дополнительный атрибут, содержащий имя. Никакие два поля последовательности не могут иметь одинаковое имя ссылки.

Если поле последовательности не является целью ссылки, этот атрибут может быть опущен.

```
<define name="text-sequence-ref-name">
    <optional>
        <attribute name="text:ref-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

6.3.10 Поля выражений

Поля выражений содержат выражения, которые вычисляются, и после чего отображается результирующее значение. Значение выражения форматируется в соответствии с выбранным стилем форматирования.

С элементом `<text:expression>` могут быть связаны следующие атрибуты:

- `text:formula`

  Этот атрибут содержит текущее выражение, используемое для вычисления значения поля выражения (см. раздел 6.7.6 для получения информации об использовании этого атрибута).

- `office:value-type` и соответствующий атрибут значения (см. раздел 6.7.1 для получения информации об использовании этих атрибутов).

- `text:display`

  Этот атрибут применяется для отображения:

  – значения поля;

  – формулы, используемой для вычисления значения.

  См. раздел 6.7.5 для получения информации об использовании этого атрибута.

- `style:data-style-name`

  Этот атрибут определяет стиль данных, применяемый для форматирования числовой, логической переменной или переменной типа дата/время. Если стиль данных не указан, используется стандартный стиль данных (см. раздел 6.7.7 для получения информации об использовании этого атрибута).

```
<define name="paragraph-content" combine="choice">
    <element name="text:expression">
        <interleave>
            <ref name="common-field-formula-attlist"/>
            <optional>
                <ref name="common-value-and-type-attlist"/>
            </optional>
            <ref name="common-field-display-value-formula-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
        </interleave>
        <text/>
    </element>
</define>
```

6.3.11 Поля ввода текста

Поля ввода текста являются полями-переменными. С точки зрения пользовательского интерфейса поля ввода текста подобны полям `<text:variable-input>` и `<text:user-field-input>`. Тем не менее, поле ввода текста не изменяет значение какой-либо переменной.

С элементом `<text:text-input>` может быть связан атрибут:

- `text:description`

  Этот атрибут содержит короткое сообщение, которое представляется пользователю, во время запроса ввода. Сообщение должно давать пользователю достаточно информации о назначении поля и о том, как оно будет использоваться в пределах документа, чтобы позволить ему выбрать подходящее значение (см. раздел 6.7.4 для получения информации по использованию этого атрибута).

```
<define name="paragraph-content" combine="choice">
    <element name="text:text-input">
        <ref name="common-field-description-attlist"/>
        <text/>
    </element>
</define>
```

6.4 Поля метаданных

Поля метаданных отображают метаинформацию о документе, такую как дату создания документа или время последнего распечатывания документа. Имена элементов поля метаданных аналогичны элементам метаданных, описанным в разделе 3.

Все элементы поля метаданных могут быть отмечены как фиксированные с помощью атрибута `text:fixed` (см. раздел 6.7.2).

Несколько полей метаданных отображают дату или время. Элементы этих полей требуют наличия атрибутов `text:date-value` или `text:time-value` и необязательного атрибута `style:data-style-name` (см. раздел 6.7.1 для получения более подробной информации по этим атрибутам).

6.4.1 Автор первоначального документа

Этот элемент представляет имя автора, создавшего исходный документ.

```
<define name="paragraph-content" combine="choice">
    <element name="text:initial-creator">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

6.4.2 Дата создания документа

Этот элемент представляет дату создания документа.

```
<define name="paragraph-content" combine="choice">
    <element name="text:creation-date">
        <interleave>
            <ref name="common-field-fixed-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
            <optional>
                <attribute name="text:date-value">
                    <ref name="dateOrDateTime"/>
                </attribute>
            </optional>
        </interleave>
        <text/>
    </element>
</define>
```

6.4.3 Время создания документа

Этот элемент представляет время создания документа.

```
<define name="paragraph-content" combine="choice">
    <element name="text:creation-time">
        <interleave>
            <ref name="common-field-fixed-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
            <optional>
                <attribute name="text:time-value">
                    <ref name="timeOrDateTime"/>
                </attribute>
            </optional>
        </interleave>
        <text/>
    </element>
</define>
```

6.4.4 Описание документа

Этот элемент содержит краткое описание документа.

```
<define name="paragraph-content" combine="choice">
    <element name="text:description">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

6.4.5 Информация о документе, определяемая пользователем

Этот элемент содержит информацию о документе, определяемую пользователем. Он отображает информацию, находящуюся внутри элемента `<meta:user-defined>`, имеющем то же самое имя.

```
<define name="paragraph-content" combine="choice">
    <element name="text:user-defined">
        <interleave>
            <ref name="common-field-fixed-attlist"/>
            <attribute name="text:name">
                <ref name="string"/>
            </attribute>
            <ref name="common-field-data-style-name-attlist"/>
            <optional>
                <attribute name="office:value">
                    <ref name="double"/>
                </attribute>
            </optional>
            <optional>
                <attribute name="office:date-value">
                    <ref name="dateOrDateTime"/>
                </attribute>
            </optional>
            <optional>
                <attribute name="office:time-value">
                    <ref name="duration"/>
                </attribute>
            </optional>
            <optional>
                <attribute name="office:boolean-value">
                    <ref name="boolean"/>
                </attribute>
            </optional>
            <optional>
                <attribute name="office:string-value">
                    <ref name="string"/>
                </attribute>
            </optional>
        </interleave>
        <text/>
    </element>
</define>
```

6.4.6 Время печати

Этот элемент представляет время последней печати документа.

```
<define name="paragraph-content" combine="choice">
    <element name="text:print-time">
        <interleave>
            <ref name="common-field-fixed-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
            <optional>
                <attribute name="text:time-value">
                    <ref name="time"/>
                </attribute>
            </optional>
        </interleave>
        <text/>
    </element>
</define>
```

6.4.7 Дата печати

Этот элемент представляет дату последней печати документа.

```
<define name="paragraph-content" combine="choice">
    <element name="text:print-date">
        <interleave>
            <ref name="common-field-fixed-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
            <optional>
                <attribute name="text:date-value">
                    <ref name="date"/>
                </attribute>
            </optional>
        </interleave>
        <text/>
    </element>
</define>
```

6.4.8 Кем распечатан документ

Этот элемент представляет имя человека, который последним распечатал документ.

```
<define name="paragraph-content" combine="choice">
    <element name="text:printed-by">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

6.4.9 Заголовок документа

Этот элемент представляет заголовок документа.

```
<define name="paragraph-content" combine="choice">
    <element name="text:title">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

6.4.10 Тема документа

Этот элемент представляет тему документа.

```
<define name="paragraph-content" combine="choice">
    <element name="text:subject">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

6.4.11 Ключевые слова документа

Этот элемент содержит список ключевых слов, используемых для описания документа.

```
<define name="paragraph-content" combine="choice">
    <element name="text:keywords">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

6.4.12 Номер редакции документа

Этот элемент содержит номер редакции документа. При создании документа номер редакции устанавливается равным 1. Каждый раз, когда документ сохраняется, номер редакции документа увеличивается на 1.

```
<define name="paragraph-content" combine="choice">
    <element name="text:editing-cycles">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

Примечание — Так как поле `<text:editing-cycles>` не может быть отформатировано, номер редакции может быть прочитан из содержимого элемента. Поэтому нет необходимости в дополнительном атрибуте.

6.4.13 Продолжительность редактирования документа

Каждый раз, когда документ редактируется, офисное приложение записывает продолжительность интервала между временем открытия документа и временем его закрытия. Затем оно прибавляет эту продолжительность к внутреннему счетчику, тем самым отслеживая общее время, затраченное на редактирование документа.

```
<define name="paragraph-content" combine="choice">
```

```
    <element name="text:editing-duration">
        <interleave>
            <ref name="common-field-fixed-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
            <optional>
                <attribute name="text:duration">
                    <ref name="duration"/>
                </attribute>
            </optional>
        </interleave>
        <text/>
    </element>
</define>
```

6.4.14 Время изменения документа

Этот элемент представляет время последнего изменения документа.

Этот элемент отображает информацию из элемента `<meta:date>`. Такое имя выбрано во избежание путаницы с полями `<text:date>`.

```
<define name="paragraph-content" combine="choice">
    <element name="text:modification-time">
        <interleave>
            <ref name="common-field-fixed-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
            <optional>
                <attribute name="text:time-value">
                    <ref name="time"/>
                </attribute>
            </optional>
        </interleave>
        <text/>
    </element>
</define>
```

6.4.15 Дата изменения документа

Этот элемент представляет дату последнего изменения документа.

Этот элемент отображает информацию из элемента `<meta:date>`. Такое имя выбрано во избежание путаницы с полями `<text:date>`.

```
<define name="paragraph-content" combine="choice">
    <element name="text:modification-date">
        <interleave>
            <ref name="common-field-fixed-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
            <optional>
                <attribute name="text:date-value">
                    <ref name="date"/>
                </attribute>
            </optional>
        </interleave>
        <text/>
    </element>
</define>
```

6.4.16 Автор изменений документа

Этот элемент представляет имя человека, который последним изменял документ.

```
<define name="paragraph-content" combine="choice">
    <element name="text:creator">
        <ref name="common-field-fixed-attlist"/>
        <text/>
    </element>
</define>
```

6.4.17 Поля статистики документа

Эти поля отображают количество объектов определенного типа, содержащихся в документе. Они могут отображать число:

- страниц;
- абзацев;
- слов;
- символов;
- таблиц;
- изображений;
- вложенных объектов.

```
<define name="paragraph-content" combine="choice">
    <element>
        <choice>
            <name>text:page-count</name>
            <name>text:paragraph-count</name>
            <name>text:word-count</name>
            <name>text:character-count</name>
            <name>text:table-count</name>
            <name>text:image-count</name>
            <name>text:object-count</name>
        </choice>
        <ref name="common-field-num-format-attlist"/>
        <text/>
    </element>
</define>
```

6.5 Поля базы данных

Документы могут ссылаться на базы данных и отображать информацию из базы данных как текстовое содержимое. Для отображения информации базы данных схема OpenDocument использует группу текстовых полей, под общим названием полей базы данных. Офисные приложения могут использовать таблицы базы данных с SQL-серверов, следовательно поля базы данных могут

быть использованы для доступа к любой базе данных SQL при условии доступа к соответствующим драйверам.

База данных может содержать следующие компоненты:

- таблицы, хранящие фактические данные;
- запросы, извлекающие подмножество данных из одной или более таблиц;
- формы, представляющие данные;
- отчеты, суммирующие содержимое базы данных.

Поскольку формы и отчеты базы данных не относятся к текстовому содержанию, то они в данном разделе не обсуждаются. С точки зрения вставки информации базы данных в текстовый документ OpenDocument, запросы и таблицы представляются одинаково. Поэтому в оставшейся части этого раздела фраза *таблица базы данных* относится равно как к таблицам базы данных, так и к запросам.

Сами по себе поля базы данных не извлекают информацию из базы данных. В дополнение к полям базы данных, набор строк базы данных также добавляется в документ. Когда к документу добавляются новые данные, то обновляются все поля базы данных, относящиеся к связанной таблице базы данных. С помощью пользовательского интерфейса офисного приложения строки базы данных могут быть добавлены одним из следующих способов:

- вручную, с использованием навигатора источников данных или функции передачи данных в поля;
- с использованием пункта меню «Стандартное письмо» в меню «Файл», этот пункт меню добавляет каждую строку из выбранного множества данных в отдельно созданную копию стандартного письма.

Для отображения данных из таблицы базы данных используйте элемент `<text:database-display>`. Элементы `<text:database-select>` и `<text:database-next>` могут использоваться для определения строка в пределах текущего выбора, которая должна отображаться. Номер текущей строки для конкретной таблицы может быть отображен с использованием элемента `<text:database-row-number>`. И в заключение, поле `<text:database-name>` отображает имя последней используемой базы данных, которой по умолчанию является база данных файла адресной книги.

6.5.1 Источник данных поля базы данных

Источник полей базы данных может быть или именем базы данных, или IRI, содержащий данные ресурса подключения базы данных. Если источником является имя базы данных, тогда это имя используется всеми компонентами офисных приложений для идентификации базы данных. Все поля базы данных содержат имя базы данных или ресурса подключения, а большинство полей также содержит имя таблицы базы данных, которая должна храниться в базе данных. Дополнительный атрибут определяет, имеет ли отношение таблица базы данных к таблице SQL, запросу OpenDocument или является результатом команды SQL.

```
<define name="common-field-database-table">
    <ref name="common-field-database-table-attlist"/>
    <ref name="common-field-database-name"/>
</define>
```

Имя базы данных

Атрибут `text:database-name` устанавливает исходную базу данных по ее имени.

```
<define name="common-field-database-name" combine="choice">
    <optional>
        <attribute name="text:database-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Ресурс подключения

Элемент `<form:connection-resource>` устанавливает исходную базу данных по [XLink]. Ее атрибут `xlink:href` является ссылкой на файл, содержащий базу данных, или содержит информацию о том, как осуществить соединение с базой данных, например [JDBC] URL (см. также раздел 11.1.20).

```
<define name="common-field-database-name" combine="choice">
    <ref name="form-connection-resource"/>
</define>
```

Имя таблицы базы данных

Атрибут `text:table-name` определяет таблицу в пределах исходной базы данных.

```
<define name="common-field-database-table-attlist" combine="interleave">
    <attribute name="text:table-name">
```

```
        <ref name="string"/>
    </attribute>
</define>
```

Тип базы данных

Атрибут `text:table-type` определяет, имеет ли отношение таблица базы данных к таблице SQL, запросу OpenDocument или является результатом команды SQL.

```
<define name="common-field-database-table-attlist" combine="interleave">
    <optional>
        <attribute name="text:table-type">
            <choice>
                <value>table</value>
                <value>query</value>
                <value>command</value>
            </choice>
        </attribute>
    </optional>
</define>
```

6.5.2 Отображение содержимого базы данных

Элемент `<text:database-display>` отображает данные из базы данных. Когда в документ добавляется новый набор данных, все поля, отображающие данные из таблиц базы данных, обновляют свое содержимое.

С элементом `<text:database-display>` могут быть связаны следующие атрибуты:

- `text:database-name`, `text:table-name` и `text:table-type`

  Эти атрибуты определяют базу данных и таблицу базы данных, используемые этими полями.

- `text:database-column-name`

  См. ниже раздел *Имя столбца* для получения информации об этом атрибуте.

- `style:data-style-name`

  Если столбец определяет числовое, логическое значение или значение даты или времени, данные форматируются в соответствии с подходящим стилем данных. Если не определено никакого стиля данных, то используется стиль данных связанный с этим столбцом (см. раздел 6.7.7 для получения более подробной информации по использованию этого атрибута).

```
<define name="paragraph-content" combine="choice">
    <element name="text:database-display">
        <ref name="text-database-display-attlist"/>
        <text/>
    </element>
</define>
<define name="text-database-display-attlist" combine="interleave">
    <ref name="common-field-database-table"/>
</define>
<define name="text-database-display-attlist" combine="interleave">
    <ref name="common-field-data-style-name-attlist"/>
</define>
```

Имя столбца

Атрибут `text:column-name` определяет столбец, из которого отображаются данные. Значение этого атрибута должно быть столбцом, содержащимся в указанной базе данных.

```
<define name="text-database-display-attlist" combine="interleave">
    <attribute name="text:column-name">
        <ref name="string"/>
    </attribute>
</define>
```

6.5.3 Выборка следующей строки базы данных

Элемент `<text:database-next>` изменяет строку в текущей выборке, используемой для отображения во всех последующих полях `<text:database-display>`. Следующая строка из текущего выделения выбирается, если она удовлетворяет заданным условиям. Если требуется следующая строка независимо от каких-либо условий, условие может быть опущено или установлено в значение `true`.

С элементом `<text:database-next>` могут быть связаны следующие атрибуты:

- `text:database-name`, `text:table-name` и `text:table-type`

  Эти атрибуты определяют базу данных и таблицу базы данных, которые это поле использует.

- `text:condition`

  См. ниже раздел *Условие* для получения информации об этом атрибуте.

```
<define name="paragraph-content" combine="choice">
    <element name="text:database-next">
        <ref name="text-database-next-attlist"/>
    </element>
</define>
<define name="text-database-next-attlist" combine="interleave">
```

```
    <ref name="common-field-database-table"/>
</define>
```

Условие

Атрибут text:condition определяет условное выражение. Выражение вычисляется, и если результат интерпретируется как логическое значение «истина», в качестве новой текущей строки используется следующая строка. Значения поля базы данных могут использоваться в выражениях, путем указания в квадратных скобках имени базы данных, имени таблицы и имени столбца, разделенных точками.

Если атрибут text:condition не представлен, считается, что результатом выражения является true, означающее, что следующая строка выбирается без условия.

```
<define name="text-database-next-attlist" combine="interleave">
    <optional>
        <attribute name="text:condition">
            <ref name="formula"/>
        </attribute>
    </optional>
</define>
```

Пример –

```
text:formula='ooo-w:[address book file.address.FIRSTNAME] == "Юлия"'
```

Этот пример определяет условие, которое истинно, если текущая строка из таблицы базы данных адресной книги является адресом человека по имени Юлия. Если показанное в этом примере условие используется в элементе <text:database-next>, произойдет следующее.

- Элементы <text:database-display> отобразят данные из первой строки текущего выделения.
- Если столбец FIRSTNAME текущей строки читается как Юлия, текущая строка заменяется, в противном случае, ничего не происходит.
- Если первая строка Юлия, следующие элементы <text:database-display> отображают данные из второй строки, в противном случае, они отображают данные из первой строки.

См. раздел 6.7.6 для получения более подробной информации по синтаксису формулы атрибута text:condition, который аналогичен атрибуту text:formula.

6.5.4 Выбор номера строки

Элемент `<text:database-row-select>` выбирает определенную строку из текущего выделения. Как и с элементом `<text:database-row-next>`, условие может быть определено так, что указанная строка выбирается только если условие принимает значение `true`.

С элементом `<text:database-row-select>` могут быть связаны следующие атрибуты:

- `text:database-name`, `text:table-name` и `text:table-type`

  Эти атрибуты определяют базу данных и таблицу базы данных, которые используются данным полем.

- `text:condition`

  Этот атрибут определяет условное выражение (см. раздел 6.5.3 для получения полного представления об использовании этого атрибута).

- `text:row-number`

  См. ниже атрибут *Выбор номера строки* для получения дополнительной информации.

```
<define name="paragraph-content" combine="choice">
    <element name="text:database-row-select">
        <ref name="text-database-row-select-attlist"/>
    </element>
</define>
<define name="text-database-row-select-attlist" combine="interleave">
    <ref name="common-field-database-table"/>
</define>
<define name="text-database-row-select-attlist" combine="interleave">
    <optional>
        <attribute name="text:condition">
            <ref name="formula"/>
        </attribute>
    </optional>
</define>
```

Выбор номера строки

Этот атрибут определяет номер выбираемой строки, когда условие принимает значение `true`.

```
<define name="text-database-row-select-attlist" combine="interleave">
    <optional>
        <attribute name="text:row-number">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

6.5.5 Отображение номера строки

Элемент `<text:database-row-number>` отображает номер текущей строки для заданной таблицы. Имейте в виду, что элемент отображает фактически номер строки из базы данных, а не номер строки в текущем выделении, который используется как значение атрибута в элементе `<text:database-row-select>`.

С элементом `<text:database-row-number>` могут быть связаны следующие атрибуты:

- `text:database-name`, `text:table-name` и `text:table-type`

  Эти атрибуты определяют базу данных и таблицу базы данных, которые использует данное поле.

- `style:num-format` и `style:num-letter-sync`

  Эти атрибуты определяют, как должно форматироваться число (см. раздел 6.7.8 для получения более подробной информации по использованию этого атрибута).

- `text:value`

  Этот атрибут определяет номер текущей строки. Номер изменяется, когда в текущий документ добавляются новые данные.

```
2113 <define name="paragraph-content" combine="choice">
2114     <element name="text:database-row-number">
2115         <interleave>
2116             <ref name="common-field-database-table"/>
2117             <ref name="common-field-num-format-attlist"/>
2118             <optional>
2119                 <attribute name="text:value">
2120                     <ref name="nonNegativeInteger"/>
2121                 </attribute>
2122             </optional>
2123         </interleave>
2124         <text/>
2125     </element>
2126 </define>
```

6.5.6 Отображение текущей базы данных и таблицы

Офисные приложения могут отслеживать последние базу данных и таблицу, которые использовались в документе. Другими словами, таблицу, которая использовалась последним полем, вставленным в документ. Элемент `<text:database-name>` отображает базу данных и имя таблицы, которая использовалась самой последней.

С элементом `<text:database-name>` могут быть связаны следующие атрибуты:

- `text:database-name`, `text:table-name` и `text:table-type`

  Эти атрибуты определяют базу данных и таблицу базы данных, которые использует данное поле.

```
<define name="paragraph-content" combine="choice">
    <element name="text:database-name">
        <ref name="common-field-database-table"/>
        <text/>
    </element>
</define>
```

6.6 Прочие поля

6.6.1 Поля переменных страницы

Переменные страницы позволяют определять альтернативную схему нумерации страниц. Есть только одна переменная страницы, и она задается любым способом задания поля переменной страницы в документе. Значение переменной страницы увеличиваются на каждой странице таким же образом, как и обычные номера страниц.

Установка значений полей переменных страницы

Для установки значения поля переменной страницы используйте элемент `<text:variable-page-set>`.

```
<define name="paragraph-content" combine="choice">
    <element name="text:page-variable-set">
        <ref name="text-set-page-variable-attlist"/>
        <text/>
    </element>
</define>
```

Включение или выключение переменных страницы

В начале документа переменная страницы неактивна. Атрибут `text:active` может быть использован для блокировки переменной страницы после того, как она была использована в документе.

```
<define name="text-set-page-variable-attlist" combine="interleave">
    <optional>
        <attribute name="text:active">
            <ref name="boolean"/>
        </attribute>
```

```
    </optional>
</define>
```

Корректировка переменной страницы

Атрибут `text:page-adjust` определяет корректировку страницы. Значение активной переменной страницы — номер текущей страницы плюс значение корректировки ближайшей страницы, которое было установлено ранее.

```
<define name="text-set-page-variable-attlist" combine="interleave">
    <optional>
        <attribute name="text:page-adjust">
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

Отображение полей переменных страницы

Элемент `<text:variable-page-get>` отображает значение переменной страницы. Поле может быть форматировано так же, как и поля обычных номеров страниц.

```
<define name="paragraph-content" combine="choice">
    <element name="text:page-variable-get">
        <ref name="text-get-page-variable-attlist"/>
        <text/>
    </element>
</define>
```

С элементом `<text:get-page-variable>` могут быть связаны следующие атрибуты:

- `style:num-format` и `style:num-letter-sync`

  Эти атрибуты определяют, как должен быть отформатирован номер (см. раздел 6.7.8 для получения более подробной информации об использовании этих атрибутов).

```
<define name="text-get-page-variable-attlist" combine="interleave">
    <ref name="common-field-num-format-attlist"/>
</define>
```

6.6.2 Поля для заполнения

Формат OpenDocument использует поля для заполнения с целью указания места в документе, которое пользователь должен заполнить некоторой информацией. Например, в шаблоне письма раздел документа может быть зарезервирован для адреса получателя. Поле для заполнения отображает текст,

информирующий пользователя о назначении поля для заполнения, и иногда содержит описание. Поля для заполнения могут представлять различные текстовые элементы, такие как текст или таблицы.

Этот элемент содержит некоторый короткий текст, который отображается полем для заполнения.

```
<define name="paragraph-content" combine="choice">
    <element name="text:placeholder">
        <ref name="text-placeholder-attlist"/>
        <text/>
    </element>
</define>
```

С элементом `<text:placeholder>` могут быть связаны следующие атрибуты:

- тип поля для заполнения;
- описание поля для заполнения.

Тип поля для заполнения

Имеется пять различных типов полей для заполнения, представляющих пять возможных типов содержимого: текст, таблицы, текстовые блоки, изображения или объекты. Атрибут `text:placeholder-type` представляет тип содержимого. Этот атрибут обязателен, и он указывает, какой тип текстового содержимого представляет поле для заполнения. Атрибут может принимать значения `text`, `text-box`, `image`, `table` или `object`.

```
<define name="text-placeholder-attlist" combine="interleave">
    <attribute name="text:placeholder-type">
        <choice>
            <value>text</value>
            <value>table</value>
            <value>text-box</value>
            <value>image</value>
            <value>object</value>
        </choice>
    </attribute>
</define>
```

Описание поля для заполнения

В дополнение к краткому тексту, хранящемуся в содержимом элемента, с элементом поля для заполнения может быть связан атрибут `text:description`. Этот атрибут необязателен. Предназначение — содержать более подробное описание цели полей для заполнения, чем описание, сохраненное в содержимом

элемента (см. раздел 6.7.4 для получения информации по использованию атрибута `text:description`).

```
<define name="text-placeholder-attlist" combine="interleave">
    <ref name="common-field-description-attlist"/>
</define>
```

6.6.3 Условные текстовые поля

Текстовые поля могут быть использованы для отображения одного или другого текста, в зависимости от условия. Условные текстовые поля задаются с помощью условия и двух текстовых строк. Если условие истинно, отображается первая из текстовых строк. Если условие ложно, отображается другая текстовая строка.

```
<define name="paragraph-content" combine="choice">
    <element name="text:conditional-text">
        <ref name="text-conditional-text-attlist"/>
        <text/>
    </element>
</define>
```

С элементом `<text:conditional-text>` могут быть связаны следующие атрибуты:

- условие;
- текст, отображаемый в случае, если условие истинно;
- текст, отображаемый в случае, если условие ложно;
- текущее состояние.

Атрибут `text:condition` содержит логическое выражение. В зависимости от результата, отображается значение атрибута `text:display-if-true` или `text:display-if-false`.

```
<define name="text-conditional-text-attlist" combine="interleave">
    <attribute name="text:condition">
        <ref name="formula"/>
    </attribute>
</define>
```

Отображаемый текст, если условие истинно

Атрибут `text:string-value-if-true` содержит текстовую строку, которая отображается, если условие вернуло значение `true`.

```
<define name="text-conditional-text-attlist" combine="interleave">
    <attribute name="text:string-value-if-true">
        <ref name="string"/>
```

```
    </attribute>
</define>
```

Отображаемый текст, если условие ложно

Атрибут `text:string-value-if-false` содержит текстовую строку, которая отображается, если условие вернуло значение `false`.

```
<define name="text-conditional-text-attlist" combine="interleave">
    <attribute name="text:string-value-if-false">
        <ref name="string"/>
    </attribute>
</define>
```

Текущее значение и состояние

Атрибут `text:current-value` содержит результат вычисления условия, заданного выражением в атрибуте `text:condition`. Результат, заданный в явном виде, позволяет приложениям отложить вычисление результата пока он не потребуется. Этот атрибут полезен по следующим причинам:

- если вычисление выражения требует больших издержек, например, выражение содержит ссылки на несколько баз данных;
- чтобы позволить преобразованиям корректно отображать состояние документа без необходимости анализа и вычисления условия.

```
<define name="text-conditional-text-attlist" combine="interleave">
    <optional>
        <attribute name="text:current-value">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Примечание — Значение этого атрибута перезаписывается каждый раз, как только приложение вычисляет выражение. Этот атрибут не имеет другой функции, кроме упрощения преобразования или первоначального отображения документа.

6.6.4 Скрытое текстовое поле

Скрытое текстовое поле тесно связано с условным текстовым полем. Оно отображает определенный текст, кроме случаев, когда условие возвращает значение `true`. В этом случае оно ничего не отображает.

```
<define name="paragraph-content" combine="choice">
    <element name="text:hidden-text">
        <ref name="text-hidden-text-attlist"/>
        <text/>
```

```
    </element>
</define>
```

С элементом `<text:hidden-text>` могут быть связаны следующие атрибуты:

- условие;
- текст;
- скрытость.

Условие

Атрибут `text:condition` содержит логическое выражение. Если результат вычисления выражения — `true`, то текст скрытый.

```
<define name="text-hidden-text-attlist" combine="interleave">
    <attribute name="text:condition">
        <ref name="formula"/>
    </attribute>
</define>
```

Текст

Атрибут `text:string-value` определяет отображаемый текст, если значение условия — `false`.

```
<define name="text-hidden-text-attlist" combine="interleave">
    <attribute name="text:string-value">
        <ref name="string"/>
    </attribute>
</define>
```

Скрытость

Атрибут `text:is-hidden` определяет, является ли в настоящий момент поле видимым или нет. Цель этого атрибута подобна назначению атрибута `text:current-value` в поле `text:condition`. Запись результатов позволяет при изменениях корректно представлять документ без необходимости анализа условного выражения или вычисления условия, при загрузке документа.

```
<define name="text-hidden-text-attlist" combine="interleave">
    <optional>
        <attribute name="text:is-hidden">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Примечание — Значение этого атрибута перезаписывается каждый раз, как только приложение вычисляет выражение. Этот атрибут не имеет другой функции, кроме упрощения преобразования или первоначального отображение документа.

6.6.5 Поля ссылок

Формат OpenDocument использует четыре типа поля ссылки и каждый тип представлен своим собственным элементом. Типы полей ссылок основываются на типах элементов, на которые они ссылаются: сноски, закладки, ссылки и последовательности. Каждая ссылка содержит формат ссылки, определяющий какая информация отображается о указанном объекте цели. Например, ссылки могут отображать:

- номер страницы указанного объекта;
- номер главы указанного объекта;
- текст, отражающий, выше или ниже относительно поля ссылки находится указанный объект.

Кроме того, каждое поле ссылки должно определять свою цель, что, как правило, делается с использованием атрибута имени. Закладки и ссылки задаются именем соответствующей закладки или ссылки. Обычные сноски, концевые сноски и последовательности связываются именами, используемыми приложением для автоматического создания файла формата OpenDocument.

```
2233 <define name="paragraph-content" combine="choice">
2234     <element>
2235         <choice>
2236             <name>text:reference-ref</name>
2237             <name>text:bookmark-ref</name>
2238         </choice>
2239         <interleave>
2240             <ref name="text-common-ref-content"/>
2241             <ref name="text-ref-content"/>
2242         </interleave>
2243     </element>
2244 </define>
2245 <define name="paragraph-content" combine="choice">
2246     <element name="text:note-ref">
2247         <interleave>
2248             <ref name="text-common-ref-content"/>
2249             <ref name="text-note-ref-content"/>
2250             <ref name="text-ref-content"/>
2251         </interleave>
2252     </element>
2253 </define>
2254 <define name="paragraph-content" combine="choice">
2255     <element name="text:sequence-ref">
2256         <interleave>
2257             <ref name="text-common-ref-content"/>
```

```
            <ref name="text-sequence-ref-content"/>
        </interleave>
    </element>
</define>
<define name="text-common-ref-content" combine="interleave">
    <text/>
</define>
```

С элементами поля ссылки могут быть связаны следующие атрибуты:

- имя ссылки;
- формат ссылки.

Имя ссылки

Атрибут `text:ref-name` однозначно определяет ссылочный элемент. Поскольку закладки и ссылки имеют имя, оно используется соответствующими полями ссылок. Обычные сноски, концевые сноски и последовательности определяются именем, которое обычно генерируется автоматически.

```
<define name="text-common-ref-content" combine="interleave">
    <optional>
        <attribute name="text:ref-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Класс сноски

Для элементов `<text:note-ref>` атрибут `text:note-class` определяет, ссылается ли поле на обычную или на концевую сноску.

```
<define name="text-note-ref-content" combine="interleave">
    <ref name="text-note-class"/>
</define>
```

Формат ссылки

Атрибут `text:reference-format` определяет, какую информацию о ссылке следует отображать. Если формат ссылки не определен, то по умолчанию используется формат страницы.

Все типы полей ссылок поддерживают следующие значения форматов для этого атрибута:

- `page` — отображается номер страницы, на которой встречается указанный объект;

- chapter — отображается номер главы, на которой встречается указанный объект;
- direction — отображается, выше или ниже относительно поля ссылки находится указанный объект;
- text — отображается текст указанного объекта.

Ссылки на поля последовательности поддерживают следующие три дополнительных значения:

- category-and-value — отображается имя и значение последовательности;
- caption — отображается заголовок, в котором используется последовательность;
- value — отображается значение последовательности.

```
<define name="text-ref-content" combine="interleave">
    <optional>
        <attribute name="text:reference-format">
            <choice>
                <value>page</value>
                <value>chapter</value>
                <value>direction</value>
                <value>text</value>
            </choice>
        </attribute>
    </optional>
</define>
<define name="text-sequence-ref-content" combine="interleave">
    <optional>
        <attribute name="text:reference-format">
            <choice>
                <value>page</value>
                <value>chapter</value>
                <value>direction</value>
                <value>text</value>
                <value>category-and-value</value>
                <value>caption</value>
                <value>value</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Пример – Различные форматы ссылок и их отображения.

Следующая таблица иллюстрирует все возможные форматы ссылок и результирующее отображение ссылки, которое может быть использовано, чтобы сослаться на саму таблицу. В левом столбце перечислены значения атрибута text:reference-format, а в правом – отображение ссылки:

Формат ссылки	*Отображение ссылки*
page	138
chapter	3.7.27
text	Таблица 2: Примеры форматов ссылок
direction	выше
category-and-value	Таблица 1
caption	Примеры форматов ссылок
value	1

6.6.6 Поля скриптов

Поле скрипта хранит скрипты или фрагменты скриптов. Поле может быть использовано для хранения или редактирования скриптов, которые прикреплены к документу. Первичная цель этого поля — обеспечить эквивалент тегу `<script>` в [HTML4] так, чтобы содержание тега `<script>` из HTML могло импортироваться, редактироваться и экспортироваться с использованием офисных приложений.

Исходный код скрипта может быть сохранен одним из следующих способов.

- Исходный код может содержаться в элементе `<text:script>`.
- Исходный код может храниться во внешнем файле. Используйте атрибут `xlink:href`, чтобы определить местоположение файла.

Элемент должен иметь атрибут `xlink:href` или содержимое, но не оба одновременно.

```
<define name="paragraph-content" combine="choice">
    <element name="text:script">
        <interleave>
            <choice>
                <group>
                    <attribute name="xlink:href">
                        <ref name="anyURI"/>
                    </attribute>
                    <optional>
                        <attribute name="xlink:type" a:defaultValue="simple">
                            <value>simple</value>
                        </attribute>
                    </optional>
                </group>
                <text/>
            </choice>
            <optional>
                <attribute name="script:language">
                    <ref name="string"/>
                </attribute>
            </optional>
        </interleave>
    </element>
</define>
```

URL скрипта

Атрибут `xlink:href` определяет местоположение файла, содержащего исходный код скрипта. Поле скрипта должно иметь либо атрибут URL, либо содержимое, но не оба одновременно.

Язык скрипта

Атрибут `script:language` определяет язык, на котором написан исходный код скрипта, например JavaScript.

6.6.7 Поля макросов

Поле макроса содержит имя макроса, который выполняется, когда поле активируется. Поле также содержит описание, которое отображается как содержимое поля.

С элементом `<text:execute-macro>` может быть связан атрибут:

- имя макроса.

```
<define name="paragraph-content" combine="choice">
    <element name="text:execute-macro">
        <optional>
            <attribute name="text:name">
                <ref name="string"/>
            </attribute>
        </optional>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <text/>
    </element>
</define>
```

Имя макроса

Атрибут `text:name` определяет макрос, который выполняется, когда активируется поле.

6.6.8 Скрытые поля абзаца

Скрытое поле абзаца имеет функционал, подобный функционалу скрытых текстовых полей. Однако скрытое поле абзаца не имеет какого-либо содержимого. Оно скрывает абзац, в котором находится, что позволяет скрывать или

отображать абзац форматированного текста в зависимости от того, какое значение вернуло определенное условие: `true` или `false`.

Скрытые поля абзаца часто используются совместно со стандартным письмом. Например, если состояние зависит от поля базы данных, скрытое поле абзаца может использоваться для выборочного включения абзацев в стандартное письмо, в зависимости от содержимого базы данных. В одном абзаце может находиться множество полей абзаца. Абзац отображается, если хотя бы в одном скрытом поле абзаца условие вернуло значение `false`. Как вариант, условия, связанные с несколькими скрытыми полями абзаца могут быть объединены в одно в единственном поле с помощью логических операций в условиях.

Примечание — В отличие от большинства полей это поле не отображает текст, а влияет на абзац, в котором оно содержится.

С элементом `<text:hidden-paragraph>` могут быть связаны следующие атрибуты:

- условие;
- скрытость.

```
<define name="paragraph-content" combine="choice">
    <element name="text:hidden-paragraph">
        <ref name="text-hidden-paragraph-attlist"/>
        <text/>
    </element>
</define>
```

Условие

Атрибут `text:condition` содержит логическое выражение. Если результатом выполнения логического выражения является `true`, абзац скрывается. Если же результатом выполнения логического выражения является `false`, абзац отображается.

```
<define name="text-hidden-paragraph-attlist" combine="interleave">
    <attribute name="text:condition">
        <ref name="formula"/>
    </attribute>
</define>
```

Скрытость

Атрибут `text:is-hidden` хранит значение: отображается ли абзац или нет. Его цель аналогична соответствующему атрибуту скрытого текстового поля, а именно, позволить правильно отображать абзац без необходимости сначала

вычислять условное выражение. Значение этого атрибута перезаписывается новым значением каждый раз, когда приложение вычисляет значение выражения.

Примечание — Этот атрибут не имеет другого назначения, кроме упрощения преобразований или начального отображения документа.

```
<define name="text-hidden-paragraph-attlist" combine="interleave">
    <optional>
        <attribute name="text:is-hidden">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

6.6.9 Поля DDE-соединения

Поле DDE позволяет отображать информацию из DDE-соединения. Только один параметр требуется для поля DDE — имя DDE-соединения, которое передает данные в это поле. Этот элемент DDE-соединения определяет фактическое поле DDE, которое появляется в тексте.

Элемент поля содержит самые последние данные, которые были пересланы через DDE-соединение. Это может быть использовано для представления документа в том случае, если невозможно получить доступ к DDE-соединению.

```
<define name="paragraph-content" combine="choice">
    <element name="text:dde-connection">
        <attribute name="text:connection-name">
            <ref name="string"/>
        </attribute>
        <text/>
    </element>
</define>
```

С элементом `<text:dde-connection>` может быть связан атрибут:

- имя DDE-соединения.

Имя DDE-соединения

Атрибут `text:name` определяет имя DDE-соединения, на которое ссылается поле.

6.6.10 Поля измерений

В пределах текста, содержащегося в измеряемых векторных объектах (см. раздел 9.2.11), поле `<text:measure>` отображает текущие размеры. Атрибут

`draw:kind` определяет, какую часть измерения отображать. Он может принимать одно из следующих значений:

- `value` — отображается измеренное значение, например «12»;
- `unit` — отображается единица измерения, например «см»;
- `gap` — промежуток или пробел отображается тогда и только тогда, когда направление написания текста измерения перпендикулярно линии измерения. Цель этого значения — добавить немного свободного пространства между линией измерения и текстом, если текст отображается перпендикулярно к линии измерения.

```
<define name="paragraph-content" combine="choice">
    <element name="text:measure">
        <attribute name="text:kind">
            <choice>
                <value>value</value>
                <value>unit</value>
                <value>gap</value>
            </choice>
        </attribute>
        <text/>
    </element>
</define>
```

6.6.11 Поле формулы таблицы

Поле формулы таблицы унаследовано от предыдущих версий современных офисных приложений. Оно не должно использоваться в новых документах. В нем хранится формула, применяемая в таблицах, функция которой лучше выполняется атрибутом `table:formula` ячейки таблицы.

Примечание — Этот элемент не должен использоваться в новых документах.

Элемент поле формулы таблицы может иметь следующие атрибуты:

- `text:formula`

  Этот атрибут содержит выражение, используемое для вычисления значения поля формулы таблицы (см. раздел 6.7.6 для получения информации по использованию этого атрибута).

- `text:display`

  Используйте этот атрибут для следующего:

  — отобразить значение поля;

  — отобразить формулу, используемую для вычисления значения.

См. раздел 6.7.5 для получения информации по использованию этого атрибута.

- `style:data-style-name`

  Этот атрибут определяет стиль данных, используемый для форматирования числовой, логической переменной или переменной типа дата/время. Если стиль данных не определен, используется стандартный стиль данных (см. раздел 6.7.7 для получения информации по использованию этого атрибута).

```
<define name="paragraph-content" combine="choice">
    <element name="text:table-formula">
        <interleave>
            <ref name="common-field-formula-attlist"/>
            <ref name="common-field-display-value-formula-attlist"/>
            <ref name="common-field-data-style-name-attlist"/>
        </interleave>
        <text/>
    </element>
</define>
```

6.7 Общие атрибуты полей

Атрибуты, описанные в этом разделе, могут быть использованы различными элементами поля.

6.7.1 Переменные значения и типы значения

Переменные и большинство полей переменных имеют текущие значения. Каждая переменная имеет тип значения, который должен быть определен, когда поле поддерживает множество типов значений. Тип значения определяется использованием атрибута `office:value-type`.

```
<define name="common-value-type-attlist">
    <attribute name="office:value-type">
        <ref name="valueType"/>
    </attribute>
</define>
```

В зависимости от типа значения, само значение записывается в различные атрибуты значения. Поддерживаемые типы значения, их соответствующие атрибуты и способ кодирования,описаны в следующей таблице:

Тип значения	*Атрибут(ы) значения*	*Кодируется как...*	*Пример*
`float`	`office:value`	Числовое значение	`"12.345"`
`percentage`	`office:value`	Числовое значение	`"0.50"`

Тип значения	*Атрибут(ы) значения*	*Кодируется как...*	*Пример*
currency	office:value и office:currency	Числовое значение и символ валюты	"100" "USD"
date	office:date-value	Значение даты, как определено в § 3.2.9 в [xmlschema-2], или значение даты и времени, как определено в § 3.2.7 в [xmlschema-2]	"2003-04-17"
time	office:time-value	Длительность, как определено в § 3.2.6 в [xmlschema-2]	"PT03H30M00S"
boolean	office:boolean-value	true или false	"true"
string	office:string-value	Строки	"abc def"

Понятие значений полей OpenDocument, типов значений и их кодирование в XML подобно понятиям соответствующих элементов XML для атрибутов ячеек таблицы (см. раздел 8.1.3 для получения информации по ячейкам таблиц и их атрибутам).

Сущность %value-attlist; определяется следующим образом:

```
<define name="common-value-and-type-attlist">
    <choice>
        <group>
            <attribute name="office:value-type">
                <value>float</value>
            </attribute>
            <attribute name="office:value">
                <ref name="double"/>
            </attribute>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>percentage</value>
            </attribute>
            <attribute name="office:value">
                <ref name="double"/>
            </attribute>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>currency</value>
            </attribute>
            <attribute name="office:value">
                <ref name="double"/>
            </attribute>
            <optional>
                <attribute name="office:currency">
                    <ref name="string"/>
                </attribute>
            </optional>
        </group>
        <group>
```

```
            <attribute name="office:value-type">
                <value>date</value>
            </attribute>
            <attribute name="office:date-value">
                <ref name="dateOrDateTime"/>
            </attribute>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>time</value>
            </attribute>
            <attribute name="office:time-value">
                <ref name="duration"/>
            </attribute>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>boolean</value>
            </attribute>
            <attribute name="office:boolean-value">
                <ref name="boolean"/>
            </attribute>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>string</value>
            </attribute>
            <optional>
                <attribute name="office:string-value">
                    <ref name="string"/>
                </attribute>
            </optional>
        </group>
    </choice>
</define>
```

6.7.2 Фиксация

Атрибут `text:fixed` определяет фиксировано или нет значение элемента поля. Если значение поля фиксировано, то значение элемента поля, к которому присоединен этот атрибут, будет неизменно на протяжении всех будущих редактирований документа. Если значение поля не фиксировано, то оно может быть заменено новым значением в процессе редактирования документа.

Этот атрибут может использоваться с:

- полями даты;
- полями времени;
- полями номеров страниц;
- всеми полями отправителя;
- всеми полями автора.

```
<define name="common-field-fixed-attlist">
    <optional>
```

```
        <attribute name="text:fixed">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

6.7.3 Имя переменной

Используйте атрибут `text:name` для задания имени переменной при ее объявлении, установке значения или отображении. Этот атрибут может использоваться с любым из следующих элементов:

- `<text:variable-decl>`
- `<text:variable-set>`
- `<text:variable-get>`
- `<text:variable-input>`
- `<text:user-field-decl>`
- `<text:user-field-get>`
- `<text:user-field-input>`
- `<text:sequence-decl>`
- `<text:sequence>`

Когда этот атрибут используется для определения имени отображаемой переменной, переменная соответствующего типа и с таким же именем уже должна быть объявлена.

```
<define name="common-field-name-attlist">
    <attribute name="text:name">
        <ref name="variableName"/>
    </attribute>
</define>
```

6.7.4 Описание

Атрибут `text:description` содержит краткое сообщение, которое отображается, когда у пользователя запрашивается ввод данных. Этот атрибут может использоваться с любым из следующих элементов:

- `<text:placeholder>`
- `<text:variable-input>`
- `<text:user-field-input>`

- `<text:text-input>`

```
<define name="common-field-description-attlist">
    <optional>
        <attribute name="text:description">
            <text/>
        </attribute>
    </optional>
</define>
```

6.7.5 Отображение

Атрибут `text:display` поддерживает до трех следующих величин:

- `value`

  Это значение отображает значение поля. Некоторые поля не поддерживают это значение. Тогда атрибут `text:display` только лишь принимает значения `value` или `none`, и `value` или `formula`, соответственно.

- `formula`

  Это значение позволяет отображать формулу, а не значение поля. Некоторые поля не поддерживают это значение. Тогда атрибут `text:display` только принимает значения `value` или `none`, и `value` или `formula`, соответственно.

- `none`

  Несколько полей переменных поддерживают это значение, которое скрывает содержимое поля. Это позволяет задавать значения переменных в одной части документа, а отображать их в другой части документа.

Этот атрибут может использоваться с любым из следующих элементов:

- `<text:variable-set>`
- `<text:variable-get>`
- `<text:variable-input>`
- `<text:user-field-get>`
- `<text:expression>`

```
<define name="common-field-display-value-none-attlist">
    <optional>
        <attribute name="text:display">
            <choice>
                <value>value</value>
```

```
                <value>none</value>
            </choice>
        </attribute>
    </optional>
</define>
<define name="common-field-display-value-formula-none-attlist">
    <optional>
        <attribute name="text:display">
            <choice>
                <value>value</value>
                <value>formula</value>
                <value>none</value>
            </choice>
        </attribute>
    </optional>
</define>
<define name="common-field-display-value-formula-attlist">
    <optional>
        <attribute name="text:display">
            <choice>
                <value>value</value>
                <value>formula</value>
            </choice>
        </attribute>
    </optional>
</define>
```

6.7.6 Формула

Атрибут `text:formula` содержит формулу или выражение для вычисления значения поля. Этот атрибут может использоваться с любым из следующих элементов:

- `<text:variable-set>`
- `<text:user-field-decl>`
- `<text:sequence>`
- `<text:expression>`

Формула должна начинаться с префикса, указывающего на синтаксис и семантику, используемые в формуле.

```
<define name="common-field-formula-attlist">
    <optional>
        <attribute name="text:formula">
            <ref name="formula"/>
        </attribute>
    </optional>
</define>
```

6.7.7 Стиль форматирования

Атрибут `style:data-style-name` ссылается на стиль данных, используемый при форматировании числового значения. Для получения общей информации о стилях см. раздел 14. Для получения более подробной информации по стилям данных см. раздел 14.7.

Для строковых переменных этот атрибут должен быть опущен. В других случаях этот атрибут необходим.

Имя должно соответствовать имени стиля данных.

Этот атрибут может использоваться с любым из следующих элементов:

- `<text:date>`
- `<text:time>`
- `<text:page-number>`
- `<text:variable-set>`
- `<text:variable-get>`
- `<text:variable-input>`
- `<text:user-field-get>`
- `<text:user-field-input>`
- `<text:expression>`

```
<define name="common-field-data-style-name-attlist">
    <optional>
        <attribute name="style:data-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

6.7.8 Стиль форматирования чисел

Числа, используемые в числовых последовательностях, таких как номера страниц или поля последовательности, могут форматироваться в соответствии со стилями чисел, описанными в разделе 12.2. Поддерживаются следующие стили чисел:

- числовой — 1, 2, 3, ... ;
- алфавитный — a, b, c, ... или A, B, C, ... ;
- римский — i, ii, iii, iv, ... или I, II, III, IV, ...

Примечание — Значение этого атрибута может быть любым из [XSLT] ключей форматов чисел `1`, `i`, `I`, `a`, или `A`.

Алфавитный стиль чисел нуждается в дополнительном атрибуте, определяющем способ отображения чисел, которые не могут быть представлены одной буквой. Формат OpenDocument поддерживает:

- синхронизированную буквенную нумерацию, где буквы используются многократно, например aa, bb, cc и т. д.;
- несинхронизированную буквенную нумерацию, например aa, ab, ac и т. д.

См. раздел 12.2 для получения более подробной информации.

```
<define name="common-field-num-format-attlist">
    <optional>
        <ref name="common-num-format-attlist"/>
    </optional>
</define>
```

7 Текстовые индексы

Текст документов OpenDocument может содержать автоматически сформированные индексы. Индекс содержит отсортированный список всех элементов определенного типа, где критерии сортировки (положение в документе, алфавитный и т. д.) и типы элементов (заголовки глав, таблиц, и т. д.) определены типом индекса.

7.1 Индексные метки

Существует три типа индексных меток, соответствующие трем типам индексов, использующих индексные метки:

- индексные метки оглавления;
- определяемые пользователем индексные метки;
- алфавитные индексные метки.

XML-код для индексных меток похож на код для закладок и перекрестных ссылок. Ниже указаны некоторые основные правила для индексных меток:

- каждая индексная метка представлена начальным и конечным элементами;
- оба элемента используют атрибут ID, чтобы определить соответствующее начало и конец метки;
- для каждого начального элемента должен быть определен конечный элемент в том же самом абзаце, с начальным элементом определенным первым;
- атрибуты, связанные с индексной меткой, присоединяются к начальному элементу;
- текст, размещенный между начальным и конечным элементами, является текстом индексной записи;
- атрибуты форматирования индексных меток могут перекрываться.

7.1.1 Индексные метки оглавления

Элемент меток `<text:toc-mark-start>` определяет начало индексной записи оглавления. ID, определяемый атрибутом `text:id`, должен быть уникальным, за исключением соответствующих конечных элементов индексных меток. Для каждого начального элемента должен быть определен конечный элемент в том же самом абзаце, с начальным элементом, определенным первым.

```
<define name="paragraph-content" combine="choice">
    <element name="text:toc-mark-start">
        <ref name="text-toc-mark-start-attrs"/>
    </element>
</define>
```

С элементом `<text:toc-mark-start>` могут быть связаны следующие атрибуты:

- атрибут `text:id`, позволяющий начальному и конечному элементам соответствовать друг другу;
- атрибут `text:outline-level`, устанавливающий уровень структуры индексов оглавления.

```
<define name="text-toc-mark-start-attrs">
    <ref name="text-id"/>
    <ref name="text-outline-level"/>
</define>
<define name="text-outline-level">
    <optional>
        <attribute name="text:outline-level">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
<define name="text-id">
    <attribute name="text:id">
        <ref name="string"/>
    </attribute>
</define>
```

Элемент `<text:toc-mark-end>` определяет окончание индексной записи оглавления. Для каждого конечного элемента должен быть начальный элемент с таким же значением `text:id`, расположенный в том же абзаце, с начальным элементом, определенным первым.

```
<define name="paragraph-content" combine="choice">
    <element name="text:toc-mark-end">
        <ref name="text-id"/>
    </element>
</define>
```

Индексные метки в оглавлении также имеют вариант, который предполагает не включать текст для индексирования. Это реализуется использованием элемента `<text:toc-mark>` с атрибутом `text:string-value` для индекса. В этом случае атрибут `text:id` не требуется, потому что нет стартового и конечного элементов.

```
<define name="paragraph-content" combine="choice">
    <element name="text:toc-mark">
        <attribute name="text:string-value">
            <ref name="string"/>
        </attribute>
        <ref name="text-outline-level"/>
    </element>
</define>
```

7.1.2 Индексные метки, определяемые пользователем

Элемент `<text:user-index-mark-start>` определяет начало определяемой пользователем индексной метки. ID, определяемый атрибутом `text:id`, должен быть уникальным, за исключением соответствующих конечных элементов индексных меток. Для каждого начального элемента должен быть определен конечный элемент, расположенный в том же абзаце, с начальным элементом, определенным первым.

```
<define name="paragraph-content" combine="choice">
    <element name="text:user-index-mark-start">
        <ref name="text-id"/>
        <ref name="text-outline-level"/>
        <ref name="text-index-name"/>
    </element>
</define>
```

Элемент `<text:user-index-mark-end>` обозначает окончание определяемой пользователем индексной метки. Для каждого конечного элемента должен быть начальный элемент с таким же значением text:id, расположенный в том же абзаце, с начальным элементом, определенным первым.

```
<define name="paragraph-content" combine="choice">
    <element name="text:user-index-mark-end">
        <ref name="text-id"/>
        <ref name="text-outline-level"/>
    </element>
</define>
```

Определяемые пользователем индексные метки также имеют вариант, который не предполагает внесения текста для индексирования. Это реализуется использованием элемента `<text:user-index-mark>` с атрибутом

text:string-value для текста индексирования. В этой ситуации атрибут text:id не является обязательным, так как нет соответствующих начального и конечного элементов.

```
<define name="paragraph-content" combine="choice">
    <element name="text:user-index-mark">
        <attribute name="text:string-value">
            <ref name="string"/>
        </attribute>
        <ref name="text-outline-level"/>
        <ref name="text-index-name"/>
    </element>
</define>
```

Имя индекса, определяемого пользователем

Можно использовать больше одного определяемого пользователем индекса. В этом случае для определяемого пользователем индекса должно быть определено имя атрибутом text:index-name. Этот атрибут определяет, какому определяемому пользователем индексу соответствует индексная метка. Если имя не определено, то используется определяемый пользователем индекс, заданный по умолчанию.

```
<define name="text-index-name">
    <attribute name="text:index-name">
        <ref name="string"/>
    </attribute>
</define>
```

7.1.3 Алфавитные индексные метки

Элемент <text:alpha-index-mark-start> определяет начало алфавитной индексной записи. Существует два дополнительных атрибута, которые могут содержать ключи для алфавитных индексов, позволяющие их структурировать. Есть также атрибут логического типа, определяющий их вхождение в главную конструкцию, если имеется несколько равносильных полей.

ID, определяемый атрибутом text:id, должен быть уникальным, за исключением соответствующих конечных элементов индексных меток. Для каждого начального элемента должен быть определен конечный элемент в том же самом абзаце, с начальным элементом, определенным первым.

```
<define name="paragraph-content" combine="choice">
    <element name="text:alphabetical-index-mark-start">
        <ref name="text-id"/>
        <ref name="text-alphabetical-index-mark-attrs"/>
    </element>
```

```
</define>
```

С элементом `<text:toc-mark-start>` могут быть связаны следующие атрибуты:

- атрибут `text:id`, определяющий соответствующие начальные и конечные элементы;
- дополнительные ключи;
- главная запись.

Элемент `<text:alpha-index-mark-end>` определяет окончание алфавитной индексной записи. Для каждого конечного элемента должен быть начальный элемент с таким же значением text:id, расположенный в том же абзаце, с начальным элементом, определенным первым.

```
<define name="paragraph-content" combine="choice">
    <element name="text:alphabetical-index-mark-end">
        <ref name="text-id"/>
    </element>
</define>
```

Алфавитные индексные метки имеют вариант, который не предполагает внесения текста для индексирования. Это обеспечивается использованием элемента `<text:alpha-index-mark>`, с атрибутом `text:string-value` для текста в индексной записи. В этой ситуации атрибут `text:id` не является обязательным, так как нет соответствующих начального и конечного элементов.

```
<define name="paragraph-content" combine="choice">
    <element name="text:alphabetical-index-mark">
        <attribute name="text:string-value">
            <ref name="string"/>
        </attribute>
        <ref name="text-alphabetical-index-mark-attrs"/>
    </element>
</define>
```

Дополнительные ключи

Атрибуты `text:key1` и `text:key2` определяют дополнительные ключи для алфавитных индексов. Если используется только один ключ, то он должен быть определен в атрибуте `text:key1`.

```
<define name="text-alphabetical-index-mark-attrs" combine="interleave">
    <optional>
        <attribute name="text:key1">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
```

```
        <attribute name="text:key2">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Фонетические ключи

Для идеографических языков иногда характерны ситуации, когда нет очевидных или общих способов сортировки символов. Один из общих способов определения алфавитных индексов для них — сортировка на основании их фонетических описаний времени поиска. В файле формата OpenDocument это достигается за счет дополнительных атрибутов для текстового значения и двух ключей для фонетического описания. Для отображения используются значение и ключевые атрибуты, но если фонетические варианты существуют, то они должны использоваться для сортировки индексов.

```
<define name="text-alphabetical-index-mark-attrs" combine="interleave">
    <optional>
        <attribute name="text:string-value-phonetic">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:key1-phonetic">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:key2-phonetic">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Главная запись

Если есть несколько индексных меток в одной записи, одна из этих записей может быть объявлена как главная при помощи атрибута `text:main-entry`.

```
<define name="text-alphabetical-index-mark-attrs" combine="interleave">
    <optional>
        <attribute name="text:main-entry" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

7.1.4 Библиографические индексные метки

Элемент `<text:bibliography-mark>` включает текст и информацию для библиографического индекса. Этот элемент поддерживает атрибуты для всех типов библиографических описаний, которые могут быть в них включены.

```
<define name="paragraph-content" combine="choice">
    <element name="text:bibliography-mark">
        <attribute name="text:bibliography-type">
            <ref name="text-bibliography-types"/>
        </attribute>
        <zeroOrMore>
            <attribute>
                <choice>
                    <name>text:identifier</name>
                    <name>text:address</name>
                    <name>text:annote</name>
                    <name>text:author</name>
                    <name>text:booktitle</name>
                    <name>text:chapter</name>
                    <name>text:edition</name>
                    <name>text:editor</name>
                    <name>text:howpublished</name>
                    <name>text:institution</name>
                    <name>text:journal</name>
                    <name>text:month</name>
                    <name>text:note</name>
                    <name>text:number</name>
                    <name>text:organizations</name>
                    <name>text:pages</name>
                    <name>text:publisher</name>
                    <name>text:school</name>
                    <name>text:series</name>
                    <name>text:title</name>
                    <name>text:report-type</name>
                    <name>text:volume</name>
                    <name>text:year</name>
                    <name>text:url</name>
                    <name>text:custom1</name>
                    <name>text:custom2</name>
                    <name>text:custom3</name>
                    <name>text:custom4</name>
                    <name>text:custom5</name>
                    <name>text:isbn</name>
                    <name>text:issn</name>
                </choice>
                <ref name="string"/>
            </attribute>
        </zeroOrMore>
        <text/>
    </element>
</define>
<define name="text-bibliography-types">
    <choice>
        <value>article</value>
        <value>book</value>
        <value>booklet</value>
        <value>conference</value>
        <value>custom1</value>
        <value>custom2</value>
        <value>custom3</value>
```

```
        <value>custom4</value>
        <value>custom5</value>
        <value>email</value>
        <value>inbook</value>
        <value>incollection</value>
        <value>inproceedings</value>
        <value>journal</value>
        <value>manual</value>
        <value>mastersthesis</value>
        <value>misc</value>
        <value>phdthesis</value>
        <value>proceedings</value>
        <value>techreport</value>
        <value>unpublished</value>
        <value>www</value>
    </choice>
</define>
```

7.2 Структура индексов

Индекс состоит из двух частей: источника индекса и его тела. Каждая из частей описывается собственным элементом, которые в свою очередь разбиваются на два дочерних элемента.

Источник индекса зависит от типа индекса, для которого он будет использоваться. Источник содержит информацию, необходимую для генерирования содержания индекса. Источник индекса не имеет графического отображения.

Тело индекса одинаково для всех индексных типов. Оно содержит текст, сформированный, исходя из данных выбранного источника. Текст, указанный в теле индекса, никоим образом не отличается и не обособляется от текста, используемого в другом месте этой спецификации.

Содержимое тела индекса может быть восстановлено в любое время, исходя из информации, указанной в источнике индекса и остальной части документа. Можно сказать, что источник индекса содержит всю логическую информацию об индексе, в то время как тело индекса содержит его исполнение. Средства, извлекающие информацию о структуре документа, могут просматривать только источник индекса, в то время как программа может просматривать только тело индекса.

7.2.1 Источник индекса

Элемент источника индекса содержит информацию, необходимую для генерации тела индекса. В дополнение к ряду флагов, которые определяют, какая

информация включается в индекс, источник индекса содержит ряд шаблонов. Такие шаблоны определяют, как должно быть представлено поле, включенное в индекс.

Например, оглавление может быть представлено следующим образом:

1 Введение.....7
1.1 Пространства имен.....7
1.2 Префикс схемы Relax-NG.....8

Источник индекса для оглавления содержал бы флаги, которые указывали на то, что заголовок глав должен содержать хотя бы 2 уровня. Шаблоны индекса определили бы, что конструкция состоит из номера главы, пробела, названия главы, табуляции (заполненной символом «.») и номера страницы.

Различные шаблоны индексов описаны вместе с их элементами. Использование элементов шаблона индексов описано в разделе 7.12.

Различные элементы источников индексов описаны вместе с соответствующими элементами индексов.

7.2.2 Тело индекса

Тело индекса содержит текущее текстовое наполнения индекса. Его формат такой же, как и формат регулярного текста в этой спецификации, например текст разделов, за исключением разделов индексного заголовка.

```
<define name="text-index-body">
    <element name="text:index-body">
        <zeroOrMore>
            <ref name="index-content-main"/>
        </zeroOrMore>
    </element>
</define>
<define name="index-content-main">
    <choice>
        <ref name="text-content"/>
        <ref name="text-index-title"/>
    </choice>
</define>
```

7.2.3 Заголовок индекса

Заголовок индекса обычно содержится в собственном разделе, поскольку это дает возможность размещения заголовка индекса в популярных макетах поперек всей страницы, тогда как сам индекс занимает две колонки.

```
<define name="text-index-title">
```

```
    <element name="text:index-title">
        <ref name="sectionAttr"/>
        <zeroOrMore>
            <ref name="index-content-main"/>
        </zeroOrMore>
    </element>
</define>
```

7.3 Оглавление

Оглавление позволяет пользователю перемещаться по содержанию документа. Обычно оно размещается в начале документа, содержит заголовки глав и соответствующие им номера страниц. Пример оглавления можно посмотреть в начале этого документа.

Пункты, которые могут быть перечислены в оглавлении:

- заголовки (определенные структурой документа) до выбираемого уровня;
- индексные метки оглавления;
- форматирование абзаца с выбираемым набором стилей.

Оглавление представляется элементом `<text:table-of-content>`, который поддерживает те же атрибуты стиля (и класса), что и разделы текста (см. раздел 4.4).

```
<define name="text-table-of-content">
    <element name="text:table-of-content">
        <ref name="sectionAttr"/>
        <ref name="text-table-of-content-source"/>
        <ref name="text-index-body"/>
    </element>
</define>
```

7.3.1 Источник оглавления

Элемент `<text:table-of-content-source>` определяет, как генерируется оглавление. Он также определяет, как собраны записи оглавления.

Элемент `<text:table-of-content-source>` включает:

- дополнительный шаблон для заголовка индекса;
- дополнительные шаблоны для индексных записей, по одному для уровня;
- необязательный список стилей, которые используются для индексных записей.

```
<define name="text-table-of-content-source">
    <element name="text:table-of-content-source">
        <ref name="text-table-of-content-source-attlist"/>
        <optional>
            <ref name="text-index-title-template"/>
```

```
2749        </optional>
2750        <zeroOrMore>
2751            <ref name="text-table-of-content-entry-template"/>
2752        </zeroOrMore>
2753        <zeroOrMore>
2754            <ref name="text-index-source-styles"/>
2755        </zeroOrMore>
2756    </element>
2757 </define>
```

С элементом `<text:table-of-content-source>` могут быть связаны следующие атрибуты:

- уровень структуры;
- использование структуры;
- использование индексных меток;
- использование стилей источников индексов;
- источник индексов;
- относительное положение табулостопов.

Уровень структуры

Атрибут `text:outline-level` определяет, какой уровень структуры используется для формирования оглавления.

Значения этого атрибута должны быть целыми числами больше нуля. Если этот атрибут пропущен, то все уровни структуры используются по умолчанию.

```
2758 <define name="text-table-of-content-source-attlist" combine="interleave">
2759     <optional>
2760         <attribute name="text:outline-level">
2761             <choice>
2762                 <ref name="positiveInteger"/>
2763             </choice>
2764         </attribute>
2765     </optional>
2766 </define>
```

Использование структуры

Атрибут `text:use-outline-level` определяет, используются ли заголовки для генерации индексных полей. Если значение равно `true`, то записи оглавления формируются из заголовков. Атрибут `text:outline-level` определяет уровень, до которого будут включены заголовки (см. раздел 7.1 для получения более подробной информации об индексных метках).

```
2767 <define name="text-table-of-content-source-attlist" combine="interleave">
2768     <optional>
```

```
2769        <attribute name="text:use-outline-level" a:defaultValue="true">
2770            <ref name="boolean"/>
2771        </attribute>
2772    </optional>
2773 </define>
```

Использование индексных меток

Атрибут `text:use-index-marks` определяет, используются или нет индексные метки для формирования индексных полей. Если значение равно `true`, то в оглавление включаются записи, сформированные индексными метками оглавления. Атрибут `text:outline-level` определяет уровень, до которого будут включены индексные метки (см. раздел 7.1 для получения более подробной информации об индексных метках).

```
2774 <define name="text-table-of-content-source-attlist" combine="interleave">
2775     <optional>
2776         <attribute name="text:use-index-marks">
2777             <ref name="boolean"/>
2778         </attribute>
2779     </optional>
2780 </define>
```

Использование стилей источников индексов

Атрибут `text:use-index-source-styles` определяет, сформированы или нет индексные записи из абзацев, отформатированных определенными стилями абзацев. Если значение равно `true`, то в оглавление добавляется запись для каждого абзаца, в форматировании которого используется один из стилей, определенных в элементе `<text:index-source-style>`. Атрибут `text:outline-level` определяет уровень, до которого используются стили источников индексов.

```
2781 <define name="text-table-of-content-source-attlist" combine="interleave">
2782     <optional>
2783         <attribute name="text:use-index-source-styles">
2784             <ref name="boolean"/>
2785         </attribute>
2786     </optional>
2787 </define>
```

Охват индекса

Атрибут `text:index-scope` определяет, формируется ли содержание для всего документа или только для текущей главы.

```
2788 <define name="text-table-of-content-source-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="text:index-scope">
            <choice>
                <value>document</value>
                <value>chapter</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Относительное положение табулостопов

Атрибут `text:relative-tab-stop-position` определяет, относительно левого поля или левого отступа, определенного стилем абзаца, отсчитывается положение табулостопов. Это обычно используется для копирования конфигурации записей всех уровней структуры, поскольку относительная позиция табуляции не должна быть настроена так же, как в соответствующем абзаце.

```
<define name="text-table-of-content-source-attlist" combine="interleave">
    <optional>
        <attribute name="text:relative-tab-stop-position">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

7.3.2 Шаблоны записей оглавления

Элемент `<text:table-of-content-entry-template>` определяет формат индексной записи для определенного уровня структуры. Для каждого оглавления может быть не более одного элемента для любого уровня структуры (см. ниже).

```
<define name="text-table-of-content-entry-template">
    <element name="text:table-of-content-entry-template">
        <ref name="text-table-of-content-entry-template-attlist"/>
        <zeroOrMore>
            <ref name="text-table-of-content-children"/>

        </zeroOrMore>
    </element>
</define>
```

Шаблоны записей оглавления поддерживают следующие текстовые элементы:

- глава и номер страницы;
- текст ссылки;
- диапазон текста;
- табулостопы;

- начало и конец гиперссылки.

```
<define name="text-table-of-content-children">
    <choice>
        <ref name="text-index-entry-chapter"/>
        <ref name="text-index-entry-page-number"/>
        <ref name="text-index-entry-text"/>
        <ref name="text-index-entry-span"/>
        <ref name="text-index-entry-tab-stop"/>
        <ref name="text-index-entry-link-start"/>
        <ref name="text-index-entry-link-end"/>
    </choice>
</define>
```

С элементом `<text:table-of-content-entry-template>` могут быть связаны следующие атрибуты:

- шаблон уровня структуры;
- стиль абзаца.

Шаблон уровня структуры

Этот атрибут определяет, к какому уровню структуры применяется конфигурация записи. Уровни структуры должны быть уникальными для элементов шаблона в пределах одного источника индекса.

```
<define name="text-table-of-content-entry-template-attlist"
        combine="interleave">
    <attribute name="text:outline-level">
        <ref name="positiveInteger"/>
    </attribute>
</define>
```

Стиль абзаца

Атрибут `text:style-name` определяет стиль абзаца, используемый в этом шаблоне.

```
<define name="text-table-of-content-entry-template-attlist"
        combine="interleave">
    <attribute name="text:style-name">
        <ref name="styleNameRef"/>
    </attribute>
</define>
```

7.4 Индексы иллюстраций

Индексы иллюстраций — это список всех рисунков и изображений в текущем документе или главе. Входные индексы могут быть получены из заголовков иллюстраций или их названий.

С элементом `<text:illustration-index>` может быть связан следующий атрибут:

- `text:style-name`

  Этот атрибут определяет стиль раздела, используемый для индекса иллюстраций.

```
<define name="text-illustration-index">
    <element name="text:illustration-index">
        <ref name="sectionAttr"/>
        <ref name="text-illustration-index-source"/>
        <ref name="text-index-body"/>
    </element>
</define>
```

7.4.1 Индексы источников иллюстраций

Элемент `<text:illustration-index-source>` устанавливает, как формируется индекс иллюстраций.

```
<define name="text-illustration-index-source">
    <element name="text:illustration-index-source">
        <ref name="text-illustration-index-source-attrs"/>
        <optional>
            <ref name="text-index-title-template"/>
        </optional>
        <optional>
            <ref name="text-illustration-index-entry-template"/>
        </optional>
    </element>
</define>
```

С элементом `<text:illustration-index-source>` могут быть связаны следующие атрибуты:

- использование подписи;
- имя последовательности подписей;
- формат последовательности подписей;
- охват индекса — атрибут указывает, устанавливается ли индекс для всего документа или только для текущей главы;
- `text:relative-tab-stop-position` — атрибут определяет, относительно левого края или левого отступа отсчитывается положение табулостопов.

```
<define name="text-illustration-index-source-attrs" combine="interleave">
    <ref name="text-index-scope-attr"/>
</define>
<define name="text-index-scope-attr">
    <optional>
        <attribute name="text:index-scope" a:defaultValue="document">
```

```
            <choice>
                <value>document</value>
                <value>chapter</value>
            </choice>
        </attribute>
    </optional>
</define>
<define name="text-illustration-index-source-attrs" combine="interleave">
    <ref name="text-relative-tab-stop-position-attr"/>
</define>
<define name="text-relative-tab-stop-position-attr">
    <optional>
        <attribute name="text:relative-tab-stop-position"
                   a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Использование подписи

Каждый объект, находящийся в документе, имеет свое название (имя). Помимо этого, рисунки также имеют подпись. Подпись рисунка или название рисунка могут быть объединены для индекса иллюстрации.

```
<define name="text-illustration-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:use-caption" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Имя последовательности подписей

Подписи ассоциируются с именем последовательности. Если атрибут `text:use-caption` установлен в `true`, то этот атрибут должен использоваться для определения последовательности, с которым связаны подписи.

Если этот атрибут опущен, то по умолчанию используется последовательность в зависимости от типа объектов, например, последовательность «Иллюстрация» используется для иллюстраций.

```
<define name="text-illustration-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:caption-sequence-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Формат последовательности подписей

Если индексные записи иллюстраций получены из подписей рисунков, то этот атрибут должен использоваться для определения формата записей.

```
<define name="text-illustration-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:caption-sequence-format">
            <choice>
                <value>text</value>
                <value>category-and-value</value>
                <value>caption</value>
            </choice>
        </attribute>
    </optional>
</define>
```

7.4.2 Шаблон индексных записей иллюстраций

Шаблон индексных записей иллюстраций определяет формат индексной записи для определенного уровня структуры.

```
<define name="text-illustration-index-entry-template">
    <element name="text:illustration-index-entry-template">
        <ref name="text-illustration-index-entry-content"/>
    </element>
</define>
<define name="text-illustration-index-entry-content">
    <ref name="text-illustration-index-entry-template-attrs"/>
    <zeroOrMore>
        <choice>
            <ref name="text-index-entry-page-number"/>
            <ref name="text-index-entry-text"/>
            <ref name="text-index-entry-span"/>
            <ref name="text-index-entry-tab-stop"/>
        </choice>
    </zeroOrMore>
</define>
```

С элементом `<text:illustration-index-entry-template>` может быть связан следующий атрибут:

- стиль абзаца.

Стиль абзаца

Данный атрибут определяет стиль абзаца, используемый данным шаблоном.

```
<define name="text-illustration-index-entry-template-attrs">
    <attribute name="text:style-name">
        <ref name="styleNameRef"/>
    </attribute>
</define>
```

7.5 Индексы таблиц

Индекс таблиц перечисляет все таблицы в текущем документе или главе. Он обрабатывается точно так же, как и индекс иллюстраций.

```
<define name="text-table-index">
    <element name="text:table-index">
        <ref name="sectionAttr"/>
        <ref name="text-table-index-source"/>
        <ref name="text-index-body"/>
    </element>
</define>
```

7.5.1 Источник индекса таблиц

Элемент `<text:table-index-source>` определяет, как формируется индекс таблиц.

Атрибуты, которые могут быть связаны с этим элементом, те же, что могут быть связаны с элементом `<text:illustration-index-source>` (см. раздел 7.4.1 для получения детальной информации об этих атрибутах).

```
<define name="text-table-index-source">
    <element name="text:table-index-source">
        <ref name="text-illustration-index-source-attrs"/>
        <optional>
            <ref name="text-index-title-template"/>
        </optional>
        <optional>
            <ref name="text-table-index-entry-template"/>
        </optional>
    </element>
</define>
```

7.5.2 Шаблон индексных записей таблицы

Шаблон индексных записей таблицы определяет формат индексной записи для определенного уровня структуры.

Атрибуты, которые связаны с этим элементом, те же, что могут быть связаны с элементом `<text:illustration-index-entry-template>` (см. раздел 7.4.2 для получения детальной информации об этих атрибутах).

```
<define name="text-table-index-entry-template">
    <element name="text:table-index-entry-template">
        <ref name="text-illustration-index-entry-content"/>
    </element>
</define>
```

7.6 Индексы объектов

Индекс объектов перечисляет все объекты в текущем документе или главе. Он формируется из записей объектов известных типов.

```
<define name="text-object-index">
    <element name="text:object-index">
        <ref name="sectionAttr"/>
        <ref name="text-object-index-source"/>
        <ref name="text-index-body"/>
    </element>
</define>
```

7.6.1 Источник индекса объектов

Элемент `<text:object-index-source>` определяет типы объектов, включенных в индекс объектов. Он также поддерживает стандартные атрибуты источников индексов.

```
<define name="text-object-index-source">
    <element name="text:object-index-source">
        <ref name="text-object-index-source-attrs"/>
        <optional>
            <ref name="text-index-title-template"/>
        </optional>
        <optional>
            <ref name="text-object-index-entry-template"/>
        </optional>
    </element>
</define>
```

С элементом `<text:object-index-source>` могут быть связаны следующие атрибуты:

- использование атрибутов `text:use-*-objects`;
- охват индексов (см. раздел 7.4.1) — атрибут определяет, применяется ли индекс ко всему документу или только к текущей главе;
- относительная позиция табулостопов (см. раздел 7.4.1) — атрибут определяет, относительно левого края или левого отступа отсчитывается положение табулостопов.

```
<define name="text-object-index-source-attrs" combine="interleave">
    <ref name="text-index-scope-attr"/>
</define>
<define name="text-object-index-source-attrs" combine="interleave">
    <ref name="text-relative-tab-stop-position-attr"/>
</define>
```

Использование атрибутов

Атрибуты `text:use-*-objects` определяют, какой тип объектов включать в индекс объектов. Существуют атрибуты для каждого типа объектов:

- `text:use-spreadsheet-objects`
- `text:use-draw-objects`
- `text:use-chart-objects`
- `text:use-math-objects`

Другие объекты, включенные или пропущенные, используют атрибут:

- `text:use-other-objects`

```
<define name="text-object-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:use-spreadsheet-objects" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
<define name="text-object-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:use-math-objects" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
<define name="text-object-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:use-draw-objects" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
<define name="text-object-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:use-chart-objects" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
<define name="text-object-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:use-other-objects" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

7.6.2 Шаблон индексных записей объекта

Шаблон индексных записей объекта определяет формат индексной записи для определенного уровня структуры.

```
<define name="text-object-index-entry-template">
    <element name="text:object-index-entry-template">
        <ref name="text-illustration-index-entry-content"/>
    </element>
</define>
```

Атрибуты, которые могут быть связаны с этим элементом, те же, что могут быть связаны с элементом `<text:illustration-index-entry-template>` (см. раздел 7.4.2 для получения детальной информации об этих атрибутах).

7.7 Индексы, определяемые пользователем

Индексы, определяемые пользователем, комбинируют возможности индексов, описанных ранее в этом разделе. Определяемый пользователем индекс может содержать записи из следующих источников:

- индексные метки;
- отформатированные определенными стилями абзацы;
- таблицы, рисунки или объекты;
- текстовые фреймы.

Элемент `<text:user-index>` представляет определяемый пользователем индекс.

```
<define name="text-user-index">
    <element name="text:user-index">
        <ref name="sectionAttr"/>
        <ref name="text-user-index-source"/>
        <ref name="text-index-body"/>
    </element>
</define>
```

7.7.1 Источник индекса, определяемого пользователем

Элемент `<text:user-index-source>` может содержать несколько атрибутов, которые определяют, как сформированы индексные записи. Этот элемент также поддерживает атрибут, определяющий, как сформированы уровни структуры индексных записей.

Форматирование абзаца, которое используется как индексные метки, заключено в элементах `<text:index-source-styles>` точно так же, как и в элементах `<text:table-of-content-source>`.

```
<define name="text-user-index-source">
    <element name="text:user-index-source">
        <ref name="text-user-index-source-attr"/>
        <optional>
```

```
3023            <ref name="text-index-title-template"/>
3024        </optional>
3025        <zeroOrMore>
3026            <ref name="text-user-index-entry-template"/>
3027        </zeroOrMore>
3028        <zeroOrMore>
3029            <ref name="text-index-source-styles"/>
3030        </zeroOrMore>
3031    </element>
3032 </define>
```

С элементом `<text:user-index-source>` могут быть связаны следующие атрибуты.

- Использование атрибутов — `text:use-*`
- Копирование уровня структуры.
- Охват индексов (см. раздел 7.4.1).

  Этот атрибут определяет, применяется ли индекс ко всему документу или только к текущей главе.
- Имя индекса.

  Для поддержки нескольких определяемых пользователем индексов с различным содержимым, определяемые пользователем метки индексов имеют атрибут `text:index-name`. Этот атрибут может быть использован с элементом `<text:user-index-source>`, чтобы определить, какие индексные метки относятся к текущему индексу.
- Относительная позиция табулостопов (см. раздел 7.4.1).

  Данный атрибут определяет, относительно левого края или левого отступа отсчитывается положение табулостопов.

```
3033 <define name="text-user-index-source-attr" combine="interleave">
3034     <ref name="text-index-scope-attr"/>
3035     <ref name="text-relative-tab-stop-position-attr"/>
3036     <attribute name="text:index-name">
3037         <ref name="string"/>
3038     </attribute>
3039 </define>
```

Использование атрибутов

Атрибуты `text:use-*` определяют, какие записи включить в пользовательский индекс. Существуют следующие атрибуты:

- `text:use-index-marks`
- `text:use-graphics`
- `text:use-tables`

- text:use-floating-frames
- text:use-objects

```
<define name="text-user-index-source-attr" combine="interleave">
    <optional>
        <attribute name="text:use-index-marks" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:use-graphics" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:use-tables" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:use-floating-frames"
                    a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:use-objects" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Копирование уровня структуры

Этот атрибут может принимать значение `true` или `false`.

Если установлено значение `true`, то поля собираются на уровне структуры элемента источника, к которым они относятся.

Если установлено значение `false`, все записи индекса собираются в главном уровне структуры. Например, если изображение появляется в разделе 1.2.3, то поле изображения располагается на уровне структуры 3.

```
<define name="text-user-index-source-attr" combine="interleave">
    <optional>
        <attribute name="text:copy-outline-levels"
                    a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

7.7.2 Шаблоны индексных записей, определяемых пользователем

Шаблоны индексных записей, определяемых пользователем, поддерживают элементы записей для номера главы, номера страницы, текста записи, диапазона текста и табулостопов.

```
<define name="text-user-index-entry-template">
    <element name="text:user-index-entry-template">
        <ref name="text-user-index-entry-template-attrs"/>
        <zeroOrMore>
            <choice>
                <ref name="text-index-entry-chapter"/>
                <ref name="text-index-entry-page-number"/>
                <ref name="text-index-entry-text"/>
                <ref name="text-index-entry-span"/>
                <ref name="text-index-entry-tab-stop"/>
            </choice>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<text:user-index-entry-template>` могут быть связаны следующие атрибуты:

- уровень структуры шаблона;
- стиль абзаца.

Уровень структуры шаблона

Атрибут `text:outline-level` определяет, к какому уровню структуры относится данная конфигурация.

Все элементы `<text:outline-level>`, включенные в предыдущий элемент, определяют различные уровни структуры.

```
<define name="text-user-index-entry-template-attrs" combine="interleave">
    <attribute name="text:outline-level">
        <ref name="positiveInteger"/>
    </attribute>
</define>
```

Стиль абзаца

Атрибут `text:style-name` определяет стиль абзаца, используемый в шаблоне.

```
<define name="text-user-index-entry-template-attrs" combine="interleave">
    <attribute name="text:style-name">
        <ref name="styleNameRef"/>
    </attribute>
</define>
```

7.8 Алфавитные индексы

Записи алфавитных индексов формируются только из индексных меток.

```
<define name="text-alphabetical-index">
    <element name="text:alphabetical-index">
        <ref name="sectionAttr"/>
        <ref name="text-alphabetical-index-source"/>
        <ref name="text-index-body"/>
    </element>
</define>
```

7.8.1 Источник алфавитного индекса

Элемент `<text:alphabetical-index-source>` определяет, как формируется алфавитный индекс.

```
<define name="text-alphabetical-index-source">
    <element name="text:alphabetical-index-source">
        <ref name="text-alphabetical-index-source-attrs"/>
        <optional>
            <ref name="text-index-title-template"/>
        </optional>
        <zeroOrMore>
            <ref name="text-alphabetical-index-entry-template"/>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<text:alphabetical-index-source>` могут быть связаны следующие атрибуты:

- без учета регистра;
- имя стиля основной записи;
- алфавитные разделители;
- атрибуты объединенных записей;
- использование ключей в качестве записей;
- записи заглавными буквами;
- записи с разделителями в виде запятых;
- сортировка по языку, стране и алгоритму;
- охват индекса (см. раздел 7.4.1) — атрибут определяет, относится ли индекс ко всему документу или только к текущей главе;
- относительная позиция табулостопов (см. раздел 7.4.1) — атрибут определяет, относительно левого поля или левого отступа отсчитывается положение табулостопов.

```
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <ref name="text-index-scope-attr"/>
```

```
    <ref name="text-relative-tab-stop-position-attr"/>
</define>
```

Без учета регистра

Атрибут `text:ignore-case` определяет, игнорируется ли верхний регистр записи. Если значение равно `true`, то верхний регистр игнорируется, и записи, которые различаются только регистром, считаются одинаковыми. Если значение равно `false`, то учитывается разница между заглавными и строчными буквами.

```
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:ignore-case" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Имя стиля основной записи

Атрибут `text:main-entry-style-name` определяет стиль символов, применяемый для основных записей. Подзаписи форматируются с использованием стиля символов по умолчанию, определенного стилем абзаца записей.

```
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:main-entry-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Алфавитные разделители

Атрибут `text:alphabetical-separators` определяет, будут ли группироваться записи, начинающиеся с одной буквы, и отделяться от записей, начинающихся с другой буквы, и т. д.

Значение этого атрибута может принимать значение `true` или `false`.

Если значение `true`, то все записи, начинающиеся с одинаковых букв, группируются. Индекс включает заголовки для каждой секции, например А для всех записей, начинающихся на букву А, Б для всех записей, начинающихся с буквы Б, и т. д.

```
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <optional>
```

```
        <attribute name="text:alphabetical-separators" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Объединение записей

Есть несколько вариантов обработки в общей ситуации, когда существуют множественные индексные записи для одинаковых слов или фраз, такие как:

- многократная запись для одного и того же слова может быть объединена в одну, с использованием атрибута `text:combine-entries` ;
- страницы, на которые ссылаются сгруппированные записи, могут быть отформатированы:
  - как диапазон чисел разделенных дефисом при помощи атрибута `text:combine-entries-with-dash` ;
  - как начальный номер с меткой pp, или соответствующей меткой для выбранного языка, при помощи атрибута `text:combine-entries-with-pp` .

```
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:combine-entries" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:combine-entries-with-dash"
                   a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="text:combine-entries-with-pp" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Пример — комбинирование индексных записей.

Индексная метка для слова «XML» находится на страницах 45, 46, 47, и 48. Записи могут быть отформатированы следующими способами:

Формат записи	*Результат*
Отдельные записи	XML 45 XML 46 и т.д.
Простая комбинированная	XML 45, 46, 47, 48

запись	
Записи, объединенные через дефис	XML 45-48
Сгруппированная запись с меткой pp	XML 45pp

Использование ключей в качестве записей

В дополнение к ключевому слову, индексные метки могут иметь до двух ключей. Если значение этого атрибута равно `true`, то ключи используются как дополнительные записи. Если значение этого атрибута `false`, то ключи используются как подзаписи.

```
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:use-keys-as-entries" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Записи заглавными буквами

Атрибут `text:capitalize-entries` определяет, должны ли индексные записи отображаться заглавными буквами.

```
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:capitalize-entries" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Записи с разделителями в виде запятых

Атрибут `text:comma-separated` определяет, как рассматривать многократные записи индекса. Вместо того чтобы перечислять каждое вхождение индекса на отдельной строке, многократные записи могут быть перечислены в одной строке, разделяясь запятой. Если значение этого атрибута равно `true`, то многократные записи перечисляются в одной строке. По умолчанию значение этого атрибута равно `false`, и каждая индексная запись указывается на отдельной строке.

```
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <optional>
```

```
        <attribute name="text:comma-separated" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Сортировка по языку, стране и алгоритму

Если индексные записи должны быть отсортированы, то данные атрибуты могут быть использованы для определения принципа сортировки. Атрибуты страны и языка определяют местную специфику сортировки. Для некоторых местных специфик применяют разнообразные алгоритмы сортировки. В этом случае атрибут алгоритма может быть использован для определения названия определенного алгоритма.

```
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="fo:language">
            <ref name="languageCode"/>
        </attribute>
    </optional>
</define>
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="fo:country">
            <ref name="countryCode"/>
        </attribute>
    </optional>
</define>
<define name="text-alphabetical-index-source-attrs" combine="interleave">
    <optional>
        <attribute name="text:sort-algorithm">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

7.8.2 Файл с автоматическими метками

Алфавитный индекс поддерживает так называемый файл с автоматическими метками. Такой файл содержит список терминов, и каждое вхождение термина включается в алфавитный индекс. Файл с автоматическими алфавитными метками декларируется как часть текстовых объявлений (см. раздел 4.7). Данный элемент — это XLink, указывающий на ресурс, содержащий список терминов.

```
<define name="text-alphabetical-index-auto-mark-file">
    <element name="text:alphabetical-index-auto-mark-file">
        <attribute name="xlink:href">
            <ref name="anyURI"/>
        </attribute>
        <optional>
            <attribute name="xlink:type" a:defaultValue="simple">
```

```
                <value>simple</value>
            </attribute>
        </optional>
    </element>
</define>
```

7.8.3 Шаблон алфавитных индексных записей

Алфавитные индексы поддерживают три уровня — один уровень для главной индексной записи и два дополнительных уровня для ключей, ассоциированных с индексными записями. Алфавитные индексы также используют шаблон записи для алфавитного разделения.

```
<define name="text-alphabetical-index-entry-template">
    <element name="text:alphabetical-index-entry-template">
        <ref name="text-alphabetical-index-entry-template-attrs"/>
        <zeroOrMore>
            <choice>
                <ref name="text-index-entry-chapter"/>
                <ref name="text-index-entry-page-number"/>
                <ref name="text-index-entry-text"/>
                <ref name="text-index-entry-span"/>
                <ref name="text-index-entry-tab-stop"/>
            </choice>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<text:alphabetical-index-entry-template>` могут быть связаны следующие атрибуты:

- уровень структуры шаблона;
- стиль абзаца.

Уровень структуры шаблона

Этот атрибут определяет, применяется ли шаблон к одному из трех уровней (1, 2 или 3) или алфавитному разделителю.

```
<define name="text-alphabetical-index-entry-template-attrs"
        combine="interleave">
    <attribute name="text:outline-level">
        <choice>
            <value>1</value>
            <value>2</value>
            <value>3</value>
            <value>separator</value>
        </choice>
    </attribute>
</define>
```

Стиль абзаца

Атрибут `text:style-name` определяет стиль абзаца, используемый в шаблоне.

```
<define name="text-alphabetical-index-entry-template-attrs"
        combine="interleave">
    <attribute name="text:style-name">
        <ref name="styleNameRef"/>
    </attribute>
</define>
```

7.9 Библиография

В библиографическом индексе собираются записи из библиографических индексных меток. Элемент `<text:bibliography>` представляет библиографию.

```
<define name="text-bibliography">
    <element name="text:bibliography">
        <ref name="sectionAttr"/>
        <ref name="text-bibliography-source"/>
        <ref name="text-index-body"/>
    </element>
</define>
```

7.9.1 Источник библиографического индекса

Элемент `<text:bibliography-source>` определяет, как формируется библиография.

```
<define name="text-bibliography-source">
    <element name="text:bibliography-source">
        <optional>
            <ref name="text-index-title-template"/>
        </optional>
        <zeroOrMore>
            <ref name="text-bibliography-entry-template"/>
        </zeroOrMore>
    </element>
</define>
```

7.9.2 Шаблон библиографических записей

Шаблоны библиографических записей поддерживают элементы записей для библиографических данных, диапазонов текста и табулостопов. Для каждого типа записи существует только один шаблон записей.

```
<define name="text-bibliography-entry-template">
    <element name="text:bibliography-entry-template">
        <ref name="text-bibliography-entry-template-attrs"/>
        <zeroOrMore>
            <choice>
                <ref name="text-index-entry-span"/>
```

```
                <ref name="text-index-entry-tab-stop"/>
                <ref name="text-index-entry-bibliography"/>
            </choice>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<text:bibliography-entry-template>` могут быть связаны следующие атрибуты:

- тип библиографии;
- стиль абзаца.

Тип библиографии

Этот атрибут определяет, к какому типу библиографической записи применяется шаблон. Этот атрибут должен быть уникальным среди всех элементов `<text:bibliography-type>` в пределах того же самого исходного элемента.

```
<define name="text-bibliography-entry-template-attrs" combine="interleave">
    <attribute name="text:bibliography-type">
        <ref name="text-bibliography-types"/>
    </attribute>
</define>
```

Стиль абзаца

Атрибут `text:style-name` определяет стиль абзаца, используемый в шаблоне.

```
<define name="text-bibliography-entry-template-attrs" combine="interleave">
    <attribute name="text:style-name">
        <ref name="styleNameRef"/>
    </attribute>
</define>
```

7.10 Стили источника индекса

Некоторые индексы могут собирать записи из глав, отформатированных с помощью определенного стиля абзаца. Элемент `<text:index-source-styles>` содержит все элементы `<text:index-source-style>` для задания определенного уровня структуры. Атрибут `text:outline-levels` определяет, на каком уровне структуры выстраивать индексные записи, собранные из соответствующих стилей абзацев. Может быть только один элемент `<text:index-source-style>` для каждого уровня структуры.

```
<define name="text-index-source-styles">
    <element name="text:index-source-styles">
        <attribute name="text:outline-level">
            <ref name="positiveInteger"/>
        </attribute>
        <zeroOrMore>
            <ref name="text-index-source-style"/>
        </zeroOrMore>
    </element>
</define>
```

7.10.1 Стиль источника индекса

Все абзацы, отформатированные при помощи стиля или класса, указанного в элементе `<text:index-source-style>`, включаются в индекс.

```
<define name="text-index-source-style">
    <element name="text:index-source-style">
        <attribute name="text:style-name">
            <ref name="styleName"/>
        </attribute>
        <empty/>
    </element>
</define>
```

7.11 Шаблон заголовков индекса

Элемент `<text:index-title-template>` определяет стиль и содержание заголовка индекса. Только один элемент `<text:index-title-template>` может содержаться в элементе `<text:table-of-content-source>`.

```
<define name="text-index-title-template">
    <element name="text:index-title-template">
        <optional>
            <attribute name="text:style-name">
                <ref name="styleNameRef"/>
            </attribute>
        </optional>
        <text/>
    </element>
</define>
```

7.12 Записи индексного шаблона

Существует восемь типов индексных полей:

- информация о главе;
- текст записи;
- номер станицы;
- фиксированная строка;
- библиографическая информация;

- табулостоп;
- начало и конец гиперссылки.

7.12.1 Информация о главе

Элемент `<text:index-entry-chapter>` показывает номер главы индексной записи. Стиль символов номера главы может быть включен в элемент индексной записи как атрибут `text:style-name`.

```
<define name="text-index-entry-chapter">
    <element name="text:index-entry-chapter">
        <optional>
            <attribute name="text:style-name">
                <ref name="styleNameRef"/>
            </attribute>
        </optional>
        <ref name="text-index-entry-chapter-attrs"/>
    </element>
</define>
```

Примечание — Этот элемент может отображать только номер главы. Для отображения названия главы нужно использовать элементы `<text:index-entry-text>`.

Отображение формата главы

Атрибут `text:display` отображает номер главы, название главы или оба эти значения.

```
<define name="text-index-entry-chapter-attrs">
    <optional>
        <attribute name="text:display" a:defaultValue="number">
            <choice>
                <value>name</value>
                <value>number</value>
                <value>number-and-name</value>
            </choice>
        </attribute>
    </optional>
</define>
```

7.12.2 Текст записи

Элемент `<text:index-entry-text>` отображает текст индексной записи, например название главы, если поле выведено из заголовка, или фразу, содержащуюся в индексной метке, если запись получена из индексной метки. Стиль символов текста записи может быть включен в элемент индексной записи как атрибут `text:style-name`.

```
<define name="text-index-entry-text">
    <element name="text:index-entry-text">
        <optional>
            <attribute name="text:style-name">
                <ref name="styleNameRef"/>
            </attribute>
        </optional>
    </element>
</define>
```

7.12.3 Номер страницы

Элемент `<text:index-entry-page-number>` отображает номер страницы, на которой располагается индексная запись. Стиль абзаца номера страницы может быть включен в элемент индексной записи как атрибут `text:style-name`.

```
<define name="text-index-entry-page-number">
    <element name="text:index-entry-page-number">
        <optional>
            <attribute name="text:style-name">
                <ref name="styleNameRef"/>
            </attribute>
        </optional>
    </element>
</define>
```

7.12.4 Фиксированная строка

Элемент `<text:index-entry-span>` представляет фиксированную строку в пределах индексной записи. Стиль символа текста записи может быть включен в элемент индексной записи как атрибут `text:style-name`. В отличие от элемента `<text:span>`, элемент `<text:index-entry-span>` не имеет никаких дочерних элементов.

```
<define name="text-index-entry-span">
    <element name="text:index-entry-span">
        <optional>
            <attribute name="text:style-name">
                <ref name="styleNameRef"/>
            </attribute>
        </optional>
        <text/>
    </element>
</define>
```

7.12.5 Библиографическая информация

Элемент `<text:index-entry-bibliography>` представляет библиографические данные в шаблонах индексных полей.

```
<define name="text-index-entry-bibliography">
```

```
    <element name="text:index-entry-bibliography">
        <ref name="text-index-entry-bibliography-attrs"/>
    </element>
</define>
```

С элементом `<text:index-entry-bibliography>` могут быть связаны следующие атрибуты:

- `text:style-name`
- `text:bibliography-data-field`

Имя стиля текста

Атрибут `text:style-name` определяет стиль для отображения записи.

```
<define name="text-index-entry-bibliography-attrs" combine="interleave">
    <optional>
        <attribute name="text:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Поле идентификатора библиографических данных

Атрибут `text:bibliography-data-field` определяет, какая часть поля библиографических данных будет отображена.

```
<define name="text-index-entry-bibliography-attrs" combine="interleave">
    <attribute name="text:bibliography-data-field">
        <choice>
            <value>address</value>
            <value>annote</value>
            <value>author</value>
            <value>bibliography-type</value>
            <value>booktitle</value>
            <value>chapter</value>
            <value>custom1</value>
            <value>custom2</value>
            <value>custom3</value>
            <value>custom4</value>
            <value>custom5</value>
            <value>edition</value>
            <value>editor</value>
            <value>howpublished</value>
            <value>identifier</value>
            <value>institution</value>
            <value>isbn</value>
            <value>issn</value>
            <value>journal</value>
            <value>month</value>
            <value>note</value>
            <value>number</value>
            <value>organizations</value>
            <value>pages</value>
            <value>publisher</value>
            <value>report-type</value>
```

```
            <value>school</value>
            <value>series</value>
            <value>title</value>
            <value>url</value>
            <value>volume</value>
            <value>year</value>
        </choice>
    </attribute>
</define>
```

7.12.6 Табулостоп

Элемент `<text:index-entry-tab-stop>` представляет табулостоп в индексной записи. Данный элемент также содержит информацию о позиции табулостопа.

```
<define name="text-index-entry-tab-stop">
    <element name="text:index-entry-tab-stop">
        <optional>
            <attribute name="text:style-name">
                <ref name="styleNameRef"/>
            </attribute>
        </optional>
        <ref name="text-index-entry-tab-stop-attrs"/>
    </element>
</define>
```

С элементом `<text:index-entry-tab-stop>` могут быть связаны следующие атрибуты:

- `style:leader-char`
- `style:type`
- `style:position`

Первый символ

Атрибут `style:leader-char` определяет первый символ.

```
<define name="text-index-entry-tab-stop-attrs" combine="interleave">
    <optional>
        <attribute name="style:leader-char">
            <ref name="character"/>
        </attribute>
    </optional>
</define>
```

Тип и позиция табулостопа

Атрибут `style:type` определяет тип табулостопа. Элемент `<text:index-entry-tab-stop>` поддерживает только два типа табулостопа: `left` и `right`.

Если значение этого атрибута равно `left`, то также должен быть использован атрибут `style:position`. В противном случае этот атрибут не должен указываться. Атрибут `style:position` определяет позицию табулостопа. В зависимости от значения атрибута `text:relative-tab-stop-position` элемента `<text:index-entry-config>`, позиция табулятора определяется относительно левого поля или левого отступа.

```
<define name="text-index-entry-tab-stop-attrs" combine="interleave">
    <choice>
        <attribute name="style:type">
            <value>right</value>
        </attribute>
        <group>
            <attribute name="style:type">
                <value>left</value>
            </attribute>
            <attribute name="style:position">
                <ref name="length"/>
            </attribute>
        </group>
    </choice>
</define>
```

7.12.7 Начало и конец гиперссылки

Элементы `<text:index-entry-link-start>` и `<text:index-entry-link-end>` помечают начало и конец гиперссылки индексной записи. Стиль символов для гиперссылки может быть включен в элемент индексной записи как атрибут `text:style-name`.

```
<define name="text-index-entry-link-start">
    <element name="text:index-entry-link-start">
        <optional>
            <attribute name="text:style-name">
                <ref name="styleNameRef"/>
            </attribute>
        </optional>
    </element>
</define>
<define name="text-index-entry-link-end">
    <element name="text:index-entry-link-end">
        <optional>
            <attribute name="text:style-name">
                <ref name="styleNameRef"/>
            </attribute>
        </optional>
    </element>
</define>
```

7.12.8 Пример конфигурации индексной записи

Рассмотрим пример кода XML для оглавления с именем «Table of Content» со следующими особенностями:

- используются два верхних уровня структуры;
- каждая запись состоит из номера главы, закрывающей скобки, названия главы, табулостопа и номера страницы;
- для уровня структуры, номера страницы используется стиль с именем bold;
- для второго уровня структуры вместо закрывающей скобки используется квадратная скобка.

```
Example: Table of Content
<text:table-of-content>
    <text:table-of-content-source
        text:outline-level="2"
        text:use-index-marks="false"
        text:index-scope="document">

        <text:index-title-template text:style-name="Index 1">
            Table of Content
        </text:index-title-template>

        <text:index-entry-template
            text:outline-level="1"
            text:style-name="Contents 1">
            <text:index-entry-chapter text:display="number"/>
            <text:index-entry-span>) </text:index-entry-span>
            <text:index-entry-text/>
            <text:index-entry-tab-stop style:type="right"/>
            <text:index-entry-page-number text:style-name="bold"/>
        </text:index-entry-template>

        <text:index-entry-template
            text:outline-level="2"
            text:style-name="Contents 2">
            <text:index-entry-chapter text:display="number"/>
            <text:index-entry-span>] </text:index-entry-span>
            <text:index-entry-text/>
            <text:index-entry-tab-stop style:type="right"/>
            <text:index-entry-page-number/>
        </text:index-entry-template>

    </text:table-of-content-source>

    <text:table-of-content-body>
        [... header ...]
        <text:p text:style-name="[...]">1) Chapter
            <text:tab-stop/><text:span stylename="bold"> 1 </text:span>
        </text:p>
        <text:p text:style-name="[...]">1.1] Subchapter
            <text:tab-stop/>1
        </text:p>
        [... more entries ...]
    </text:table-of-content-body>
```

```
</text:table-of-content>
```

8 Таблицы

В данном разделе описывается структура таблиц, используемая для таблиц, вложенных в текстовые документы, и электронных таблиц.

8.1 Общая модель таблиц

Структура таблиц OpenDocument подобна структуре таблиц [HTML4] или [XSL], и, как и они, таблицы могут вкладываться.

Представление таблиц основано на сетке строк и столбцов. Строки имеют больший приоритет, чем столбцы. Таблица делится на строки, строки делятся на ячейки. Каждый столбец включает описание столбца, но это описание не содержит ячеек.

Строки таблиц могут быть пустыми, различные строки могут содержать разное количество ячеек. Это не является ошибкой, но разные приложения могут трактовать это неодинаково. Приложения электронных таблиц обычно работают с большими таблицами, которые имеют фиксированные, зависящие от приложения, номера строк и столбцов, но могут содержать неиспользуемые области. В файле сохраняется только используемая область таблицы. Когда в приложении электронных таблиц загружается таблица с пустыми или неполными строками, пустые строки обычно представляются строками по умолчанию (как пустая таблица), а неполные строки заполняются пустыми ячейками (подобно пустой таблице). Все другие приложения обычно имеют фиксированный размер таблицы. Неполные строки обычно обрабатываются как имеющие достаточное количество пустых ячеек. Так же обрабатываются пустые строки. Пустые ячейки обычно занимают пространство пустого абзаца.

Строки и столбцы фигурируют в **группах строк** и **группах столбцов**. Эти группы определяют, должны ли повторяться строки или столбцы на следующей странице.

8.1.1 Элемент таблицы

Элемент таблицы является корнем для таблиц.

```
<define name="table-table">
    <element name="table:table">
        <ref name="table-table-attlist"/>
        <optional>
            <ref name="table-table-source"/>
        </optional>
        <optional>
            <ref name="office-dde-source"/>
        </optional>
        <optional>
            <ref name="table-scenario"/>
        </optional>
        <optional>
            <ref name="office-forms"/>
        </optional>
        <optional>
            <ref name="table-shapes"/>
        </optional>
        <ref name="table-columns-and-groups"/>
        <ref name="table-rows-and-groups"/>
    </element>
</define>
```

Модель содержимого таблиц весьма сложна. Детали будут рассмотрены в разделе 8.2. Сейчас можно допустить, что элемент таблицы состоит из элементов строки и столбца.

```
<define name="table-columns-and-groups">
    <oneOrMore>
        <choice>
            <ref name="table-table-column-group"/>
            <ref name="table-columns-no-group"/>
        </choice>
    </oneOrMore>
</define>

<define name="table-columns-no-group">
    <choice>
        <group>
            <ref name="table-columns"/>
            <optional>
                <ref name="table-table-header-columns"/>
                <optional>
                    <ref name="table-columns"/>
                </optional>
            </optional>
        </group>
        <group>
            <ref name="table-table-header-columns"/>
            <optional>
                <ref name="table-columns"/>
            </optional>
        </group>
    </choice>
</define>

<define name="table-columns">
    <choice>
        <ref name="table-table-columns"/>
        <oneOrMore>
                <ref name="table-table-column"/>
```

```
3518        </oneOrMore>
3519    </choice>
3520 </define>
3521
3522 <define name="table-rows-and-groups">
3523    <oneOrMore>
3524        <choice>
3525            <ref name="table-table-row-group"/>
3526            <ref name="table-rows-no-group"/>
3527        </choice>
3528    </oneOrMore>
3529 </define>
3530
3531 <define name="table-rows-no-group">
3532    <choice>
3533        <group>
3534            <ref name="table-rows"/>
3535            <optional>
3536                <ref name="table-table-header-rows"/>
3537                <optional>
3538                    <ref name="table-rows"/>
3539                </optional>
3540            </optional>
3541        </group>
3542        <group>
3543            <ref name="table-table-header-rows"/>
3544            <optional>
3545                <ref name="table-rows"/>
3546            </optional>
3547        </group>
3548    </choice>
3549 </define>
3550
3551 <define name="table-rows">
3552    <choice>
3553        <ref name="table-table-rows"/>
3554        <oneOrMore>
3555                <ref name="table-table-row"/>
3556        </oneOrMore>
3557    </choice>
3558 </define>
```

Имя таблицы

Атрибут `table:name` определяет имя таблицы.

```
3559 <define name="table-table-attlist" combine="interleave">
3560    <optional>
3561        <attribute name="table:name">
3562            <ref name="string"/>
3563        </attribute>
3564    </optional>
3565 </define>
```

Стиль таблицы

Атрибут `table:style-name` ссылается на стиль таблицы, т. е. на элемент `<style:style>` типа таблица. Стиль таблицы описывает свойства

форматирования таблицы, такие как ширина и цвет фона. Стиль таблицы может быть автоматическим или общим стилем.

```
<define name="table-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Пример – Стиль таблицы:

```
<style:style style:name="Table 1" style:family="table">
    <style:table-properties style:width="12cm"
     fo:background-color="light-grey"/>
</style:style>

<table:table table:name="Table 1" table:style-name="Table 1">
    ...
</table:table>
```

Защищенность

Атрибут `table:protected` определяет, защищена ли таблица от редактирования. Если таблица защищена, атрибут `table:protection-key` может определять пароль с целью предотвращения возможности сброса флага защищенности пользователем для разрешения редактирования. Если таблица защищена, все ее элементы и элементы ячеек с атрибутом `style:cell-protect`, со значением `true`, являются защищенными.

Во избежание сохранения пароля напрямую в XML-файл, в атрибуте `table:protection-key` сохраняется только значение хеш-функции пароля.

```
<define name="table-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:protected" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="table:protection-key">
            <text/>
        </attribute>
    </optional>
</define>
```

Печать

Атрибут `table:print` определяет, является ли таблица печатаемой. Он принимает логическое значение. Если оно равно `true`, таблица печатаемая, если

false, таблица непечатаемая. Значение по умолчанию равно true. Атрибут table:print переопределяется атрибутом table:display, описанным в разделе 15.8.14. Т. е. если таблица неотображаемая, то она непечатаемая.

Если таблица печатаемая, фактически печатаемый диапазон таблицы может быть определен атрибутом table:print-ranges (см. следующий раздел). Если этот атрибут не существует, будет распечатана используемая область таблицы.

```
<define name="table-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:print" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Диапазон печати

Атрибут table:print-ranges определяет области печати таблицы, т. е. ячейки, которые должны быть напечатаны. Он содержит список адресов ячеек или диапазон адресов ячеек, как описано в разделе 8.3.1.

```
<define name="table-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:print-ranges">
            <ref name="cellRangeAddressList"/>
        </attribute>
    </optional>
</define>
```

8.1.2 Строка таблицы

Элемент <table:table-row> представляет строку таблицы. Он содержит элементы, определяющие ячейки данной строки таблицы.

Элемент <table:table-row> подобен элементу [XSL] <fo:table-row>.

```
<define name="table-table-row">
    <element name="table:table-row">
        <ref name="table-table-row-attlist"/>
        <oneOrMore>
            <choice>
                <ref name="table-table-cell"/>
                <ref name="table-covered-table-cell"/>
            </choice>
        </oneOrMore>
    </element>
</define>
```

Количество повторяемых строк

Атрибут table:number-rows-repeated определяет количество строк, к которым применяется данный элемент строки. Если две или более строки являются соседними, имеют одинаковое содержимое и свойства и не содержат вертикально объединенных ячеек, то они могут быть описаны одним элементом <table:table-row>, который имеет атрибут table:number-rows-repeated со значением большим 1.

```
<define name="table-table-row-attlist" combine="interleave">
    <optional>
        <attribute name="table:number-rows-repeated" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Стиль строки

Стиль строки таблицы хранит свойства форматирования строки таблицы, такие как высота и цвет фона. Стиль строки определяется элементом <style:style> со значением семейства атрибутов table-row. Стиль строки таблицы может быть автоматическим либо общим стилем. Он связыается атрибутом строки таблицы table:style-name.

```
<define name="table-table-row-attlist" combine="interleave">
    <optional>
        <attribute name="table:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Стиль ячейки по умолчанию

Атрибут table:default-cell-style-name определяет стиль ячейки по умолчанию. Ячейки, содержащиеся в строке, не имеющей индивидуального стиля ячейки, используют этот стиль по умолчанию.

```
<define name="table-table-row-attlist" combine="interleave">
    <optional>
        <attribute name="table:default-cell-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Видимость

Атрибут `table:visibility` определяет, является ли строка видимой, отфильтрованной или свернутой. Отфильтрованные и свернутые строки невидимы. Отфильтрованные строки невидимы, так как фильтр применяется к таблице, которая не выбирает свою строку. Свернутые строки становятся невидимыми при помощи пользовательского интерфейса.

```
<define name="table-table-row-attlist" combine="interleave">
    <optional>
        <attribute name="table:visibility" a:defaultValue="visible">
            <ref name="table-visibility-value"/>
        </attribute>
    </optional>
</define>

<define name="table-visibility-value">
    <choice>
        <value>visible</value>
        <value>collapse</value>
        <value>filter</value>
    </choice>
</define>
```

Пример – Таблица с тремя строками и тремя столбцами.

В данном примере приведен код OpenDocument для таблицы с тремя строками и тремя столбцами. Цвет фона первых двух строк таблицы — синий.

```
<style:style style:name="Table 1" style:family="table">
    <style:table-properties style:width="12cm"
        fo:background-color="light-grey"/>
</style:style>
<style:style style:name="Col1" style:family="table-column">
    <style:table-column-properties style:column-width="2cm"/>
</style:style>
<style:style style:name="Col2" style:family="table-column">
    <style:table-column-properties style:column-width="4cm"/>
</style:style>
<style:style style:name="Col3" style:family="table-column">
    <style:table-column-properties style:column-width="6cm"/>
</style:style>
<style:style style:name="Row1" style:family="table-row">
    <style:table-row-properties fo:background-color="blue"/>
</style:style>

<table:table table:name="Table 1" table:style-name="Table 1">
    <table:table-columns>
        <table:table-column table:style-name="Col1"/>
        <table:table-column table:style-name="Col2"/>
        <table:table-column table:style-name="Col3"/>
    </table:table-columns>
    <table:table-rows>
        <table:table-row table:style-name="Row1">
            ...
        </table:table-row>
        <table:table-row table:style-name="Row1">
            ...
```

```
        </table:table-row>
        <table:table-row>
            ...
        </table:table-row>
    <table:table-rows>
</table:table>
```

8.1.3 Ячейки таблицы

Элементы `<table:table-cell>` и `<table:covered-table-cell>` определяют содержимое ячеек таблицы. Они содержатся в элементах строк таблицы. Ячейки таблицы могут содержать абзацы и другое текстовое содержимое, включая подтаблицы. Ячейки таблицы могут быть пустыми.

Элемент `<table:table-cell>` очень похож на элемент ячейки таблицы [XSL] и [HTML4]. Существующие HTML и XSL правила, относящиеся к ячейкам, охватывающим несколько столбцов или строк, также применимы и к спецификации OpenDocument. Это означает, что элементы `<table:table-cell>` отсутствуют в строках/столбцах сетки для позиций, которые перекрыты объединенными ячейками, охватывающими несколько столбцов или строк. С целью определения ячеек для таких позиций используется элемент `<table:covered-table-cell>`. Он должен появляться в каждой позиции строки/столбца сетки, перекрытой ячейкой, охватывающей несколько строк или столбцов. Эта позиция в сетке вычисляется в предположении, что диапазон объединения столбца и строки равен 1 для всех ячеек, независимо от того, определены они элементом `<table:table-cell>` или `<table:covered-table-cell>`. Элемент `<table:covered-table-cell>` обычно используется в приложениях электронных таблиц, для которых характерно наличие содержимого в объединенной ячейке.

```
<define name="table-table-cell">
    <element name="table:table-cell">
        <ref name="table-table-cell-attlist"/>
        <ref name="table-table-cell-attlist-extra"/>
        <ref name="table-table-cell-content"/>
    </element>
</define>

<define name="table-covered-table-cell">
    <element name="table:covered-table-cell">
        <ref name="table-table-cell-attlist"/>
        <ref name="table-table-cell-content"/>
    </element>
</define>

<define name="table-table-cell-content">
    <optional>
        <ref name="table-cell-range-source"/>
```

```
    </optional>
    <optional>
        <ref name="office-annotation"/>
    </optional>
    <optional>
        <ref name="table-detective"/>
    </optional>
    <zeroOrMore>
        <ref name="text-content"/>
    </zeroOrMore>
</define>
```

Количество повторяемых ячеек

Атрибут `table:number-columns-repeated` определяет количество последовательных столбцов, в которых повторяются ячейки. Он может быть использован для описания двух и более соседних ячеек в одном элементе ячейки, если выполняются следующие условия:

- ячейки имеют одинаковые содержимое и свойства;
- ячейки не объединены по горизонтали или по вертикали.

В этом случае для задания количества последовательных столбцов, в которых будут повторяться ячейки, должен использоваться атрибут `table:number-columns-repeated`. Этот атрибут определяестя с одним из двух элементов: `<table:table-cell>` или `<table:covered-table-cell>`.

```
<define name="table-table-cell-attlist" combine="interleave">
    <optional>
        <attribute name="table:number-columns-repeated" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Количество объединенных строк и столбцов

Эти атрибуты определяют количество объединенных ячейкой строк и столбцов. Атрибуты могут использоваться только с элементом `<table:table-cell>`.

В том случае, когда ячейка перекрывает другие ячейки, так как значение объединения столбцов или строк более одного, для представления объединенной ячейки должен использоваться элемент `<table:covered-table-cell>`.

```
<define name="table-table-cell-attlist-extra" combine="interleave">
    <optional>
        <attribute name="table:number-columns-spanned" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
```

```
    </optional>
    <optional>
        <attribute name="table:number-rows-spanned" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Стиль ячейки

Стиль ячейки таблицы хранит свойства форматирования ячейки, такие как:

- цвет фона;
- формат числа;
- вертикальное выравнивание;
- обрамление.

Стиль ячейки таблицы может быть автоматическим или общим стилем. Стиль определяется атрибутом `table:style-name`. Если ячейка не имеет назначенного стиля ячейки, приложение проверяет, определен ли в текущей строке стиль ячейки по умолчанию. Если текущая строка не имеет стиля ячейки по умолчанию, приложение проверяет, определен ли стиль ячейки по умолчанию в текущем столбце.

```
<define name="table-table-cell-attlist" combine="interleave">
    <optional>
        <attribute name="table:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Проверка содержимого ячейки

Атрибут `table:content-validation-name` указывается, если в ячейке осуществляется проверка правильности содержимого. Значение этого атрибута является именем элемента `<table:cell-content-validation>`. Если атрибут не представлен, ячейка может иметь произвольное содержимое.

```
<define name="table-table-cell-attlist" combine="interleave">
    <optional>
        <attribute name="table:content-validation-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Более детальные сведения о проверке содержимого ячейки и об элементе `<table:cell-content-validation>` приведены в разделе 8.5.3.

Формулы

Формулы позволяют производить вычисления в ячейках таблицы. Каждая формула начинается с префикса пространства имен, задающего синтаксис и семантику, используемые в формуле. Обычно формула начинается со знака равенства (=) и может включать следующие компоненты:

- числа;
- текст;
- именованные диапазоны;
- знаки арифметических операций;
- логические операторы;
- вызовы функций;
- адреса ячеек, содержащих числа — адрес может быть относительным или абсолютным (см. раздел 8.3.1), адреса в формулах начинаются с символа «[» и заканчиваются символом «]» (сведения об адресации ячеек или диапазонов ячеек приведены в разделе 8.3.1).

Рассмотрим пример простой формулы:

```
=sum([.A1:.A5])
```

Формула вычисляет сумму значений во всех ячейках в диапазоне «`.A1:.A5`». Используется функция «`sum`». Параметры заключены в круглые скобки «`(`» в начале и «`)`» в конце. Если функция содержит более одного параметра, параметры разделяются символом «`;`».

Формулу, приведенную выше, можно представить в виде:

```
=sum([.A1];[.A2];[.A3];[.A4];[.A5])
```

Результат будет тем же. Компоненты, используемые в формуле, зависят от используемого приложения.

Атрибут `table:formula` содержит формулу ячейки таблицы.

```
<define name="table-table-cell-attlist" combine="interleave">
    <optional>
        <attribute name="table:formula">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

В дополнение, вычисленное значение формулы доступно обычным способом. Текущее значение ячейки представляется одним из следующих атрибутов:

- `office:value`
- `office:date-value`
- `office:time-value`
- `office:boolean-value`
- `office:string-value`

Матрицы

В том случае, когда приложение выполняет вычисления электронной таблицы, связный диапазон ячеек, содержащих значения, называется матрицей. Если диапазон ячеек содержит *m* строк и *n* столбцов, матрица называется матрицей размера *m x n*. Минимально возможные матрицы состоят из двух соседних ячеек и имеют размеры *1 x 2* или *2 x 1*. Для использования матрицы в формуле в нее включается диапазон адресов ячеек матрицы. В матричной формуле допустимо использование только специальных операций с матрицами.

Количество строк и столбцов, объединенных в матрицу, представляется атрибутами `table:number-matrix-rows-spanned` и `table:number-matrix-columns-spanned`, привязанными к элементам ячеек.

```
<define name="table-table-cell-attlist-extra" combine="interleave">
    <optional>
        <attribute name="table:number-matrix-columns-spanned">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="table:number-matrix-rows-spanned">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Тип значения

Атрибут `table:value-type` определяет тип значения, размещенного в ячейке. Он может принимать одно из следующих значений:

- `float`, `percentage` или `currency` — вещественные, процентный или денежный числовые типы;
- `date` — дата;
- `time` — время;
- `boolean` — логический;
- `string` — текстовый.

```
<define name="table-table-cell-attlist" combine="interleave">
    <optional.
        <ref name="common-value-and-type-attlist"/>
    </optional>
</define>
```

Текущее числовое значение ячейки

Атрибут `office:value` определяет текущее числовое значение ячейки. Этот атрибут вычисляется только для ячеек, содержащих следующие типы данных:

- `float` — вещественный;
- `percentage` — процентный;
- `currency` — денежный.

Текущая денежная единица ячейки

Атрибут `tableoffice:currency` определяет текущую денежную единицу ячейки. Значение этого атрибута — обычно информация о валюте, например, DEM или EUR. Данный атрибут определяется для ячеек, имеющих тип данных `currency`.

Текущее значение даты ячейки

Атрибут `office:date-value` определяет текущее значение даты ячейки. Данный атрибут определяется для ячеек, имеющих тип данных `date`.

Некоторые приложения поддерживают значения даты и времени в дополнение к датам.

Текущее значение времени ячейки

Атрибут `office:time-value` определяет текущее значение времени ячейки. Данный атрибут определяется для ячеек, имеющих тип данных `time`.

Текущее логическое значение ячейки

Атрибут office:boolean-value определяет текущее логическое значение ячейки. Данный атрибут определяется для ячеек, имеющих тип данных boolean.

Текущее текстовое значение ячейки

Атрибут office:string-value определяет текущее текстовое значение в ячейке. Данный атрибут определяется для ячеек, имеющих тип данных string.

Защита ячеек таблицы

Атрибут table:protected обеспечивают защиту ячеек таблицы. Пользователи не могут редактировать содержимое ячейки, помеченной как защищенная.

```
3732 <define name="table-table-cell-attlist" combine="interleave">
3733     <optional>
3734         <attribute name="table:protect" a:defaultValue="false">
3735             <ref name="boolean"/>
3736         </attribute>
3737     </optional>
3738 </define>
```

Данный атрибут не связан с атрибутом table:protected табличного элемента (см. раздел 8.1.1) и атрибутом стиля ячейки таблицы table:cell-protect (см. раздел 15.11.14).

8.2 Расширенная модель таблиц

8.2.1 Описание столбца

Каждый столбец таблицы имеет элемент описания столбца таблицы <table:table-column>. Он подобен [XSL] элементу <fo:table-column> и, используется, главным образом, для указания стиля столбца, который определяет, например, ширину столбца.

```
3739 <define name="table-table-column">
3740     <element name="table:table-column">
3741         <ref name="table-table-column-attlist"/>
3742         <empty/>
3743     </element>
3744 </define>
```

Количество повторяемых столбцов

Атрибут table:number-columns-repeated определяет количество столбцов, в которых повторяется описание столбца. Если два или более соседних столбца имеют одинаковые свойства, этот атрибут позволяет описать их одним элементом <table:table-column>.

```
<define name="table-table-column-attlist" combine="interleave">
    <optional>
        <attribute name="table:number-columns-repeated" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Стиль столбца

Стиль столбца таблицы хранит информацию о свойствах форматирования столбца, таких как ширина и цвет фона. Он определяется элементом <style:style> со значением семейства атрибутов table-column и может быть автоматическим или общим стилем. Стиль столбца определяется с использованием атрибута table:style-name.

```
<define name="table-table-column-attlist" combine="interleave">
    <optional>
        <attribute name="table:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Видимость

Атрибут table:visibility определяет, является ли столбец видимым, отфильтрованным или свернутым. Более детальные сведения приведены в разделе 8.1.2.

```
<define name="table-table-column-attlist" combine="interleave">
    <optional>
        <attribute name="table:visibility" a:defaultValue="visible">
            <ref name="table-visibility-value"/>
        </attribute>
    </optional>
</define>
```

Стиль ячейки по умолчанию

Атрибут `table:default-cell-style-name` определяет стиль ячейки по умолчанию. Ячейки, стиль которых не определен, используют данный стиль в том случае, если не задан стиль по умолчанию ячеек строки.

```
<define name="table-table-column-attlist" combine="interleave">
    <optional>
        <attribute name="table:default-cell-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Пример – Таблица с тремя столбцами.

В примере приведен код OpenDocument для таблицы с тремя столбцами:

```
<style:style style:name="Table 1" style:family="table">
    <style:table-properties style:width="12cm"
        fo:background-color="light-grey"/>
</style:style>
<style:style style:name="Col1" style:family="table-column">
    <style:table-column-properties style:column-width="2cm"/>
</style:style>
<style:style style:name="Col2" style:family="table-column">
    <style:table-column-properties style:column-width="4cm"/>
</style:style>
<style:style style:name="Col3" style:family="table-column">
    <style:table-column-properties style:column-width="6cm"/>
</style:style>

<table:table table:name="Table 1" table:style-name="Table 1">
    <table:table-columns>
        <table:table-column table:style-name="Col1"/>
        <table:table-column table:style-name="Col2"/>
        <table:table-column table:style-name="Col3"/>
    </table:table-columns>
    ...
</table:table>
```

8.2.2 Заголовки столбцов

Если таблица не может быть размещена на одной странице, то несколько соседних столбцов могут автоматически повторяться на каждой странице. Для этого описания столбцов должны быть включены в элемент `<table:table-header-columns>`. Описания столбцов, которые не должны повторяться на каждой странице, могут быть включены в элемент `<table:table-columns>`, но это не обязательно. Таблица не должна содержать более одного элемента `<table:table-header-columns>`, и за элементом `<table:table-columns>` не должен следовать другой элемент `<table:table-columns>`. Единственным исключением являются таблицы, содержащие сгруппированные столбцы (см.

раздел 8.2.3). Такие таблицы содержат более одного элемента `<table:table-header-columns>`, при этом они должны входить в различные группы столбцов, состоящих из соседних элементов.

Приложения, которые не поддерживают заголовки столбцов, должны обрабатывать их описания так же, как и описания столбцов без заголовков.

Элементы `<table:table-header-columns>` и `<table:table-columns>` очень похожи на элементы [HTML4] `<THEAD>` и `<TBODY>` для строк.

```
<define name="table-table-header-columns">
    <element name="table:table-header-columns">
        <oneOrMore>
            <ref name="table-table-column"/>
        </oneOrMore>
    </element>
</define>

<define name="table-table-columns">
    <element name="table:table-columns">
        <oneOrMore>
            <ref name="table-table-column"/>
        </oneOrMore>
    </element>
</define>
```

8.2.3 Группы столбцов

Соседние столбцы таблицы могут быть сгруппированы с использованием элемента `<table:table-column-group>`. Каждая группа может содержать новую группу, столбцы и заголовки столбцов. Группа столбцов может быть видимой и скрытой. Группы столбцов могут, например, использоваться приложениями электронных таблиц для группировки столбцов, данные которых суммируются. Таким образом, отдельные столбцы, вносящие вклад в сумму, можно сделать невидимыми, в то время как сумма остается видимой.

Если набор заголовков столбцов и группа столбцов перекрываются, то заголовок группы столбцов разрывает набор заголовков столбцов. Т.е. `<table:table-column-group>` может содержать элементы `<table:table-header-columns>`, но не наоборот.

```
<define name="table-table-column-group">
    <element name="table:table-column-group">
        <ref name="table-table-column-group-attlist"/>
        <ref name="table-columns-and-groups"/>
    </element>
</define>
```

Видимость

Атрибут table:display определяет, является ли группа видимой или невидимой.

```
<define name="table-table-column-group-attlist" combine="interleave">
    <optional>
        <attribute name="table:display" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.2.4 Заголовки строк

Если таблица не может быть размещена на одной странице, то несколько соседних строк могут автоматически повторяться на каждой странице. Для этого описания строк должны быть включены в элемент <table:table-header-rows>. Описания столбцов, которые не должны повторяться на каждой странице, могут быть включены в элемент <table:table-rows>, но не обязательно. Таблица может содержать более одного элемента <table:table-header-rows>, и за <table:table-rows> не должен следовать другой элемент <table:table-rows>. Единственным исключением являются таблицы, содержащие сгруппированные строки (см. раздел 8.2.5). Такие таблицы содержат более одного элемента <table:table-header-rows>, при этом обеспечивается их вхождение в различные группы строк, состоящих из соседних элементов.

Приложения, которые не поддерживают заголовки строк, должны обрабатывать их описания так же, как и описания строк без заголовков.

Элементы <table:table-header-rows> и <table:table-rows> очень похожи на элементы [HTML4] <THEAD> и <TBODY>.

```
<define name="table-table-header-rows">
    <element name="table:table-header-rows">
        <oneOrMore>
            <ref name="table-table-row"/>
        </oneOrMore>
    </element>
</define>

<define name="table-table-rows">
    <element name="table:table-rows">
        <oneOrMore>
            <ref name="table-table-row"/>
        </oneOrMore>
    </element>
</define>
```

8.2.5 Группы строк

Соседние строки таблицы могут быть сгруппированы с использованием элемента `<table:table-row-group>`. Каждая группа может содержать новую группу, строки и заголовки строк. Группа строк может быть видимой и скрытой. Группы строк могут, например, использоваться приложениями электронных таблиц для группировки строк, данные которых суммируются. Таким образом, отдельные строки, вносящие вклад в сумму, можно сделать невидимыми, в то время как сумма остается видимой.

Если набор заголовков строк и группа строк перекрываются, то заголовок группы строк разрывает набор заголовков строк. Т. е. `<table:table-row-group>` может содержать элементы `<table:table-header-rows>`, но не наоборот.

```
<define name="table-table-row-group">
    <element name="table:table-row-group">
        <ref name="table-table-row-group-attlist"/>
        <ref name="table-rows-and-groups"/>
    </element>
</define>
```

Видимость

Атрибут `table:display` определяет, является ли группа видимой или невидимой.

```
<define name="table-table-row-group-attlist" combine="interleave">
    <optional>
        <attribute name="table:display" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.2.6 Подтаблицы

Если ячейка таблицы содержит одну таблицу и не содержит абзацев и других элементов, то такая таблица определяется как подтаблица. Она занимает всю ячейку, и в этой ячейке не может быть другого содержимого.

Обрамление подтаблицы объединяется с обрамлением ячейки, в которой она расположена. Подтаблица не содержит никаких свойств форматирования. Подтаблица — это, по существу, контейнер для некоторого количества дополнительных строк , которые неразрывно связаны с родительской таблицей.

Вложенная таблица преобразуется в подтаблицу атрибутом `table:is-subtable`, который включается в элемент таблицы. Вложенная таблица, не определенная как подтаблица, рассматривается как таблица в таблице, т. е. она имеет собственное обрамление и подчиняется правилам выравнивания родительской ячейки.

```
<define name="table-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:is-sub-table" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Пример — представление подтаблицы.

В схеме OpenDocument такая таблица может быть представлена двумя способами, представленными в *Примере 1* и *Примере 2*.

A1	***B1***	***C1***
A2	B2.1.1	B2.2.1
	B2.1.2	

Пример 1

При использовании ячеек, объединяющих несколько строк, таблица определяется следующим образом:

```
<style:style style:name="Table 1" style:family="table">
    <style:table-properties style:width="12cm"
                    fo:background-color="light-grey"/>
</style:style>
<style:style style:name="Col1" style:family="table-column">
    <style:table-column-properties style:column-width="2cm"/>
</style:style>
<style:style style:name="Col2" style:family="table-column">
    <style:table-column-properties style:column-width="4cm"/>
</style:style>
<style:style style:name="Col3" style:family="table-column">
    <style:table-column-properties style:column-width="6cm"/>
</style:style>
<style:style style:name="Row1" style:family="table-row">
    <style:table-row-properties fo:background-color="grey"/>
</style:style>
<style:style style:name="Cell1" style:family="table-cell">
    <style:table-cell-properties fo:background-color="grey"/>
</style:style>
<table:table table:name="Table 1" table:style-name="Table 1">
    <table:table-columns>
        <table:table-column table:style-name="Col1"/>
        <table:table-column table:style-name="Col2"/>
        <table:table-column table:style-name="Col3"/>
    </table:table-columns>
    <table:table-header-rows>
        <table:table-row table:style-name="Row1">
            <table:table-cell>
```

```
                <text:p text:style="Table Caption">
                    A1
                </text:p>
            </table:table-cell>
            <table:table-cell>
                <text:p text:style="Table Caption">
                    B1
                </text:p>
            </table:table-cell>
            <table:table-cell>
                <text:p text:style="Table Caption">
                    C1
                </text:p>
            </table:table-cell>
        </table:table-row>
    </table:table-header-rows>
    <table:table-rows>
        <table:table-row>
            <table:table-cell table:number-rows-spanned="2"
                table:style-name="Cell1">
                <text:p text:style="Table Body">
                    A2
                </text:p>
            </table:table-cell>
            <table:table-cell>
                <text:p text:style="Table Body">
                    B2.1.1
                </text:p>
            </table:table-cell>
            <table:table-cell>
                <text:p text:style="Table Body">
                    B2.2.1
                </text:p>
            </table:table-cell>
        </table:table-row>
        <table:table-row>
            <table:covered-table-cell/>
            <table:table-cell table:number-columns-spanned="2">
                <text:p text:style="Table Body">
                B2.1.2
            </text:p>
            </table:table-cell>
            <table:covered-table-cell/>
        </table:table-row>
    </table:table-rows>
</table:table>
```

Пример 2

При использованием подтаблиц таблица определяется следующим образом:

```
<style:style style:name="Table 1" style:family="table">
    <style:table-properties fo:width="12cm" fo:background-color="light-
grey"/>
</style:style>
<style:style style:name="Col1" style:family="table-column">
    <style:table-column-properties style:column-width="2cm"/>
</style:style>
<style:style style:name="Col2" style:family="table-column">
    <style:table-column-properties style:column-width="4cm"/>
</style:style>
<style:style style:name="Col3" style:family="table-column">
    <style:table-column-properties style:column-width="6cm"/>
</style:style>
```

```
<style:style style:name="Row1" style:family="table-row">
    <style:table-row-properties fo:background-color="grey"/>
</style:style>
<style:style style:name="Cell1" style:family="table-cell">
    <style:table-cell-properties fo:background-color="grey"/>
</style:style>

<table:table table:name="Table 1" table:style-name="Table 1">
    <table:table-columns>
        <table:table-column table:style-name="Col1"/>
        <table:table-column table:style-name="Col2"/>
        <table:table-column table:style-name="Col3"/>
    </table:table-columns>
    <table:table-header-rows>
        <table:table-row table:style-name="Row1">
            <table:table-cell>
                <text:p text:style="Table Caption">
                    A1
                </text:p>
            </table:table.cell>
            <table:table-cell>
                <text:p text:style="Table Caption">
                    B1
                </text:p>
            </table:table-cell>
            <table:table-cell>
                <text:p text:style="Table Caption">
                    C1
                </text:p>
            </table:table-cell>
        </table:table-row>
    </table:table-header-rows>
    <table:table-rows>
        <table:table-row>
            <table:table-cell table:style-name="Cell1">
                <text:p text:style="Table Body">
                    A2
                </text:p>
            </table:table-cell>
            <table:table-cell table:number-columns-spanned="2">
                <table:table is-subtable="true">
                    <table:table-columns>
                        <table:table-column table:style-name="Col2"/>
                        <table:table-column table:style-name="Col3"/>
                    </table:table-columns>
                    <table:rows>
                        <table:row>
                            <table:table-cell>
                                <text:p text:style="Table Body">
                                    B2.1.1
                                </text:p>
                            </table:table-cell>
                            <table:table-cell>
                                <text:p text:style="Table Body">
                                    B2.2.1
                                </text:p>
                            </table:table-cell>
                        </table:table-row>
                        <table:table-row>
                            <table:table-cell
                                table:number-columns-spanned="2">
                                <text:p text:style="Table Body">
                                    B2.1.2
                                </text:p>
```

```
                                </table:table-cell>
                                <table:covered-table-cell/>
                            </table:table-row>
                        </table:table-rows>
                    </table:table>
                </table:table-cell>
                <table:covered-table-cell/>
            </table:table-row>
        </table:table-rows>
</table:table>
```

8.3 Расширенные таблицы

8.3.1 Ссылки на ячейки таблицы

Для ссылки на ячейки таблицы используются так называемые адреса ячеек. Адрес ячейки имеет следующую структуру:

1. Имя таблицы.
2. Точка (.).
3. Буквенное значение, идентифицирующее столбец. Буква A обозначает столбец 1, B — столбец 2, и так далее. AA представляет столбец 27, AB — столбец 28 и так далее.
4. Числовое значение, идентифицирующее строку. Число 1 представляет первую строку, число 2 — вторую и так далее.

Это означает, что A1 представляет ячейку на пересечении столбца 1 и строки 1. B1 представляет ячейку на пересечении столбца 2 и строки 1. A2 представляет ячейку на пересечении столбца 1 и строки 2.

Например, в таблице с именем `SampleTable` ссылка на ячейку на пересечении столбца 34 и строки 16 представляется адресом `SampleTable.AH16`. В некоторых случаях имя таблицы указывать не обязательно. Однако, точка должна быть представлена. Если имя таблицы не требуется, то адрес из предыдущего примера будет иметь вид `.AH16`.

Адрес ячейки подтаблицы имеет следующую структуру:

1. Адрес ячейки, содержащей подтаблицу.
2. Точка (.).
3. Адрес ячейки в подтаблице.

Например, ссылка на ячейку на пересечении столбца 1 и строки 1 подтаблицы с именем `Subtable`, расположенной в ячейке на пересечении столбца 34 и строки 16 таблицы `SampleTable`, будет представлена адресом

`SampleTable.AH16.A1`. Если имя таблицы содержит пробелы, оно должно быть заключено в апострофы (').

Абсолютная и относительная адресация

Ссылки на ячейки могут быть заданы абсолютными либо относительными адресами. При выполнении операции с ячейкой таблицы, например копировании формулы, абсолютная ссылка на ячейку не меняется. В отличие от этого, относительные ссылки адаптируются к адресу ячейки, в которую осуществляется копирование. В предыдущем примере использована относительная адресация.

Для создания абсолютного адреса необходимо установить знак доллара ($) перед именем таблицы, идентификатором столбца и идентификатором строки. Например, абсолютный адрес для предыдущего примера имеет вид `$SampleTable.$AH$16`. Абсолютные и относительные ссылки могут комбинироваться в одном адресе ячейки. Например, `SampleTable.AH$16`, ссылается относительно на таблицу и столбец, но абсолютно на строку. Абсолютные адреса должны обязательно содержать имя таблицы. Учет различий между абсолютными и относительными адресами необходим только в некоторых ситуациях. Если необходимости нет, могут использоваться ссылки без знака доллара.

```
<define name="cellAddress">
    <data type="string">
        <param name="pattern">($?([^\. ']+|'[^']+'))?\.$?[A-Z]+$?[0-9]+</param>
    </data>
</define>
```

Адрес диапазона ячеек

Диапазон ячеек — это набор соседних ячеек, образующих прямоугольник. Прямоугольник простирается от верхней левой ячейки до нижней правой.

Адрес диапазона ячеек ссылается на диапазон ячеек. Он имеет следующую структуру:

1. Адрес верхней левой ячейки диапазона.
2. Двоеточие (:).
3. Адрес нижней правой ячейки диапазона.

Например, адрес `.A1:.B2` является ссылкой на диапазон ячеек от находящейся на пересечении столбца 1 и строки 1 до находящейся на

пересечении столбца 2 и строки 2. Минимальный диапазон определяется как одна ячейка. В этом случае адрес диапазона совпадает с адресом ячейки.

```
<define name="cellRangeAddress">
    <data type="string">
        <param name="pattern">($?([^\. ']+|'[^']+'))?\.$?[A-Z]+$?[0-9]+(:($?
([^\. ']+|'[^']+'))?\.$?[A-Z]+$?[0-9]+)?</param>
    </data>
</define>
```

Список адресов диапазонов ячеек

Список адресов диапазонов ячеек — это список диапазонов ячеек и адресов ячеек. Элементы списка отделяются пробелом. Если имя таблицы содержит пробелы, оно должно быть заключено в апострофы (').

```
<define name="cellRangeAddressList">
    <!-- Value is a space separated list of "cellRangeAddress" patterns -->
    <data type="string"/>
</define>
```

8.3.2 Связанные таблицы

Если таблица связана с первичной таблицей, информация о таблице-источнике содержится в элементе `<table:table-source>`. С элементом `<table:table-source>` могут быть связаны следующие атрибуты::

- режим;
- имя таблицы;
- URL ;
- имя фильтра;
- параметры фильтра;
- интервал обновления.

```
<define name="table-table-source">
    <element name="table:table-source">
        <ref name="table-table-source-attlist"/>
        <ref name="table-linked-source-attlist"/>
        <empty/>
    </element>
</define>
```

Режим

Атрибут `table:mode` определяет, какие данные должны быть скопированы из исходной таблицы в целевую таблицу. Если атрибут имеет значение, равное

copy-all, то будут скопированы формулы и стили. Если атрибут имеет значение copy-results-only (копировать только результаты), то будут скопированы только результаты вычислений и невычисляемое содержимое ячеек.

```
<define name="table-table-source-attlist" combine="interleave">
    <optional>
        <attribute name="table:mode" a:defaultValue="copy-all">
            <choice>
                <value>copy-all</value>
                <value>copy-results-only</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Имя таблицы

Атрибут table:table-name определяет имя таблицы первичного документа. Если имя таблицы не определено, то будет использована первая таблица документа.

```
<define name="table-table-source-attlist" combine="interleave">
    <optional>
        <attribute name="table:table-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

URL

Первичная таблица определяется посредством XLink, где атрибут xlink:href определяет URL документа, содержащего первичную таблицу.

```
<define name="table-linked-source-attlist" combine="interleave">
    <optional>
        <attribute name="xlink:type" a:defaultValue="simple">
            <value>simple</value>
        </attribute>
    </optional>
    <optional>
        <attribute name="xlink:actuate" a:defaultValue="onRequest">
            <value>onRequest</value>
        </attribute>
    </optional>
    <attribute name="xlink:href">
        <ref name="anyURI"/>
    </attribute>
</define>
```

Имя фильтра

Атрибут `table:filter-name` определяет тип файла документа, содержащего первичную таблицу. Значение этого атрибута зависит от конкретного приложения.

```
<define name="table-linked-source-attlist" combine="interleave">
    <optional>
        <attribute name="table:filter-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Параметры фильтра

Атрибут `table:filter-options` определяет необязательные настройки типа файла. Значение этого атрибута зависит от конкретного приложения.

```
<define name="table-linked-source-attlist" combine="interleave">
    <optional>
        <attribute name="table:filter-options">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Интервал обновления

Атрибут `table:refresh-delay` определяет интервал времени между обновлениями связанной таблицы.

```
<define name="table-linked-source-attlist" combine="interleave">
    <optional>
        <attribute name="table:refresh-delay">
            <ref name="duration"/>
        </attribute>
    </optional>
</define>
```

8.3.3 Таблицы сценариев

Сценарий — это область таблицы, данные в которой временно связаны с другой таблицей, называемой таблицей сценария. Если несколько сценариев определены для одной и той же области, пользователь может переключаться от одного к другому. Будет ли таблица сценария видимой, определяется стилем таблицы. Только одна таблица сценария может быть активной в отдельной таблице.

Таблица, содержащая `<table:scenario>`, представляет собой таблицу сценария. Имя таблицы и имя сценария совпадают. Сценарий отображается в обычной таблице, предшествующей таблице сценария. Если таблица сценария для таблицы существует, сценарий отображается в этой таблице автоматически. Это означает, что существование таблицы сценария предполагает существование сценария.

С данным элементом могут быть связаны следующие атрибуты:

- диапазоны сценария;
- активность;
- отображение обрамления;
- цвет обрамления;
- копирование обратно;
- копирование стилей;
- копирование формул;
- комментарий;
- защищенность.

```
<define name="table-scenario">
    <element name="table:scenario">
        <ref name="table-scenario-attlist"/>
        <empty/>
    </element>
</define>
```

Диапазоны сценария

Атрибут `table:scenario-ranges` определяет диапазон ячеек таблицы, который отображается как сценарий. Значением этого атрибута является список адресов диапазонов ячеек.

```
<define name="table-scenario-attlist" combine="interleave">
    <attribute name="table:scenario-ranges">
        <ref name="cellRangeAddressList"/>
    </attribute>
</define>
```

Активность

Атрибут `table:is-active` определяет, является ли активным сценарий, принадлежащий таблице сценария.

```
<define name="table-scenario-attlist" combine="interleave">
    <attribute name="table:is-active">
```

```
        <ref name="boolean"/>
    </attribute>
</define>
```

Отображение обрамления

Атрибут `table:display-border` определяет, отображается ли обрамление вокруг сценария, принадлежащего таблице сценария.

```
<define name="table-scenario-attlist" combine="interleave">
    <optional>
        <attribute name="table:display-border" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Цвет обрамления

Атрибут `table:border-color` определяет цвет обрамления, отображаемого вокруг сценария, принадлежащего таблице сценария.

```
<define name="table-scenario-attlist" combine="interleave">
    <optional>
        <attribute name="table:border-color">
            <ref name="color"/>
        </attribute>
    </optional>
</define>
```

Копирование обратно

Атрибут `table:copy-back` определяет, копируются ли данные обратно в таблицу сценария, если активизируется другой сценарий.

```
<define name="table-scenario-attlist" combine="interleave">
    <optional>
        <attribute name="table:copy-back" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Копирование стилей

Атрибут `table:copy-styles` определяет, копируются ли стили из таблицы сценария в целевую таблицу вместе с данными.

```
<define name="table-scenario-attlist" combine="interleave">
    <optional>
        <attribute name="table:copy-styles" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
```

```
    </optional>
</define>
```

Копирование формул

Атрибут `table:copy-formulas` определяет, копируются ли формулы из таблицы сценария в целевую таблицу. Значение атрибута может принимать значение `true` или `false`. Если значение равно `true`, то формулы копируются. Если значение равно `false`, то будут копироваться только результаты формул.

```
<define name="table-scenario-attlist" combine="interleave">
    <optional>
        <attribute name="table:copy-formulas" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Комментарий

Атрибут `table:comment` содержит комментарий о сценарии.

```
<define name="table-scenario-attlist" combine="interleave">
    <optional>
        <attribute name="table:comment">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Защищенность

Атрибут `table:protected` определяет, защищены ли от редактирования, данные, отображаемые в сценарии. Значение атрибута анализируется только в том случае, если таблица, в которой отображается сценарий также является защищенной (см. раздел 8.1.1).

```
<define name="table-scenario-attlist" combine="interleave">
    <optional>
        <attribute name="table:protected">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.3.4 Фигуры

Элемент `<table:shapes>` содержит все графические фигуры с привязкой к таблице, потомком которой является элемент. Этот элемент является контейнером и не имеет сопутствующих атрибутов.

```
<define name="table-shapes">
    <element name="table:shapes">
        <oneOrMore>
            <ref name="shape"/>
        </oneOrMore>
    </element>
</define>
```

8.4 Расширенные ячейки таблицы

8.4.1 Связанные ячейки таблицы

Диапазон ячеек может быть связан с диапазоном базы данных или именованным диапазоном другого файла. В этом случае информация о первичном диапазоне базы данных или именованном диапазоне содержится в элементе `<table:cell-range-source>`, расположенном в первой ячейке диапазона. С данным элементом могут быть связаны следующие атрибуты:

- имя;
- последний размер;
- URL;
- имя фильтра;
- параметры фильтра;
- интервал обновления.

```
<define name="table-cell-range-source">
    <element name="table:cell-range-source">
        <ref name="table-table-cell-range-source-attlist"/>
        <ref name="table-linked-source-attlist"/>
        <empty/>
    </element>
</define>
```

Имя

Атрибут `table:name` определяет имя исходного диапазона базы данных или именованного диапазона.

```
<define name="table-table-cell-range-source-attlist" combine="interleave">
    <attribute name="table:name">
        <ref name="string"/>
```

```
    </attribute>
</define>
```

Последний размер

Атрибуты `table:last-column-spanned` и `table:last-row-spanned` определяют последний известный размер диапазона. Если размер диапазона изменяется с момента последней операции, значение данного атрибута ошибочно.

```
<define name="table-table-cell-range-source-attlist" combine="interleave">
    <attribute name="table:last-column-spanned">
        <ref name="positiveInteger"/>
    </attribute>
    <attribute name="table:last-row-spanned">
        <ref name="positiveInteger"/>
    </attribute>
</define>
```

URL, имя фильтра, параметры фильтра и интервал обновления

Атрибуты `xlink:href`, `xlink:type`, `xlink:actuate`, `table:filter-name` и `table:filter-options` подобны атрибутам связанных таблиц. Детальные сведения приведены в разделе 8.3.2.

8.4.2 Аннотация ячейки

Формат OpenDocument позволяет размещать аннотацию внутри ячеек таблицы. Для получения детальной информации об аннотациях см. раздел 12.1.

8.4.3 Слежение

Элемент `<table:detective>` имеет двойное назначение. С одной стороны, он содержит информацию о связях, которые существуют между ячейками таблицы как следствие применения формул и которые должны выделяться в пользовательском интерфейсе цветом. С другой стороны, элемент содержит информацию о ячейках, которые на данный момент выделены цветом в пользовательском интерфейсе либо вследствие упомянутых выше связей, либо вследствие ошибочных условий.

```
<define name="table-detective">
    <element name="table:detective">
        <zeroOrMore>
            <ref name="table-highlighted-range"/>
        </zeroOrMore>
```

```
        <zeroOrMore>
            <ref name="table-operation"/>
        </zeroOrMore>
    </element>
</define>
```

В элементе <table:detective> могут содержаться следующие элементы:

- операция слежения;
- диапазон выделения цветом.

8.4.4 Операция слежения

Элемент <table:operation> определяет, какие конкретные связи, существующие между ячейкой, потомком которой является элемент, и другими ячейками, должны быть сделаны видимыми или невидимыми в пользовательском интерфейсе. Одна и та же операция слежения может быть повторена несколько раз по отношению к одной и той же ячейке. В этом случае вторая операция применяется к результирующей ячейке первой операции и так далее. Это означает, что операция не обязательно применяется только к ячейке, в которой она определена, но и к другим ячейкам, благодаря чему может взаимодействовать с операциями, определенными в других ячейках. Главным образом, это применяется в операциях, которые делают связи невидимыми. Для обеспечения детерминированного поведения, операции имеют индексы и применяются в порядке, определенном ими. С элементом <table:operation> могут быть связаны следующие атрибуты:

- имя;
- индекс.

```
<define name="table-operation">
    <element name="table:operation">
        <ref name="table-operation-attlist"/>
        <empty/>
    </element>
</define>
```

Имя

Атрибут table:name определяет имя операции слежения. Возможны следующие имена: trace-dependents, remove-dependents, trace-precedents, remove-precedents и trace-errors. Trace-dependents и remove-dependents отображают или скрывают ячейки, которые используют

значение текущей ячейки в их формулах. `Trace-precedents` и `remove-precedents` отображают или скрывают ячейки, чьи значения используются в формуле текущей ячейки. `Trace-errors` отображает ячейки, которые привели к ошибке при вычислении результата формулы текущей ячейки.

```
<define name="table-operation-attlist" combine="interleave">
    <attribute name="table:name">
        <choice>
            <value>trace-dependents</value>
            <value>remove-dependents</value>
            <value>trace-precedents</value>
            <value>remove-precedents</value>
            <value>trace-errors</value>
        </choice>
    </attribute>
</define>
```

Индекс

Атрибут `table:index` определяет порядок, в котором применяются операции слежения.

```
<define name="table-operation-attlist" combine="interleave">
    <attribute name="table:index">
        <ref name="nonNegativeInteger"/>
    </attribute>
</define>
```

8.4.5 Диапазон выделения цветом

Элемент `<table:highlighted-range>` определяет диапазон ячеек, который в пользовательском интерфейсе выделяется цветом вследствие ошибки или некорректных данных.

Информация, содержащаяся в этом элементе, может быть устаревшей, но отражает состояние, вычисленное на данный момент операциями слежения или условиями ошибки.

С элементом `<table:highlighted-range>` могут быть связаны следующие атрибуты:

- адрес диапазона ячеек;
- направление;
- наличие ошибки;
- метка некорректности.

```
<define name="table-highlighted-range">
    <element name="table:highlighted-range">
        <choice>
```

```
            <group>
                <ref name="table-highlighted-range-attlist"/>
            </group>
            <group>
                <ref name="table-highlighted-range-attlist-invalid"/>
            </group>
        </choice>
        <empty/>
    </element>
</define>
```

Адрес диапазона ячеек

Атрибут `table:cell-range-address` содержит адрес диапазона, который в данный момент выделяется цветом.

```
<define name="table-highlighted-range-attlist" combine="interleave">
    <optional>
        <attribute name="table:cell-range-address">
            <ref name="cellRangeAddress"/>
        </attribute>
    </optional>
</define>
```

Направление

Атрибут `table:direction` определяет направление связи между данной ячейкой и диапазоном, выделенным цветом. Например, направление может быть представлено стрелкой.

```
<define name="table-highlighted-range-attlist" combine="interleave">
    <attribute name="table:direction">
        <choice>
            <value>from-another-table</value>
            <value>to-another-table</value>
            <value>from-same-table</value>
        </choice>
    </attribute>
</define>
```

Наличие ошибки

Атрибут `table:contains-error` определяет, содержит ли диапазон ячеек ошибку.

```
<define name="table-highlighted-range-attlist" combine="interleave">
    <optional>
        <attribute name="table:contains-error" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Метка некорректности

Атрибут `table:marked-invalid` определяет, помечена ли текущая ячейка как некорректная. Этот атрибут не может быть использован с другими атрибутами.

```
<define name="table-highlighted-range-attlist-invalid" combine="interleave">
    <attribute name="table:marked-invalid">
        <ref name="boolean"/>
    </attribute>
</define>
```

8.5 Содержимое документов электронных таблиц

8.5.1 Защита документа

Структура документа электронной таблицы может быть защищена с использованием атрибута `table:structure-protected`, чтобы пользователи не имели возможности вставлять, удалять или переименовывать таблицы в документе. Необязательный атрибут `table:protection-key` может быть использован для задания пароля, предотвращающего сброс пользователями флага защиты для разрешения редактирования. Во избежание сохранения пароля напрямую в XML-файле, в атрибуте `table:protection-key` сохраняется только значение хеш-функции пароля.

```
<define name="office-spreadsheet-attlist" combine="interleave">
    <optional>
        <attribute name="table:structure-protected" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="table:protection-key">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

8.5.2 Настройки вычислений

Документы электронных таблиц содержат настройки, которые влияют на вычисления формул, например настройки нулевой даты и итерации. Эти настройки могут быть сохранены в документе с помощью элемента `<table:calculation-settings>`.

```
<define name="table-calculation-settings">
    <element name="table:calculation-settings">
        <ref name="table-calculation-setting-attlist"/>
```

```
        <optional>
            <ref name="table-null-date"/>
        </optional>
        <optional>
            <ref name="table-iteration"/>
        </optional>
    </element>
</define>
```

С элементом `<table:calculation-settings>` могут быть связаны следующие атрибуты:

- чувствительность к регистру;
- точность как на экране;
- распространенность условий поиска на всю ячейку;
- автоматический поиск надписей;
- использование регулярных выражений;
- нулевой год;
- нулевая дата;
- итерации.

Чувствительность к регистру

Атрибут `table:case-sensitive` определяет, учитывается ли различие между верхним и нижним регистрами при применении к содержимому ячейки операторов сравнения.

```
<define name="table-calculation-setting-attlist" combine="interleave">
    <optional>
        <attribute name="table:case-sensitive" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Точность как на экране

Атрибут `table:precision-as-shown` определяет, выполняются ли вычисления с округленными числами электронной таблицы, отображаемыми на экране, или с использованием всех цифр числа. Если значение атрибута равно `true`, вычисления выполняются с округленными числами электронной таблицы, отображаемыми на экране. Если значение атрибута равно `false`, вычисления выполняются с использованием всех цифр числа, но результаты также отображаются в округленном виде.

```
<define name="table-calculation-setting-attlist" combine="interleave">
    <optional>
        <attribute name="table:precision-as-shown" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Распространенность условий поиска на всю ячейку

Атрибут `table:search-criteria-must-apply-to-whole-cell` определяет, должно ли условие поиска, соответствующее используемому регулярному выражению, применяться ко всему содержимому ячейки.

```
<define name="table-calculation-setting-attlist" combine="interleave">
    <optional>
        <attribute name="table:search-criteria-must-apply-to-whole-cell"
                    a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Автоматический поиск надписей

Атрибут `table:automatic-find-labels` определяет, будет ли осуществляться автоматический поиск надписей строк и столбцов.

```
<define name="table-calculation-setting-attlist" combine="interleave">
    <optional>
        <attribute name="table:automatic-find-labels" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Использование регулярных выражений

Атрибут `table:use-regular-expressions` определяет, разрешено ли использование регулярных выражений при выполнении операций сравнения строк и поиске.

```
<define name="table-calculation-setting-attlist" combine="interleave">
    <optional>
        <attribute name="table:use-regular-expressions"
                    a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Нулевой год

Атрибут table:null-year определяет начальный год для значений годов, содержащих только две цифры. Две цифры значения года интерпретируются как год равный или следующий за начальным годом.

```
<define name="table-calculation-setting-attlist" combine="interleave">
    <optional>
        <attribute name="table:null-year" a:defaultValue="1930">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Нулевая дата

Элемент <table:null-date> определяет нулевую дату. Нулевая дата — это дата, которая при преобразовании ее в числовой формат дает значение 0. Нулевая дата определяется в атрибуте table:date-value элемента. Обычно используются значения 12/30/1899, 01/01/1900 и 01/01/1904.

```
<define name="table-null-date">
    <element name="table:null-date">
        <optional>
            <attribute name="table:value-type" a:defaultValue="date">
                <ref name="valueType"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="table:date-value-type"
                        a:defaultValue="1899-12-30">
                <ref name="date"/>
            </attribute>
        </optional>
        <empty/>
    </element>
</define>
```

Итерации

Элемент <table:iteration> разрешает вычисление формул с итеративными (или циклическими) ссылками после заданного количества итераций. Вычисление формул с итеративными ссылками повторяется до тех пор, пока задача не будет решена. Если итеративные вычисления не разрешены, появление в таблице формулы с итеративной ссылкой приведет к появлению сообщения об ошибке.

Итеративные вычисления разрешаются и запрещаются атрибутом table:status. Если итеративные вычисления разрешены, в атрибуте

table:steps определяется максимально допустимое количество итераций. Атрибут table:maximum-difference определяет максимально допустимую разность между двумя результатами вычислений. Итерация останавливается, если результат меньше значения данного атрибута.

```
<define name="table-iteration">
    <element name="table:iteration">
        <optional>
            <attribute name="table:status" a:defaultValue="disable">
                <choice>
                    <value>enable</value>
                    <value>disable</value>
                </choice>
            </attribute>
        </optional>
        <optional>
            <attribute name="table:steps" a:defaultValue="100">
                <ref name="positiveInteger"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="table:maximum-difference"
                        a:defaultValue="0.001">
                <ref name="double"/>
            </attribute>
        </optional>
        <empty/>
    </element>
</define>
```

8.5.3 Проверки содержимого ячеек таблицы

Данный элемент определяет правила проверки правильности содержимого ячеек таблицы. Элемент <table:content-validation> определяет такое правило. Все правила проверки, имеющиеся в документе, содержатся в элементе <table:content-validations>. Сами правила проверки именуются, и эти имена используются для ссылок на них из ячейки таблицы.

```
<define name="table-content-validations">
    <element name="table:content-validations">
        <oneOrMore>
            <ref name="table-content-validation"/>
        </oneOrMore>
    </element>
</define>

<define name="table-content-validation">
    <element name="table:content-validation">
        <ref name="table-validation-attlist"/>
        <optional>
            <ref name="table-help-message"/>
        </optional>
        <optional>
            <choice>
                <ref name="table-error-message"/>
```

```
                <group>
                    <ref name="table-error-macro"/>
                    <optional>
                        <ref name="office-event-listeners"/>
                    </optional>
                </group>
            </choice>
        </optional>
    </element>
</define>
```

С элементом `<table:content-validation>` могут быть связаны следующие атрибуты:

- имя;
- условие;
- адрес базовой ячейки;
- допустимость пустых ячеек;
- отображение списка.

Имя

Атрибут `table:name` определяет имя проверки содержимого. Он используется в ячейке, к которой должно быть применено правило, для ссылки на правило проверки. Имя создается приложением автоматически.

```
<define name="table-validation-attlist" combine="interleave">
    <attribute name="table:name">
        <ref name="string"/>
    </attribute>
</define>
```

Условие

Атрибут `table:condition` определяет условие, которое должно принимать значение `true` для всех ячеек, к которым применено правило проверки. Значение атрибута должно быть префиксом пространства имен, за которым следует логическое выражение.

Типовой синтаксис выражения может быть подобен синтаксису XPath. Ниже приведены существующие условия.

- `Condition ::= ExtendedTrueCondition | TrueFunction 'and' TrueCondition`

- TrueFunction ::= cell-content-is-whole-number() | cell-content-is-decimal-number() | cell-content-is-date() | cell-content-is-time() | cell-content-is-text()
- ExtendedTrueCondition ::= ExtendedGetFunction | cell-content-text-length() Operator Value
- TrueCondition ::= GetFunction | cell-content() Operator Value
- GetFunction ::= cell-content-is-between(Value, Value) | cell-content-is-not-between(Value, Value)
- ExtendedGetFunction ::= cell-content-text-length-is-between(Value, Value) | cell-content-text-length-is-not-between(Value, Value) | cell-content-is-in-list(StringList)
- Operator ::= '<' | '>' | '<=' | '>=' | '=' | '!='
- Value ::= NumberValue | String | Formula
- StringList ::= StringList ';' String | String
- Formula — это формула без знака равно (=) в начале. Более подробные сведения приведены в разделе 8.1.3.
- String — включает один или более символов, заключенных в кавычки.
- NumberValue — целое или вещественное число. Оно не должно содержать десятичный разделитель, если число больше или равно 1000.

```
<define name="table-validation-attlist" combine="interleave">
    <optional>
        <attribute name="table:condition">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Адрес базовой ячейки

Атрибут table:base-cell-address определяет адрес базовой ячейки для относительных адресов в формулах, появляющихся в условиях. Этот атрибут обязателен только в том случае, если условие содержит формулу. Значением данного атрибута должен быть абсолютный адрес ячейки, включающий имя таблицы.

```
<define name="table-validation-attlist" combine="interleave">
    <optional>
        <attribute name="table:base-cell-address">
            <ref name="cellAddress"/>
        </attribute>
    </optional>
</define>
```

Допустимость пустых ячеек

Атрибут `table:allow-empty-cell` определяет, может ли ячейка быть пустой.

```
<define name="table-validation-attlist" combine="interleave">
    <optional>
        <attribute name="table:allow-empty-cell" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Отображение списка

Атрибут `table:display-list` определяет, отображается ли список значений, который появляется в условии, в пользовательском интерфейсе в процессе редактирования значения ячейки. Этот атрибут может принимать значения `none`, `unsorted` или `sort-ascending`.

- `none` — список значений не отображается;
- `unsorted` — список значений отображается в порядке их появления в условии;
- `sort-ascending` — список значений отображается в порядке возрастания.

```
<define name="table-validation-attlist" combine="interleave">
    <optional>
        <attribute name="table:display-list" a:defaultValue="unsorted">
            <choice>
                <value>none</value>
                <value>unsorted</value>
                <value>sort-ascending</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Справочные сообщения

Элемент `<table:help-message>` определяет сообщения, отображаемые при выборе пользователем ячейки. Элемент имеет необязательный атрибут

table:title, который определяет заголовок справочного сообщения. Также он имеет необязательный атрибут table:display, который может быть использован для запрета отображения справочного сообщения.

```
<define name="table-help-message">
    <element name="table:help-message">
        <optional>
            <attribute name="table:title">
                <ref name="string"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="table:display" a:defaultValue="false">
                <ref name="boolean"/>
            </attribute>
        </optional>
        <zeroOrMore>
            <ref name="text-p"/>
        </zeroOrMore>
    </element>
</define>
```

Сообщения об ошибке

Элемент <table:error-message> определяет сообщение, отображаемое при попытке пользователя ввести недопустимые данные в ячейку, т. е. когда условие правила проверки принимает значение false. Элемент имеет необязательный атрибут table:title, который определяет заголовок сообщения. Также он имеет атрибут table:display , который может быть использован для запрета отображения сообщения. Атрибут table:message-type, который может принимать значения stop, warning, или information, определяет должно ли сообщение отображаться как ошибка (stop), предупреждение (warning) или только информация (information). В случае, если отображается сообщение об ошибке, операция, которая вызывала проверку правильности, останавливается (например, перемещение курсора из ячейки).

```
<define name="table-error-message">
    <element name="table:error-message">
        <optional>
            <attribute name="table:title">
                <ref name="string"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="table:display" a:defaultValue="false">
                <ref name="boolean"/>
            </attribute>
        </optional>
        <optional>
            <attribute name="table:message-type" a:defaultValue="stop">
```

```
                <choice>
                    <value>stop</value>
                    <value>warning</value>
                    <value>information</value>
                </choice>
            </attribute>
        </optional>
        <zeroOrMore>
            <ref name="text-p"/>
        </zeroOrMore>
    </element>
</define>
```

Макрос ошибки

Если ячейка содержит недопустимое содержимое, вместо вывода сообщения может вызываться макрос. В данном случае макрос определяется элементом `<office:event-listeners>`, как описано в разделе 12.4. Имя, определяющее событие, которое вызывается в случае недопустимого ввода пользователя, должно быть единственным.

В дополнение к элементу `<office:event-listeners>`, элемент `<table:error-macro>` определяет, должен ли выполняться макрос.

```
<define name="table-error-macro">
    <element name="table:error-macro">
        <optional>
            <attribute name="table:execute" a:defaultValue="true">
                <ref name="boolean"/>
            </attribute>
        </optional>
    </element>
</define>
```

8.5.4 Диапазоны надписей

Диапазоны надписей могут использоваться для назначения имен строк и столбцов, или частей строк и столбцов, где имена ячеек рассматриваются как их же содержимое. Точнее, элемент диапазона надписей `<table:label-range>` определяет диапазон ячеек надписей, содержащих надписи, и диапазон ячеек данных, определяющий строки и столбцы, на содержимое которых ссылаются надписи.

Существует два типа диапазонов надписей:

- диапазоны надписей для столбцов;
- диапазоны надписей для строк.

Диапазон ячеек данных должен иметь те же высоту и позицию по вертикали, что и диапазон ячеек надписей, если это надписи строк, или те же ширину и позицию по горизонтали, если это надписи столбцов. Сведения об определении диапазона ячеек приведены в разделе 8.3.1.

Надписи могут быть использованы в формуле так же, как и другие имена. Все диапазоны надписей, существующие в документе, содержатся в одном элементе `<table:label-ranges>`.

```
<define name="table-label-ranges">
    <element name="table:label-ranges">
        <zeroOrMore>
            <ref name="table-label-range"/>
        </zeroOrMore>
    </element>
</define>

<define name="table-label-range">
    <element name="table:label-range">
        <ref name="table-label-range-attlist"/>
        <empty/>
    </element>
</define>
```

Адрес диапазона ячеек надписей

Атрибут `table:label-cell-range-address` определяет адрес диапазона ячеек надписей.

```
<define name="table-label-range-attlist" combine="interleave">
    <attribute name="table:label-cell-range-address">
        <ref name="cellRangeAddress"/>
    </attribute>
</define>
```

Адрес диапазона ячеек данных

Атрибут `table:data-cell-range-address` определяет адрес диапазона ячеек данных.

```
<define name="table-label-range-attlist" combine="interleave">
    <attribute name="table:data-cell-range-address">
        <ref name="cellRangeAddress"/>
    </attribute>
</define>
```

Ориентация

Атрибут `table:orientation` определяет ориентацию диапазона надписей. Этот атрибут может иметь значения `column` (столбец) или `row` (строка).

```
<define name="table-label-range-attlist" combine="interleave">
    <attribute name="table:orientation">
        <choice>
            <value>column</value>
            <value>row</value>
        </choice>
    </attribute>
</define>
```

8.5.5 Именованные выражения

Элемент именованных выражений `<table:named-expressions>` содержит совокупность присваиваний имен выражениям, чтобы имена могли быть использованы для ссылок на выражения.

Имена могут иметь следующие выражения:

- диапазон ячеек;
- другие выражения, например части формулы.

```
<define name="table-named-expressions">
    <element name="table:named-expressions">
        <zeroOrMore>
            <choice>
                <ref name="table-named-range"/>
                <ref name="table-named-expression"/>
            </choice>
        </zeroOrMore>
    </element>
</define>

```

Именованный диапазон

Элемент именованный диапазон `<table:named-range>` определяет диапазон ячеек, которому назначено имя. Сведения об определении диапазона ячеек приведены в разделе 8.3.1.

Атрибут `table:name` определяет имя диапазона, а атрибут `table:cell-range-address` — его адрес. Адрес может быть абсолютным или относительным. Если адрес диапазона относительный, дополнительно должен существовать атрибут `table:base-cell-address`. Он определяет адрес базовой ячейки диапазона ячеек. Этот адрес должен быть абсолютным. Поэтому в этом адресе должно быть обязательно указано имя таблицы, но символы доллара, которые служат признаком абсолютного адреса, могут быть опущены.

Дополнительный атрибут `table:range-usable-as` определяет, может ли имя диапазона использоваться в определении диапазона печати, фильтра,

повторяющейся строки или повторяющегося столбца. Этот атрибут может принимать значения:

- none (нет);
- список, разделенный пробелами, содержащий значения print-range (диапазон печати), filter (фильтр), repeat-row (повторяющаяся строка) или repeat-column (повторяющийся столбец).

```
<define name="table-named-range">
    <element name="table:named-range">
        <ref name="table-named-range-attlist"/>
        <empty/>
    </element>
</define>

<define name="table-named-range-attlist" combine="interleave">
    <attribute name="table:name">
        <ref name="string"/>
    </attribute>
    <attribute name="table:cell-range-address">
        <ref name="cellRangeAddress"/>
    </attribute>
    <optional>
        <attribute name="table:base-cell-address">
            <ref name="cellAddress"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="table:range-usable-as" a:defaultValue="none">
            <choice>
                <value>none</value>
                <list>
                    <oneOrMore>
                        <choice>
                            <value>print-range</value>
                            <value>filter</value>
                            <value>repeat-row</value>
                            <value>repeat-column</value>
                        </choice>
                    </oneOrMore>
                </list>
            </choice>
        </attribute>
    </optional>
</define>
```

Именованное выражение

Элемент именованного выражения <table:named-expression> содержит выражения с именами, например части формулы.

Атрибут table:name определяет имя выражения, а атрибут table:expression — само выражение. Выражение не должно содержать знак равенства (=) в качестве первого символа. Если выражение содержит

именованный диапазон или другое именованное выражение, то именованный диапазон или другое именованное выражение должны быть определены первыми, до включения в выражение. Если выражение содержит относительный адрес диапазона ячеек, дополнительно должен существовать атрибут `table:base-cell-address`. Он определяет адрес базовой ячейки для диапазона ячеек. Этот адрес должен быть абсолютным. Поэтому в этом адресе должно быть обязательно указано имя таблицы, но символы доллара, которые служат признаком абсолютного адреса, могут быть опущены.

```
<define name="table-named-expression">
    <element name="table:named-expression">
        <ref name="table-named-expression-attlist"/>
        <empty/>
    </element>
</define>

<define name="table-named-expression-attlist" combine="interleave">
    <attribute name="table:name">
        <ref name="string"/>
    </attribute>
    <attribute name="table:expression">
        <ref name="string"/>
    </attribute>
    <optional>
        <attribute name="table:base-cell-address">
            <ref name="cellAddress"/>
        </attribute>
    </optional>
</define>
```

Пример – Элемент именованных выражений с именованным диапазоном и именованным выражением:

```
<table:named-expressions>
    <table:named-range table:name="sample1" table:cell-range-address=".C4"
        table:base-cell-address="sampletable.F1" table:area-type="none"/>
    <table:named-range table:name="sample2"
        table:cell-range-address=".$D$3:.$K$8"
        table:area-type="print-range filter"/>
    <table:named-expression table:name="sample3"
        table:expression="sum([.A1:.B3])"/>
</table:named-expressions>
```

8.6 Диапазоны баз данных

Диапазон баз данных — это именованная область в таблице, в которой могут выполняться операции базы данных, но также выполняются и операции другого вида, такие как фильтрация и сортировка. Элемент `<table:database-ranges>` содержит набор всех диапазонов баз данных, определенных в документе.

```
<define name="table-database-ranges">
```

```
    <element name="table:database-ranges">
        <zeroOrMore>
            <ref name="table-database-range"/>
        </zeroOrMore>
    </element>
</define>
```

8.6.1 Диапазон базы данных

Элемент `<table:database-range>` определяет один диапазон базы данных.

```
<define name="table-database-range">
    <element name="table:database-range">
        <ref name="table-database-range-attlist"/>
        <optional>
            <choice>
                <ref name="table-database-source-sql"/>
                <ref name="table-database-source-table"/>
                <ref name="table-database-source-query"/>
            </choice>
        </optional>
        <optional>
            <ref name="table-filter"/>
        </optional>
        <optional>
            <ref name="table-sort"/>
        </optional>
        <optional>
            <ref name="table-subtotal-rules"/>
        </optional>
    </element>
</define>
```

Имя диапазона базы данных

Атрибут `table:name` определяет имя диапазона базы данных, на котором выполняются операции. В одном документе только один диапазон базы данных может не иметь имени. Этот диапазон базы данных обычно создается приложением автоматически и используется для фильтрации или сортировки данных в диапазонах ячеек без явного создания пользователем диапазона базы данных.

```
<define name="table-database-range-attlist" combine="interleave">
    <optional>
        <attribute name="table:name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Выборка

Атрибут `table:is-selection` определяет, включает ли диапазон полную базу данных или только выборку записей из базы данных.

```
<define name="table-database-range-attlist" combine="interleave">
    <optional>
        <attribute name="table:is-selection" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Сохранение стиля при обновлении

Атрибут `table:on-update-keep-styles` определяет поведение диапазона базы данных при обновлении. Если значение атрибута `true`, то стили, назначенные ячейкам первой строки диапазона базы данных, не являющейся надписью, используются для всех строк в диапазоне базы данных. Если значение атрибут `false`, все ячейки диапазона базы данных используют назначенный в документе стиль ячеек по умолчанию.

```
<define name="table-database-range-attlist" combine="interleave">
    <optional>
        <attribute name="table:on-update-keep-styles" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Сохранение размера при обновлении

Атрибут `table:on-update-keep-size` определяет поведение диапазона базы данных при изменении размера данных в источнике данных. Если значение атрибута `true`, диапазон сохраняет свой размер. Если значение атрибута `false`, диапазон не изменяет свой размер.

```
<define name="table-database-range-attlist" combine="interleave">
    <optional>
        <attribute name="table:on-update-keep-size" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Постоянство данных

Атрибут `table:has-persistent-data` определяет, будут ли сохранены текущие данные диапазона базы данных при сохранении документа.

```
4455 <define name="table-database-range-attlist" combine="interleave">
4456     <optional>
4457         <attribute name="table:has-persistent-data" a:defaultValue="true">
4458             <ref name="boolean"/>
4459         </attribute>
4460     </optional>
4461 </define>
```

Ориентация

Атрибут `table:orientation` определяет ориентацию диапазона базы данных. Возможные значения этого атрибута `row` и `column`. Ориентация, например, используется при сортировке диапазона базы данных (см. раздел 8.6.5). Если ориентация имеет значение `row`, осуществляется сортировка строк, иначе — столбцов.

```
4462 <define name="table-database-range-attlist" combine="interleave">
4463     <optional>
4464         <attribute name="table:orientation" a:defaultValue="row">
4465             <choice>
4466                 <value>column</value>
4467                 <value>row</value>
4468             </choice>
4469         </attribute>
4470     </optional>
4471 </define>
```

Содержание заголовка

Атрибут `table:contains-header` определяет, должны ли первая строка или столбец диапазона базы данных использоваться для определения надписей. Если значение атрибута равно `true`, то содержимое первой ячейки строки или столбца может быть использовано для ссылки на всю строку или столбец во многих операциях электронных таблиц, например, внутри сводных таблиц.

```
4472 <define name="table-database-range-attlist" combine="interleave">
4473     <optional>
4474         <attribute name="table:contains-header" a:defaultValue="true">
4475             <ref name="boolean"/>
4476         </attribute>
4477     </optional>
4478 </define>
```

Отображение кнопок фильтра

Атрибут кнопок `table:display-filter-buttons` определяет, будут ли отображаться кнопки фильтра. Кнопки фильтра — это списковые элементы управления, отображаемые в ячейках надписей. Элементами списка являются все

значения, существующие в обозначаемой надписью строке или столбце. При выборе одного из полей фильтр с его значением применяется к диапазону базы данных и выбирает все строки или столбцы, где ячейки в обозначенной надписью строке или столбце имеют выбранное значение.

```
<define name="table-database-range-attlist" combine="interleave">
    <optional>
        <attribute name="table:display-filter-buttons"
                 a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Адрес целевого диапазона

Атрибут `table:target-range-address` определяет адрес диапазона ячеек диапазона базы данных. Выделение абсолютных и относительных адресов невозможно. Поэтому в адресе должно быть определено имя таблицы, а символ доллара проигнорирован.

```
<define name="table-database-range-attlist" combine="interleave">
    <attribute name="table:target-range-address">
        <ref name="cellRangeAddress"/>
    </attribute>
</define>
```

Интервал обновления

Атрибут `table:refresh-delay` определяет интервал времени между автоматическими обновлениями.

```
<define name="table-database-range-attlist" combine="interleave">
    <optional>
        <attribute name="table:refresh-delay">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.6.2 SQL источника базы данных

Элемент `<table:database-source-sql>` описывает SQL-базу данных, которая содержит источник данных для диапазона базы данных.

```
<define name="table-database-source-sql">
    <element name="table:database-source-sql">
        <ref name="table-database-source-sql-attlist"/>
        <empty/>
    </element>
</define>
```

Имя базы данных

Атрибут `table:database-name` определяет имя SQL-базы данных, из которой импортируются данные.

```
<define name="table-database-source-sql-attlist" combine="interleave">
    <attribute name="table:database-name">
        <ref name="string"/>
    </attribute>
</define>
```

SQL-выражение

Атрибут `table:sql-statement` определяет SQL-выражение, используемое при импортировании данных из SQL-базы данных.

```
<define name="table-database-source-sql-attlist" combine="interleave">
    <attribute name="table:sql-statement">
        <ref name="string"/>
    </attribute>
</define>
```

Анализ SQL-выражения

Атрибут `table:parse-sql-statement` определяет, осуществляется ли анализ SQL-инструкций.

```
<define name="table-database-source-sql-attlist" combine="interleave">
    <optional>
        <attribute name="table:parse-sql-statement" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.6.3 Таблица источника базы данных

Элемент таблицы источника базы данных `<table:database-source-table>` определяет, что исходные данные диапазона базы данных хранятся в таблице базы данных.

```
<define name="table-database-source-query">
    <element name="table:database-source-table">
        <ref name="table-database-source-table-attlist"/>
        <empty/>
    </element>
</define>
```

Имя базы данных

Атрибут table:database-name определяет имя базы данных, из которой импортируются данные.

```
<define name="table-database-source-table-attlist" combine="interleave">
    <attribute name="table:database-name">
        <ref name="string"/>
    </attribute>
</define>
```

Имя таблицы

Атрибут table:database-table-name определяет имя таблицы базы данных, из которой импортируются данные.

```
<define name="table-database-source-table-attlist" combine="interleave">
    <attribute name="table:database-table-name">
        <ref name="string"/>
    </attribute>
</define>
```

8.6.4 Запрос источника базы данных

Элемент запроса источника базы данных <table:database-source-query> определяет, что исходные данные диапазона базы данных являются результатом запроса к базе данных.

```
<define name="table-database-source-table">
    <element name="table:database-source-query">
        <ref name="table-database-source-query-attlist"/>
        <empty/>
    </element>
</define>
```

Имя базы данных

Атрибут table:database-name определяет имя базы данных, из которой импортируются данные.

```
<define name="table-database-source-query-attlist" combine="interleave">
    <attribute name="table:database-name">
        <ref name="string"/>
    </attribute>
</define>
```

Имя запроса

Атрибут table:query-name определяет запрос, выполняемый к базе данных, из которой импортируются данные.

```
<define name="table-database-source-query-attlist" combine="interleave">
    <attribute name="table:query-name">
        <ref name="string"/>
    </attribute>
</define>
```

8.6.5 Сортировка

Элемент сортировки `<table:sort>` описывает принципы сортировки, которые должны быть применены к диапазону базы данных.

```
<define name="table-sort">
    <element name="table:sort">
        <ref name="table-sort-attlist"/>
        <oneOrMore>
            <ref name="table-sort-by"/>
        </oneOrMore>
    </element>
</define>
```

Привязка стилей к содержимому

Атрибут `table:bind-styles-to-content` определяет, сохраняют ли ячейки свои атрибуты стиля после выполнения операции сортировки.

```
<define name="table-sort-attlist" combine="interleave">
    <optional>
        <attribute name="table:bind-styles-to-content" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Адрес целевого диапазона

Если представлен необязательный атрибут `table:target-range-address`, результат сортировки копируется в определенный им диапазон ячеек вместо исходного диапазона ячеек, определенного диапазоном базы данных. Различия между абсолютными и относительными адресами невозможны. Поэтому в адресе должно присутствовать имя таблицы, а знак доллара игнорируется.

```
<define name="table-sort-attlist" combine="interleave">
    <optional>
        <attribute name="table:target-range-address">
            <ref name="cellRangeAddress"/>
        </attribute>
    </optional>
</define>
```

Чувствительность к регистру

Атрибут `table:case-sensitive` определяет, учитывать ли в операциях сортировки регистр символов.

```
<define name="table-sort-attlist" combine="interleave">
    <optional>
        <attribute name="table:case-sensitive" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Язык

Атрибут `table:language` определяет естественный язык, в котором осуществляется сравнение.

```
<define name="table-sort-attlist" combine="interleave">
    <optional>
        <attribute name="table:language">
            <ref name="languageCode"/>
        </attribute>
    </optional>
</define>
```

Страна

Атрибут `table:country` определяет наличие специфичных для страны правил, используемых при сортировке строк на конкретном естественном языке.

```
<define name="table-sort-attlist" combine="interleave">
    <optional>
        <attribute name="table:country">
            <ref name="countryCode"/>
        </attribute>
    </optional>
</define>
```

Алгоритм

Атрибут `table:algorithm` определяет алгоритм, используемый для сравнения сортируемых ключей. Значение атрибута определяется приложением, но язык и страна также влияют на название алгоритма сортировки, например, «фонетический (сначала алфавитно-цифровые)». Для недопущения разногласий между различными приложениями название должно начинаться с префикса пространства имен.

```
<define name="table-sort-attlist" combine="interleave">
    <optional>
```

```
        <attribute name="table:algorithm">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

8.6.6 Поле сортировки

Элемент сортировки посредством `<table:sort-by>` определяет ключ или поле для сортировки, тип данных этого поля и способ сортировки.

```
<define name="table-sort-by">
    <element name="table:sort-by">
        <ref name="table-sort-by-attlist"/>
        <empty/>
    </element>
</define>
```

Номер поля

Атрибут номера `table:field-number` определяет строку или столбец для сортировки. Это номер строки или столбца в диапазоне базы данных.

```
<define name="table-sort-by-attlist" combine="interleave">
    <attribute name="table:field-number">
        <ref name="nonNegativeInteger"/>
    </attribute>
</define>
```

Тип данных

Атрибут `table:data-type` определяет тип данных поля для сортировки. Он может принимать значения `text`, `number`, `automatic` или имя определяемого пользователем параметра сортировки. Если атрибут имеет значение `automatic`, то приложение должно определить тип данных в поле. Например, определяемый пользователем порядок сортировки — список названий месяцев. Задание определяемого пользователем порядка сортировки специфично для различных приложений.

```
<define name="table-sort-by-attlist" combine="interleave">
    <optional>
        <attribute name="table:data-type" a:defaultValue="automatic">
            <choice>
                <value>text</value>
                <value>number</value>
                <value>automatic</value>
                <ref name="string"/>
            </choice>
        </attribute>
    </optional>
</define>
```

Порядок

Атрибут table:order определяет, сортируются ли данные в порядке возрастания или убывания.

```
<define name="table-sort-by-attlist" combine="interleave">
    <optional>
        <attribute name="table:order" a:defaultValue="ascending">
            <choice>
                <value>ascending</value>
                <value>descending</value>
            </choice>
        </attribute>
    </optional>
</define>
```

8.6.7 Правила промежуточных сумм

Элемент правила промежуточных сумм <table:subtotal-rules> определяет, что для диапазона базы данных должны вычисляться промежуточные результаты (называемые промежуточными суммами). Он содержит информацию о строках и столбцах, для которых должны вычисляться промежуточные результаты, а также как эти результаты вычисляются. Для вычисления промежуточных результатов ячейки строк и столбцов группируются по их значениям, т. е. все ячейки с одинаковым содержимым в одном и том же поле образуют группу. Промежуточный результат вычисляется и отображается в конце каждой группы.

```
<define name="table-subtotal-rules">
    <element name="table:subtotal-rules">
        <ref name="table-subtotal-rules-attlist"/>
        <optional>
            <ref name="table-sort-groups"/>
        </optional>
        <zeroOrMore>
            <ref name="table-subtotal-rule"/>
        </zeroOrMore>
    </element>
</define>
```

Привязка стилей к содержимому

Атрибут table:bind-styles-to-content определяет, сохраняют ли ячейки свои атрибуты стиля после вычисления промежуточных сумм. Этот атрибут оценивается, только если имеется элемент table:sort-groups.

```
<define name="table-subtotal-rules-attlist" combine="interleave">
    <optional>
        <attribute name="table:bind-styles-to-content" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
```

```
    </optional>
</define>
```

Чувствительность к регистру

Атрибут `table:case-sensitive` определяет, имеет ли значение регистр символов при сравнении записей, например, при сортировке групп.

```
<define name="table-subtotal-rules-attlist" combine="interleave">
    <optional>
        <attribute name="table:case-sensitive" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Разрывы страницы при изменении группы

Атрибут разрыва страницы при изменении группы `table:page-breaks-on-group-change` определяет, должен ли вставляться разрыв страницы после промежуточного результата каждой группы.

```
<define name="table-subtotal-rules-attlist" combine="interleave">
    <optional>
        <attribute name="table:page-breaks-on-group-change"
                 a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.6.8 Отсортированные группы промежуточных сумм

Необязательный элемент отсортированных групп `<table:sort-groups>` определяет, что столбцы или строки сортируются перед группировкой, а также способ сортировки. Он относится к элементу правил промежуточных сумм (см. раздел 8.6.7).

```
<define name="table-sort-groups">
    <element name="table:sort-groups">
        <ref name="table-sort-groups-attlist"/>
        <empty/>
    </element>
</define>
```

Тип данных

Атрибут `table:data-type` определяет тип данных сортируемых столбца или строки. Детальная информация приведена в разделе 8.6.6.

```
<define name="table-sort-groups-attlist" combine="interleave">
    <optional>
        <attribute name="table:data-type" a:defaultValue="automatic">
            <choice>
                <value>text</value>
                <value>number</value>
                <value>automatic</value>
                <ref name="string"/>
            </choice>
        </attribute>
    </optional>
</define>
```

Порядок

Атрибут `table:order` определяет, сортируются ли данные группы в порядке возрастания или убывания. Детальная информация приведена в разделе 8.6.6.

```
<define name="table-sort-groups-attlist" combine="interleave">
    <optional>
        <attribute name="table:order" a:defaultValue="ascending">
            <choice>
                <value>ascending</value>
                <value>descending</value>
            </choice>
        </attribute>
    </optional>
</define>
```

8.6.9 Правило промежуточной суммы

Элемент правила промежуточной суммы `<table:subtotal-rule>` определяет способ вычисления промежуточной суммы для конкретной строки или столбца. Правило содержит номер поля группировки, определяющий группу столбцов, для которых используется правило, и одно поле или более полей промежуточных сумм, определяющих строку или столбец, где должны быть вычислены промежуточные суммы, а также функцию, используемую для вычисления.

```
<define name="table-subtotal-rule">
    <element name="table:subtotal-rule">
        <ref name="table-subtotal-rule-attlist"/>
        <zeroOrMore>
            <ref name="table-subtotal-field"/>
        </zeroOrMore>
    </element>
</define>
```

Группировка по номеру поля

Атрибут table:group-by-field-number определяет поле, например столбец, по которому должна осуществляться группировка. Это номер строки или столбца в диапазоне базы данных.

```
<define name="table-subtotal-rule-attlist" combine="interleave">
    <attribute name="table:group-by-field-number">
        <ref name="nonNegativeInteger"/>
    </attribute>
</define>
```

8.6.10 Поле промежуточной суммы

Элемент поля промежуточной суммы <table:subtotal-field> содержит номер поля и функцию, которая используется для вычисления промежуточной суммы.

```
<define name="table-subtotal-field">
    <element name="table:subtotal-field">
        <ref name="table-subtotal-field-attlist"/>
        <empty/>
    </element>
</define>
```

Номер поля

Атрибут table:field-number определяет строку или столбец, для которых должны вычисляться промежуточные суммы. Это номер строки или столбца в диапазоне базы данных.

```
<define name="table-subtotal-field-attlist" combine="interleave">
    <attribute name="table:field-number">
        <ref name="nonNegativeInteger"/>
    </attribute>
</define>
```

Функция

Атрибут table:function определяет способ вычисления промежуточной суммы. Атрибут принимает следующие значения: auto, average, count, countnums, max, min, product, stdev, stdevp, sum, var и varp.

```
<define name="table-subtotal-field-attlist" combine="interleave">
    <attribute name="table:function">
        <choice>
            <value>auto</value>
            <value>average</value>
            <value>count</value>
            <value>countnums</value>
```

```
            <value>max</value>
            <value>min</value>
            <value>product</value>
            <value>stdev</value>
            <value>stdevp</value>
            <value>sum</value>
            <value>var</value>
            <value>varp</value>
            <ref name="string"/>
        </choice>
    </attribute>
</define>
```

Пример – Поле промежуточных сумм:

```
<table:database-range table:range-
position="sampletable.A1:sampletable.G20" table:name="sample">
    <table:database-source-table table:database-name="sampleDB"
table:table-name="sampleTable"/>
    <table:filter ...>
        ...
    </table:filter>
    <table:sort>
        <table:sort-by table:field-number=1/>
    </table:sort>
    <table:subtotal-rules>
        <table:sort-groups/>
        <table:subtotal-rule table:column-group "3">
            <table:subtotal-field table:field-number="1"
                                  table:function="sum"/>
        </table:subtotal-rule>
    </table:subtotal-rules>
</table:database-range>
```

8.7 Фильтры

Фильтры определяют, что должны быть видимыми только строки, удовлетворяющие определенным условиям.

8.7.1 Фильтр таблицы

Элемент фильтра таблицы `<table:filter>` определяет, как фильтруются данные, содержащиеся в диапазоне базы данных или сводной таблице. Условие, определенное в элементе, применяется ко всем строкам, определенным в диапазоне базы данных или сводной таблице. Строки, в которых условие не принимает истинное значение, делаются невидимыми.

```
<define name="table-filter">
    <element name="table:filter">
        <ref name="table-filter-attlist"/>
        <choice>
            <ref name="table-filter-condition"/>
            <ref name="table-filter-and"/>
            <ref name="table-filter-or"/>
        </choice>
```

```
    </element>
</define>
```

Адрес целевого диапазона

Если представлен необязательный атрибут `table:target-range-address` результат работы фильтра копируется в определенный диапазон ячеек, но все строки таблицы остаются видимыми. Выделение абсолютных и относительных адресов невозможно. Поэтому в адресе должно быть определено имя таблицы, а символ доллара проигнорирован.

```
<define name="table-filter-attlist" combine="interleave">
    <optional>
        <attribute name="table:target-range-address">
            <ref name="cellRangeAddress"/>
        </attribute>
    </optional>
</define>
```

Источник условия

Атрибут `table:condition-source` определяет, содержится ли условие в фильтре или приведено в диапазоне таблицы. Если атрибут имеет значение `self`, то условие определено в дочернем элементе элемента `<table:filter>`. Если атрибут имеет значение `cell-range`, условие содержится в диапазоне ячеек, определенном атрибутом `table:condition-source-range-address`.

```
<define name="table-filter-attlist" combine="interleave">
    <optional>
        <attribute name="table:condition-source" a:defaultValue="self">
            <choice>
                <value>self</value>
                <value>cell-range</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Адрес диапазона источника условия

Атрибут `table:condition-source-range-address` определяет диапазон ячеек, который содержит условие. Первая строка диапазона должна содержать надписи столбцов, содержимое которых должно быть отфильтровано. Следующие строки содержат условия, которые должны выполняться для ячеек, содержащихся в столбце. Условия в каждой из строк объединяются посредством логической операции «и», в то время как строки — посредством логической операции «или».

Это означает, что строки исходной таблицы отображаются, если в диапазоне условия имеется по крайней мере одна строка, в которой все условия, примененные к столбцам, определенным в первой строке диапазона условия, приняли истинное значение.

Пример –

Если диапазон источника условия E1:F3 (выделено желтым) и исходный диапазон A1:C3 (выделено зеленым), будут отображены только строки 2 и 3.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*G*	*I*
1	1	3	4		A	B			
2	1	5	6		=1	=5			
3	2	8	9		>=2				

Строка 2 отображается потому, что ячейка в столбце A имеет значение 1, а ячейка в столбце B — 5, т. е. все условия 2-й строки диапазона условия приняли истинное значение. Строка 3 отображается потому, что ячейка в столбце A больше или равна 2, и, поскольку это единственное условие, все условия 3-й строки принимают истинное значение.

```
<define name="table-filter-attlist" combine="interleave">
    <optional>
        <attribute name="table:condition-source-range-address">
            <ref name="cellRangeAddress"/>
        </attribute>
    </optional>
</define>
```

Отображение дубликатов

Атрибут `table:display-duplicates` определяет, должны ли отображаться повторяющиеся пригодные результаты.

```
<define name="table-filter-attlist" combine="interleave">
    <optional>
        <attribute name="table:display-duplicates" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.7.2 Фильтр And (И)

Элемент `<table:filter-and>` определяет, что к условиям, определенным в дочерних элементах данного элемента, применяется логический оператор AND (И).

```
<define name="table-filter-and">
    <element name="table:filter-and">
        <oneOrMore>
            <choice>
                <ref name="table-filter-or"/>
                <ref name="table-filter-condition"/>
            </choice>
        </oneOrMore>
    </element>
</define>
```

8.7.3 Фильтр Or (ИЛИ)

Элемент `<table:filter-or>` определяет, что к условиям, определенным в дочерних элементах данного элемента, применяется логический оператор OR (ИЛИ).

```
<define name="table-filter-or">
    <element name="table:filter-or">
        <oneOrMore>
            <choice>
                <ref name="table-filter-and"/>
                <ref name="table-filter-condition"/>
            </choice>
        </oneOrMore>
    </element>
</define>
```

8.7.4 Условие фильтра

Элемент таблицы `<table:filter-condition>` определяет одно условие, применяемое при фильтрации.

```
<define name="table-filter-condition">
    <element name="table:filter-condition">
        <ref name="table-filter-condition-attlist"/>
        <empty/>
    </element>
</define>
```

Номер поля

Атрибут номер поля `table:field-number` определяет, какое поле используется для условия. Номер поля — это номер строки или столбца в исходном диапазоне фильтра.

```
<define name="table-filter-condition-attlist" combine="interleave">
    <attribute name="table:field-number">
        <ref name="nonNegativeInteger"/>
    </attribute>
</define>
```

Значение

Атрибут table:value определяет значение для условия фильтра.

```
<define name="table-filter-condition-attlist" combine="interleave">
    <attribute name="table:value">
        <ref name="string"/>
    </attribute>
</define>
```

Оператор

Атрибут оператор table:operator определяет, какой оператор используется в условии фильтра. Это означает, что каждая ячейка, содержащаяся в столбце, определенном номером поля (т. е. атрибутом table:field-number), сравнивается со значением (т. е. атрибутом table:value), с использованием данного оператора. Результат этого сравнения — это результат условия фильтра, определенного элементом <table:filter-condition>.

Операторы могут использовать или не использовать регулярные выражения. Следующие операторы используют регулярные выражения:

- match (совпадает);
- !match (не совпадает).

В обоих случаях атрибут table:value содержит регулярное выражение, с которым ячейки таблицы должны совпадать или не совпадать.

Операторы отношения, которые не используются в регулярных выражениях:

- = (равно);
- != (неравно);
- < (меньше);
- > (больше);
- <= (меньше или равно);
- >= (больше или равно).

Дополнительно могут использоваться empty (пусто), !empty (не пусто), bottom values (нижнее значение), top values (верхнее значение), bottom percent (нижний процент) и top percent (верхний процент). Например, для выборки нижней и верхней величины в процентах могут использоваться два последних оператора.

```
<define name="table-filter-condition-attlist" combine="interleave">
    <attribute name="table:operator">
```

```
        <ref name="string"/>
    </attribute>
</define>
```

Чувствительность к регистру

Атрибут `table:case-sensitive` определяет, чувствительно ли условие фильтра к регистру.

```
<define name="table-filter-condition-attlist" combine="interleave">
    <optional>
        <attribute name="table:case-sensitive" a:defaultValue="false">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Тип данных

Атрибут `table:data-type` определяет, будут ли сравниваться значения как текстовые или числовые.

```
<define name="table-filter-condition-attlist" combine="interleave">
    <optional>
        <attribute name="table:data-type" a:defaultValue="text">
            <choice>
                <value>text</value>
                <value>number</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Пример – Представление фильтра:

```
<filter>
    <filter-or>
        <filter-and>
            <filter-condition table:field-number=1 table:operator="="
                              table:value="Doe"/>
            <filter-condition table:field-number=2 table:operator="="
                              table:value="John"/>
        </filter-and>
        <filter-and>
            <filter-condition table:field-number=1 table:operator="="
                              table:value="Burns"/>
            <filter-condition table:field-number=2 table:operator="="
                              table:value="Michael"/>
        </filter-and>
    </filter-or>
</filter>
```

8.8 Сводные таблицы

Сводные таблицы позволяют анализировать и оценивать данные, содержащиеся в электронных таблицах. Элемент сводные таблицы

`<table:data-pilot-tables>` содержит перечень всех сводных таблиц документа.

```
<define name="table-data-pilot-tables">
    <element name="table:data-pilot-tables">
        <zeroOrMore>
            <ref name="table-data-pilot-table"/>
        </zeroOrMore>
    </element>
</define>
```

8.8.1 Сводная таблица

Элемент `<table:data-pilot-table>` определяет одну сводную таблицу. Внутри сводных таблиц собираются все комбинации значений, существующих в выделенных столбцах, и для каждой из комбинаций к ячейкам других столбцов применяется формула.

Пример —

Дана следующая исходная таблица:

	A	B	C	D	
1	**Изделие**	**Город**	**Страна**	**Количество**	**Цена**
2	Системный блок	Гамбург	Германия	1	12
3	Монитор	Гамбург	Германия	2	15
4	Принтер	Париж	Франция	4	13
5	Монитор	Париж	Франция	2	14
6	Системный блок	Париж	Франция	1	12
7	Монитор	Гамбург	Германия	2	10
8	Принтер	Париж	Франция	2	16

Следующая сводная таблица группирует исходную таблицу по столбцам «Страна», «Город» и «Изделие», и вычисляются суммы столбцов «Количество» и «Цена» для каждой из комбинаций значений этих трех столбцов. Значения столбцов «Страна» и «Город» приведены в столбцах, в то время как значения столбца «Изделие» — в строках.

			Изделие			
Страна	Город	Данные	Системный блок	Монитор	Принтер	**Итого**
Франция	Париж	Сумма — Количество	1	2	6	**9**
		Сумма — Цена	12	14	29	**55**
Германия	Гамбург	Сумма — Количество	1	4		**5**
		Сумма — Цена	12	25		**37**
Итого сумма — Количество			**2**	**6**	**6**	**14**
Итого сумма — Цена			**24**	**39**	**29**	**92**

Столбцы, используемые для группировки (здесь «Страна», «Город» и «Изделие») называются столбцами категорий. Столбцы, для которых на основе комбинаций значений столбцов категорий вычисляется формула (здесь «Количество» и «Цена»), называются столбцами данных. Отдельные значения, которые содержатся в столбце категории, называются элементами.

В основном, поведение сводной таблицы определяется полями, где каждое поле имеет имя и, так называемую, ориентацию. Столбцы категорий определяются полями с ориентацией `row` (строка) или `column` (столбец), а столбцы данных определяются полями, которые имеют ориентацию `data` (данные). В примере выше «Изделие» — это поле со столбцовой ориентацией, в то время как «Страна» и «Город» — поля со строковой. «Количество» и «Цена» — поля с ориентацией `data`.

Третий вид полей — это поля разметки данных. Поля разметки данных не связаны со столбцами исходной таблицы и используются только для изменения разметки сводной таблицы. Например, «Данные» — это поле разметки данных.

Порядок, в котором определяются поля — реляционный. Они определяются в порядке, в котором данные столбцов категорий группируются, а результаты отображаются. Приведенная ниже сводная таблица демонстрирует, как меняется сводная таблица, если, например, поле разметки данных определяется перед полями столбцов категорий.

Пример – Сводная таблица с модифицированной разметкой:

			Изделие			
Данные	Страна	Город	Системный блок	Монитор	Принтер	**Итого**
Сумма — Количество	Франция	Париж	1	2	6	**9**
	Германия	Гамбург	1	4		**5**
Сумма — Цена	Франция	Париж	12	14	29	**55**
	Германия	Гамбург	12	25		**37**
Итого сумма — Количество			**2**	**6**	**6**	**14**
Итого сумма — Цена			**24**	**39**	**29**	**92**

С элементом сводная таблица могут быть связаны следующие атрибуты:

- имя сводной таблицы;
- данные приложения;
- итоговая сумма;
- пропуск пустых строк;
- распознавание категории;
- адрес целевого диапазона;
- отображение кнопки фильтра;
- переход к деталям по двойному щелчку.

```
<define name="table-data-pilot-table">
    <element name="table:data-pilot-table">
        <ref name="table-data-pilot-table-attlist"/>
        <optional>
            <choice>
                <ref name="table-database-source-sql"/>
                <ref name="table-database-source-table"/>
                <ref name="table-database-source-query"/>
                <ref name="table-source-service"/>
                <ref name="table-source-cell-range"/>
            </choice>
        </optional>
        <oneOrMore>
            <ref name="table-data-pilot-field"/>
        </oneOrMore>
    </element>
</define>
```

Источник сводной таблицы

Источником сводной таблицы может быть база данных, т. е. таблица базы данных, SQL-запрос или именованный запрос, или диапазон ячеек,

расположенный в самом документе. Он также может обеспечиваться внешним компонентом путем, зависящим от реализации.

Источник сводной таблицы определяется одним из следующих элементов, содержащихся в элементе `<table:data-pilot-table>` :

- `<table:database-source-sql>` (см. раздел 8.6.2);
- `<table:database-source-table>` (см. раздел 8.6.3);
- `<table:database-source-query>` (см. раздел 8.6.4);
- `<table:source-cell-range>` (см. раздел 8.8.2);
- `<table:source-service>` (см. раздел 8.8.3).

Имя сводной таблицы

Атрибут `table:name` определяет имя сводной таблицы.

```
<define name="table-data-pilot-table-attlist" combine="interleave">
    <attribute name="table:name">
        <ref name="string"/>
    </attribute>
</define>
```

Данные приложения

Атрибут `table:application-data` определяет дополнительную информацию о сводной таблице, которая может быть использована приложением, например в макросах. Эти данные не оказывают воздействия на поведение сводной таблицы.

```
<define name="table-data-pilot-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:application-data">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Итоговая сумма

Атрибут `table:grand-total` определяет, где в дополнение к значениям, вычисленным для каждой комбинации значений в столбцах категорий, должна отображаться итоговая сумма: в столбцах, строках или в обоих. В примере выше итоговая сумма разрешена. Она отображается в строке и столбце, обозначенном «Итого».

```
<define name="table-data-pilot-table-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="table:grand-total" a:defaultValue="both">
            <choice>
                <value>none</value>
                <value>row</value>
                <value>column</value>
                <value>both</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Пропуск пустых строк

Атрибут `table:ignore-empty-rows` определяет, должны ли пропускаться пустые строки в исходном диапазоне.

```
<define name="table-data-pilot-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:ignore-empty-rows" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Распознавание категории

Атрибут `table:identify-categories` определяет, должны ли строки, не содержащие значения в одном из столбцов категорий, использовать значение ближайшей предшествующей строки, имеющей значение, или такие строки должны перемещаться в собственную группу (или категорию). Если атрибут имеет значение `false`, пустые значения формируют собственную категорию.

```
<define name="table-data-pilot-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:identify-categories" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Адрес целевого диапазона

Атрибут `table:target-range-address` определяет целевой диапазон для вывода сводной таблицы, т. е. ее отображения. Выделение абсолютных и относительных адресов невозможно, т. е. адрес интерпретируется как абсолютный, даже если он содержит знаки доллара. Адрес диапазона должен содержать имя таблицы.

```
<define name="table-data-pilot-table-attlist" combine="interleave">
```

```
    <attribute name="table:target-range-address">
        <ref name="cellRangeAddress"/>
    </attribute>
</define>
```

Кнопки

Внутри сводной таблицы некоторые ячейки для обеспечения интерактивных операций с таблицей, подобных изменению порядка столбцов, могут отображаться как кнопки. Атрибут `table:buttons` определяет все ячейки, которые должны отображаться таким образом. Его значением является список адресов ячеек. Выделение абсолютных и относительных адресов невозможно, т. е. адрес интерпретируется как абсолютный, даже если он содержит знаки доллара. Все адреса должны содержать имя таблицы.

В примерах выше ячейки кнопок отображаются с серым фоном.

```
<define name="table-data-pilot-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:buttons">
            <ref name="cellRangeAddressList"/>
        </attribute>
    </optional>
</define>
```

Отображение кнопки фильтра

Атрибут `table:show-filter-button` определяет, должна ли внутри сводной таблицы в пользовательском интерфейсе отображаться кнопка фильтра. При нажатии на кнопку фильтра отображается диалог фильтра.

```
<define name="table-data-pilot-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:show-filter-button" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Переход к деталям по двойному щелчку

Атрибут `table:drill-down-on-double-click` определяет, как сводная таблица реагирует на двойной щелчок мышью внутри ее. Если значение атрибута `false`, двойной щелчок на надписи элемента или на пустой области приведет к переходу в режим редактирования ячейки, подобно ячейкам за пределами сводной таблицы. Это может использоваться для переименования полей групп или элементов. Если значение атрибута `true`, двойной щелчок на надписи

элемента или на пустой области приведет к появлению или скрытию деталей этого элемента. Двойной щелчок в любом другом месте сводной таблицы не приведет к изменениям.

```
<define name="table-data-pilot-table-attlist" combine="interleave">
    <optional>
        <attribute name="table:drill-down-on-double-click"
                   a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.8.2 Источник-диапазон ячеек

Если источником сводной таблицы является диапазон ячеек, элемент `<table:source-cell-range>` содержит информацию о диапазоне ячеек и о том, как сводная таблица получает данные из диапазона. Перед тем как исходные данные преобразуются в данные сводной таблицы, к ним должен быть применен фильтр. Этот фильтр должен быть определен дочерним элементом `<table:filter>`.

```
<define name="table-source-cell-range">
    <element name="table:source-cell-range">
        <ref name="table-source-cell-range-attlist"/>
        <optional>
            <ref name="table-filter"/>
        </optional>
    </element>
</define>
```

С данным элементом может быть связан атрибут:

- адрес диапазона ячеек.

Адрес диапазона ячеек

Атрибут `table:cell-range-address` определяет диапазон ячеек, содержащий исходные данные. Адрес исходного диапазона ячеек должен быть абсолютным. Поэтому адрес диапазона ячеек должен содержать имя таблицы, а знаки доллара пропускаются.

```
<define name="table-source-cell-range-attlist" combine="interleave">
    <attribute name="table:cell-range-address">
        <ref name="cellRangeAddress"/>
    </attribute>
</define>
```

8.8.3 Источник-служба

Источником сводной таблицы может быть «служба», т. е. данные могут предоставляться внешним компонентом. Элемент `<table:source-service>` содержит информацию о службе, которая используется для создания сводной таблицы.

```
<define name="table-source-service">
    <element name="table:source-service">
        <ref name="table-source-service-attlist"/>
        <empty/>
    </element>
</define>
```

С данным элементом могут быть связаны следующие атрибуты:

- имя службы;
- имя источника;
- имя объекта;
- имя пользователя источника;
- пароль источника.

Имя службы

Атрибут `table:name` определяет имя службы. Его значение зависит от конкретной реализации.

```
<define name="table-source-service-attlist" combine="interleave">
    <attribute name="table:name">
        <ref name="string"/>
    </attribute>
</define>
```

Имя источника

Атрибут `table:source-name` определяет имя источника, который относится к реализации службы. Его значение определяется особенностями приложения и службы.

```
<define name="table-source-service-attlist" combine="interleave">
    <attribute name="table:source-name">
        <ref name="string"/>
    </attribute>
</define>
```

Имя объекта

Атрибут table:object-name определяет имя объекта в источнике, который относится к реализации службы. Его значение определяется особенностями приложения и службы.

```
<define name="table-source-service-attlist" combine="interleave">
    <attribute name="table:object-name">
        <ref name="string"/>
    </attribute>
</define>
```

Имя пользователя источника

Атрибут table:user-name определяет имя пользователя, необходимое для доступа к источнику. Он относится к реализации службы. Его значение определяется особенностями приложения и службы.

```
<define name="table-source-service-attlist" combine="interleave">
    <optional>
        <attribute name="table:user-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Пароль источника

Атрибут table:password определяет пароль, необходимый для доступа к источнику. Он относится к реализации службы. Его значение определяется особенностями приложения и службы.

```
<define name="table-source-service-attlist" combine="interleave">
    <optional>
        <attribute name="table:password">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

8.8.4 Поле сводной таблицы

Поля сводной таблицы определяются элементами <table:data-pilot-field>.

```
<define name="table-data-pilot-field">
    <element name="table:data-pilot-field">
        <ref name="table-data-pilot-field-attlist"/>
        <optional>
            <ref name="table-data-pilot-level"/>
        </optional>
```

```
4983        <optional>
4984            <ref name="table-data-pilot-field-reference"/>
4985        </optional>
4986        <optional>
4987            <ref name="table-data-pilot-groups"/>
4988        </optional>
4989    </element>
4990 </define>
```

С данным элементом могут быть связаны следующие атрибуты:

- имя поля источника;
- ориентация;
- поле разметки данных;
- функция;
- используемая иерархия.

Имя поля источника

Для полей, которые определяют столбцы категорий или данных, атрибут `table:source-field-name` определяет имя или надпись столбца, в котором содержится поле. Если источником сводной таблицы является, например, диапазон ячеек, значением атрибута должна быть надпись столбца.

Может быть несколько элементов `<table:data-pilot-field>` с одним и тем же значением данного атрибута.

```
4991 <define name="table-data-pilot-field-attlist" combine="interleave">
4992     <attribute name="table:source-field-name">
4993         <ref name="string"/>
4994     </attribute>
4995 </define>
```

Ориентация

Атрибут `table:orientation` определяет ориентацию поля источника. Если значение атрибута `data`, то поле определяет столбец данных. Если значение атрибута `row` или `column`, то поле определяет столбец категории. Значение `hidden` (скрытый) используется для полей, которые имеют соответствующие столбцы в источнике, но невидимы в сводной таблице. Значение `page` указывает, что для соответствующего столбца должен быть сформирован автоматический фильтр (т. е. некоторый, позволяющий выбрать одно из значений, содержащихся в таблице). В этом случае для столбца должно существовать дополнительное поле с ориентацией `row`, `column` или `data`.

Если значение атрибута равно page, для определения выбранного для фильтра значения может использоваться атрибут table:selected-page.

```
<define name="table-data-pilot-field-attlist" combine="interleave">
    <choice>
        <attribute name="table:orientation">
            <choice>
                <value>row</value>
                <value>column</value>
                <value>data</value>
                <value>hidden</value>
            </choice>
        </attribute>
        <group>
            <attribute name="table:orientation">
                <value>page</value>
            </attribute>
            <attribute name="table:selected-page">
                <ref name="string"/>
            </attribute>
        </group>
    </choice>
</define>
```

Поле разметки данных

Атрибут table:is-data-layout-field определяет, является ли поле полем разметки данных (см. раздел 8.8.1). Поля разметки данных обычно не имеют имени.

```
<define name="table-data-pilot-field-attlist" combine="interleave">
    <optional>
        <attribute name="table:is-data-layout-field" a:defaultValue="false">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Функция

Атрибут table:function определяет функцию, которая применяется к значениям ячеек столбцов данных. Он рассматривается только в случае, если атрибут table:orientation имеет значение data. Возможные значения данного атрибута: auto, average, count, countnums, max, min, product, stdev, stdevp, sum, var и varp. Для столбцов категорий значение атрибута auto может использоваться для определения того, что функция вообще не применяется.

```
<define name="table-data-pilot-field-attlist" combine="interleave">
    <optional>
        <attribute name="table:function">
```

```
            <choice>
                <value>auto</value>
                <value>average</value>
                <value>count</value>
                <value>countnums</value>
                <value>max</value>
                <value>min</value>
                <value>product</value>
                <value>stdev</value>
                <value>stdevp</value>
                <value>sum</value>
                <value>var</value>
                <value>varp</value>
                <ref name="string"/>
            </choice>
        </attribute>
    </optional>
</define>
```

Используемая иерархия

Если источник сводной таблицы предоставляется внешним компонентом или службой, данные, содержащиеся в столбцах категорий, могут быть не только сгруппированы по значению, но и могут быть далее разделены на подгруппы или иерархии. Значения дат, например, могут быть группированы по правилу:

- «год», «месяц» и «день месяца»;
- «год», «неделя» и «день недели».

Если внешние компоненты поддерживают иерархии, то должны быть назначены уникальные номера. Эти номера могут быть использованы в атрибуте `table:used-hierarchy` для выбора иерархии, которая должна быть применена к полю источника. Значение по умолчанию означает, что иерархия вообще не применяется.

```
<define name="table-data-pilot-field-attlist" combine="interleave">
    <optional>
        <attribute name="table:used-hierarchy" a:defaultValue="-1">
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

8.8.5 Уровень сводной таблицы

Элемент уровень сводной таблицы `<table:data-pilot-level>` содержит дополнительную информацию о полях сводной таблицы.

```
<define name="table-data-pilot-level">
    <element name="table:data-pilot-level">
        <ref name="table-data-pilot-level-attlist"/>
        <optional>
```

```
            <ref name="table-data-pilot-subtotals"/>
        </optional>
        <optional>
            <ref name="table-data-pilot-members"/>
        </optional>
        <optional>
            <ref name="table-data-pilot-display-info"/>
        </optional>
        <optional>
            <ref name="table-data-pilot-sort-info"/>
        </optional>
        <optional>
            <ref name="table-data-pilot-layout-info"/>
        </optional>
    </element>
</define>
```

С данным элементом может быть связан атрибут:

- показ пустых полей.

Показ пустых полей

Атрибут `table:show-empty` определяет, должны ли отображаться поля, не имеющие ни одного элемента. Если данный атрибут не представлен, то приложение может либо отображать, либо не отображать такие поля.

```
<define name="table-data-pilot-level-attlist" combine="interleave">
    <optional>
        <attribute name="table:show-empty">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.8.6 Промежуточные суммы сводной таблицы

Элемент промежуточных сумм сводных таблиц `<table:data-pilot-subtotals>` содержит информацию о промежуточных результатах, которые отображаются для каждого элемента поля и функции, используемой для вычисления результата. Несколько промежуточных результатов могут вычисляться одновременно. Если элемент не представлен, приложение может отображать либо не отображать промежуточные результаты.

```
<define name="table-data-pilot-subtotals">
    <element name="table:data-pilot-subtotals">
        <zeroOrMore>
            <ref name="table-data-pilot-subtotal"/>
        </zeroOrMore>
    </element>
</define>
```

8.8.7 Промежуточная сумма сводной таблицы

Элемент промежуточной суммы сводной таблицы `<table:data-pilot-subtotal>` содержит информацию об отдельном вычислении промежуточного результата.

```
<define name="table-data-pilot-subtotal">
    <element name="table:data-pilot-subtotal">
        <ref name="table-data-pilot-subtotal-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<table:data-pilot-subtotal>` может быть связан атрибут:

- функция.

Функция

Атрибут `table:function` определяет функцию, используемую для вычисления промежуточной суммы. Возможные функции: `auto`, `average`, `count`, `countnums`, `max`, `min`, `product`, `stdev`, `stdevp`, `sum`, `var` и `varp`.

```
<define name="table-data-pilot-subtotal-attlist" combine="interleave">
    <attribute name="table:function">
        <choice>
            <value>auto</value>
            <value>average</value>
            <value>count</value>
            <value>countnums</value>
            <value>max</value>
            <value>min</value>
            <value>product</value>
            <value>stdev</value>
            <value>stdevp</value>
            <value>sum</value>
            <value>var</value>
            <value>varp</value>
            <ref name="string"/>
        </choice>
    </attribute>
</define>
```

8.8.8 Элементы сводной таблицы

Для столбцов категорий возможно управление текущим отображением определенных элементов или отображением информации для определенных элементов. Элемент `<table:data-pilot-members>` содержит такую информацию.

```
<define name="table-data-pilot-members">
    <element name="table:data-pilot-members">
```

```
        <zeroOrMore>
            <ref name="table-data-pilot-member"/>
        </zeroOrMore>
    </element>
</define>
```

8.8.9 Элемент сводной таблицы

Элемент `<table:data-pilot-member>` определяет, какая информация отображается для конкретного элемента сводной таблицы.

```
<define name="table-data-pilot-member">
    <element name="table:data-pilot-member">
        <ref name="table-data-pilot-member-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом сводной таблицы могут быть связаны следующие атрибуты:

- имя элемента;
- отображение;
- показ деталей.

Имя элемента

Атрибут `table:name` определяет значение, для которого задается отображаемая информация.

```
<define name="table-data-pilot-member-attlist" combine="interleave">
    <attribute name="table:name">
        <ref name="string"/>
    </attribute>
</define>
```

Отображение

Атрибут `table:display` определяет, будет ли вообще видим элемент сводной таблицы. Если данный атрибут не представлен, приложение может отображать либо не отображать элемент.

```
<define name="table-data-pilot-member-attlist" combine="interleave">
    <optional>
        <attribute name="table:display">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Показ деталей

Атрибут table:show-details определяет, будут ли отображаться дополнительные поля для элемента. Этот атрибут меняет поведение сводной таблицы, только если имеется несколько полей с ориентацией row или column. В этом случае и при значении атрибута, равном false, для поля с ориентацией row или column, не являющегося последним с такой ориентацией, элементы всех следующих полей, имеющих такую же ориентацию, отображаться не будут. Вместо этого данные этих полей будут суммироваться.

```
<define name="table-data-pilot-member-attlist" combine="interleave">
    <optional>
        <attribute name="table:show-details">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.8.10 Информация об отображении сводной таблицы

Элемент <table:data-pilot-display-info> ограничивает количество строк, отображаемых для поля категории до определенного числа значений поля данных.

```
<define name="table-data-pilot-display-info">
    <element name="table:data-pilot-display-info">
        <ref name="table-data-pilot-display-info-attlist"/>
        <empty/>
    </element>
</define>
```

Разрешение

Атрибут table:enabled определяет, принимается ли во внимание элемент <table:data-pilot-display-info>.

```
<define name="table-data-pilot-display-info-attlist" combine="interleave">
    <attribute name="table:enabled">
        <ref name="boolean"/>
    </attribute>
</define>
```

Поле данных

Атрибут table:data-field определяет поле данных, значения которого принимаются во внимание.

```
<define name="table-data-pilot-display-info-attlist" combine="interleave">
```

```
    <attribute name="table:data-field">
        <ref name="string"/>
    </attribute>
</define>
```

Количество элементов

Атрибут table:member-count определяет, сколько значений с верха или с низа столбца поля данных отображается.

```
<define name="table-data-pilot-display-info-attlist" combine="interleave">
    <attribute name="table:member-count">
        <ref name="nonNegativeInteger"/>
    </attribute>
</define>
```

Режим отображения элемента

Атрибут table:display-member-mode определяет, должны ли значения, определенные атрибутом table:member-count, быть получены из верха или из низа столбца поля данных.

```
<define name="table-data-pilot-display-info-attlist" combine="interleave">
    <attribute name="table:display-member-mode">
        <choice>
            <value>from-top</value>
            <value>from-bottom</value>
        </choice>
    </attribute>
</define>
```

8.8.11 Информация о сортировке сводной таблицы

Элемент <table:data-pilot-sort-info> определяет способ сортировки элементов, хранящихся в поле категории.

```
<define name="table-data-pilot-sort-info">
    <element name="table:data-pilot-sort-info">
        <ref name="table-data-pilot-sort-info-attlist"/>
        <empty/>
    </element>
</define>
```

Режим сортировки

Атрибут table:sort-mode описывает, как сортируются элементы одного поля сводной таблицы. При использовании режима data, элементы текущего поля категории сортируются в соответствии с их значениями в поле данных, определенном атрибутом data-field. При использовании режима manual,

пользователь может сортировать элементы в поле вручную. Если установлен режим `name`, элементы в поле сортируются по их именам.

```
<define name="table-data-pilot-sort-info-attlist" combine="interleave">
    <choice>
        <group>
            <attribute name="table:sort-mode">
                <value>data</value>
            </attribute>
            <attribute name="table:data-field">
                <ref name="string"/>
            </attribute>
        </group>
        <attribute name="table:sort-mode">
            <choice>
                <value>none</value>
                <value>manual</value>
                <value>name</value>
            </choice>
        </attribute>
    </choice>
</define>
```

Порядок сортировки

Атрибут `table:sort-order` определяет, должны ли элементы сортироваться в порядке возрастания или убывания.

```
<define name="table-data-pilot-sort-info-attlist" combine="interleave">
    <attribute name="table:order">
        <choice>
            <value>ascending</value>
            <value>descending</value>
        </choice>
    </attribute>
</define>
```

8.8.12 Информация о разметке сводной таблицы

Элемент `<table:data-pilot-layout-info>` описывает, каким образом размечается поле.

```
<define name="table-data-pilot-layout-info">
    <element name="table:data-pilot-layout-info">
        <ref name="table-data-pilot-layout-info-attlist"/>
        <empty/>
    </element>
</define>
```

Режим разметки

Атрибут `table:layout-mode` определяет, каким образом размечается поле. Он может принимать следующие значения:

- `tabular-layout`

  Режим табулярной разметки — это разметка, при которой каждое имя элемента расположено на той же строке, что и первый элемент следующего поля. В этом режиме промежуточные суммы всегда отображаются ниже данных элементов.

- `outline-subtotals-top`

  В режиме структуры с промежуточными суммами сверху элементы из следующего поля начинаются в строке ниже имени элемента, подобно обычным отчетам баз данных. Промежуточные суммы отображаются вверху (в той же строке, что и имя элемента). Когда промежуточные суммы занимают более одной строки (выбрано вручную или из-за наличия нескольких полей данных), они всегда отображаются ниже данных элемента, независимо от настроек.

- `outline-subtotals-bottom`

  Подобен `outline-subtotals-top`, за исключением того, что промежуточные суммы отображаются снизу (ниже данных элементов, как и в режиме табулярной разметки).

```
5210 <define name="table-data-pilot-layout-info-attlist" combine="interleave">
5211     <attribute name="table:layout-mode">
5212         <choice>
5213             <value>tabular-layout</value>
5214             <value>outline-subtotals-top</value>
5215             <value>outline-subtotals-bottom</value>
5216         </choice>
5217     </attribute>
5218 </define>
```

Добавление пустых строк

Если атрибут `table:add-empty-lines` имеет значение `true`, в сводной таблице после данных каждого элемента (включая промежуточные суммы) вставляется пустая строка.

```
5219 <define name="table-data-pilot-layout-info-attlist" combine="interleave">
5220     <attribute name="table:add-empty-lines">
5221         <ref name="boolean"/>
5222     </attribute>
5223 </define>
```

8.8.13 Ссылки полей сводной таблицы

Элемент `<table:data-pilot-field-reference>` описывает данные, которые могут быть использованы для модификации отображаемых значений полей данных.

```
<define name="table-data-pilot-field-reference">
    <element name="table:data-pilot-field-reference">
        <ref name="table-data-pilot-field-reference-attlist"/>
    </element>
</define>
```

Поле ссылки

Атрибут `table:field-name` ссылается на поле категории, элементы которого влияют на отображаемые значения поля данных, и является частью `<table:data-pilot-field-reference>`.

```
<define name="table-data-pilot-field-reference-attlist" combine="interleave">
    <attribute name="table:field-name">
        <ref name="string"/>
    </attribute>
</define>
```

Базовый элемент

Атрибут `table:member-type` определяет базовый элемент поля категории, от значения которого зависит значение текущего поля данных. Если его значение равно `next` (`previous`), используется значение поля данных для следующего (предыдущего) видимого элемента указываемого поля категории. Если его значение равно `named`, то элемент, от значения которого зависит поле данных, определяется атрибутом `table:member-name`.

При значениях `previous` и `next` пустые элементы пропускаются.

```
<define name="table-data-pilot-field-reference-attlist" combine="interleave">
    <choice>
        <group>
            <attribute name="table:member-type">
                <value>named</value>
            </attribute>
            <attribute name="table:member-name">
                <ref name="string"/>
            </attribute>
        </group>
        <attribute name="table:member-type">
            <choice>
                <value>previous</value>
                <value>next</value>
            </choice>
        </attribute>
```

```
5250     </choice>
5251 </define>
```

Тип ссылки

Атрибут table:type определяет, каким образом базовое поле категории влияет на отображаемое значение поля данных. Он может принимать одно из следующих значений:

- none (обычный)

  Это значение устанавливает, что результаты в полях данных отображаются без изменения.

- member-difference (разность)

  Из каждого результата вычитается значение, базового элемента поля категории, определенного атрибутами member-type и table:member-name.

- member-percentage (доля)

  Каждый результат делится на значение, вычисленное для элемента поля категории, определенного атрибутами table:member-type и table:member-name. Деление на ноль приводит к ошибке. Отсутствующие результаты отображаются как «0». Если атрибут table:member-type имеет значение previous, в качестве первого значения отображается «1». Если атрибут table:member-type имеет значение next, в качестве последнего значения отображается «1».

- member-percentage-difference (приведенная разность)

  Из каждого результата вычитается значение, вычисленное для элемента поля категории, определенное атрибутами table:member-type и table:member-name, а результат делится на это же значение. Деление на ноль приводит к ошибке. В других случаях применяются правила member-difference.

- running-total (нарастающий итог)

  Каждый результат добавляется к сумме результатов предыдущих элементов в базовом поле категории, в порядке его сортировки, и отображается итоговая сумма.

- row-percentage (доля по строке)

Каждый результат делится на общий результат по строке сводной таблицы. При наличии нескольких полей данных используется общий итог по полю данных результатов. Если имеются промежуточные суммы с функциями итога, выбранными вручную, общий итог вычисляется с использованием функции итогов поля данных. Деление на ноль приводит к ошибке.

- `column-percentage` (доля по столбцу)

  То же, что и `row-percentage`, но используется итог по столбцу результатов.

- `total-percentage` (доля по итогу)

  То же, что и `row-percentage`, но используется общий итог по полю данных результатов.

- `index` (индекс)

  На основании сумм по столбцу и строке, а также общей суммы по приведенным выше правилам вычисляется выражение: (первоначальный результат * общая сумма) / (сумма по строке * сумма по столбцу). Деление на ноль приводит к ошибке.

```
5252 <define name="table-data-pilot-field-reference-attlist" combine="interleave">
5253     <attribute name="table:type">
5254         <choice>
5255             <value>none</value>
5256             <value>member-difference</value>
5257             <value>member-percentage</value>
5258             <value>member-percentage-difference</value>
5259             <value>running-total</value>
5260             <value>row-percentage</value>
5261             <value>column-percentage</value>
5262             <value>total-percentage</value>
5263             <value>index</value>
5264         </choice>
5265     </attribute>
5266 </define>
```

8.8.14 Группы в сводных таблицах

Элемент `<table:data-pilot-groups>` определяет, что поле сводной таблицы является полем группировки. Поле группировки позволяет группировать другие поля. Например, если сводная таблица содержит поле столбца с именем «город», который имеет элементы «Берлин», «Мюнхен», «Франкфурт», «Гамбург», «Лондон», «Манчестер», «Хейстингс» и «Ливерпуль», и необходимо группировать города по странам, для этого в сводную таблицу может быть добавлено поле

группировки с именем «город2», которое содержит две группы, называемые «Англия» и «Германия». Каждая группа содержит список имен ее элементов. В данном примере группа «Англия» будет содержать «Лондон», «Манчестер», «Хейстингс» и «Ливерпуль». Группа «Германия» содержит «Берлин», «Мюнхен», «Франкфурт» и «Гамбург».

Группировка может быть применена к числам и датам.

```
<define name="table-data-pilot-groups">
    <element name="table:data-pilot-groups">
        <ref name="table-data-pilot-groups-attlist"/>
        <oneOrMore>
            <ref name="table-data-pilot-group"/>
        </oneOrMore>
    </element>
</define>
```

Имя поля источника

Атрибут `table:source-field-name` ссылается на поле, содержащее данные, подлежащие группировке, если эти данные не являются данными, представленными самим полем.

```
<define name="table-data-pilot-groups-attlist" combine="interleave">
    <attribute name="table:source-field-name">
        <ref name="string"/>
    </attribute>
</define>
```

Начало

Если группируются числа или даты, атрибуты `table:date-start` и `table:start` определяют начальное значение для группировки. Все значения, меньшие начального значения, содержатся в одной группе, в то время как значения, большие или равные начальному значению, группируются, как определено атрибутами `table:grouped-by` и `table:step`.

Если значение атрибута равно `auto`, в качестве начального выбирается минимальное значение поля.

```
<define name="table-data-pilot-groups-attlist" combine="interleave">
    <choice>
        <attribute name="table:date-start">
            <choice>
                <ref name="dateOrDateTime"/>
                <value>auto</value>
            </choice>
        </attribute>
        <attribute name="table:start">
            <choice>
```

```
                <ref name="double"/>
                <value>auto</value>
            </choice>
        </attribute>
    </choice>
</define>
```

Конец

Если группируются числа или даты, атрибуты `table:date-end` и `table:end` определяют конечное значение для группировки. Все значения, большие конечного значения, содержатся в одной группе, в то время как значения, меньшие или равные конечному значению, группируются, как определено атрибутами `table:grouped-by` и `table:step`.

Если значение атрибута равно `auto`, в качестве конечного выбирается максимальное значение поля.

```
<define name="table-data-pilot-groups-attlist" combine="interleave">
    <choice>
        <attribute name="table:date-end">
            <choice>
                <ref name="dateOrDateTime"/>
                <value>auto</value>
            </choice>
        </attribute>
        <attribute name="table:end">
            <choice>
                <ref name="double"/>
                <value>auto</value>
            </choice>
        </attribute>
    </choice>
</define>
```

Шаг

Атрибут `table:step` определяет группировку числовых значений, задавая интервал между группами. Например, если атрибут группировки `table:start` имеет значение 5, а атрибут `table:step` имеет значение 2, все значения, равные или большие 5, но меньшие 7, будут в одной группе. Все значения, равные или большие 7, но меньшие 9, будут в следующей группе и так далее, пока не будет достигнуто конечное значение.

```
<define name="table-data-pilot-groups-attlist" combine="interleave">
    <attribute name="table:step">
        <ref name="double"/>
    </attribute>
</define>
```

Группировка по

Атрибут `table:grouped-by` определяет группировку дат. Даты могут группироваться по секундам, минутам, часам, дням, месяцам, кварталам или годам. Если даты сгруппированы, например, по минутам, все даты и времена, которые имею одинаковые значения минут попадут в одну группу. Так, если даны даты `2004-08-27T12:34:46`, `2004-08-27T12:34:56` и `2004-08-27T12:35:46`, две первых попадут в одну группу, в то время как последняя будет единственной датой в собственной группе.

```
<define name="table-data-pilot-groups-attlist" combine="interleave">
    <attribute name="table:grouped-by">
        <choice>
            <value>seconds</value>
            <value>minutes</value>
            <value>hours</value>
            <value>days</value>
            <value>months</value>
            <value>quarters</value>
            <value>years</value>
        </choice>
    </attribute>
</define>
```

8.8.15 Группа сводной таблицы

Если группировка осуществляется по определенным именам элементов, то имена элементов одной группы определяются элементом `<table:data-pilot-group>`.

```
<define name="table-data-pilot-group">
    <element name="table:data-pilot-group">
        <ref name="table-data-pilot-group-attlist"/>
        <oneOrMore>
            <ref name="table-data-pilot-group-member"/>
        </oneOrMore>
    </element>
</define>
```

Имя

Атрибут `table:name` определяет имя группы.

```
<define name="table-data-pilot-group-attlist" combine="interleave">
    <attribute name="table:name">
        <ref name="string"/>
    </attribute>
</define>
```

8.8.16 Элемент группы сводной таблицы

Элемент `<table:data-pilot-group-member>` определяет имя одного элемента группы.

```
<define name="table-data-pilot-group-member">
    <element name="table:data-pilot-group-member">
        <ref name="table-data-pilot-group-member-attlist"/>
    </element>
</define>
```

Имя

Атрибут `table:name` определяет имя элемента.

```
<define name="table-data-pilot-group-member-attlist" combine="interleave">
    <attribute name="table:name">
        <ref name="string"/>
    </attribute>
</define>
```

8.9 Консолидация

Консолидация — сбор данных из нескольких независимых диапазонов таблиц. Новый диапазон таблицы вычисляется с применением математических функций ко всем ячейкам в исходных диапазонах таблиц, имеющим одинаковый относительный адрес внутри своих диапазонов. Консолидация определяется элементом `<table:consolidation>`.

```
<define name="table-consolidation">
    <element name="table:consolidation">
        <ref name="table-consolidation-attlist"/>
        <empty/>
    </element>
</define>
```

С данным элементом могут быть связаны следующие атрибуты:

- функция;
- адреса диапазона ячеек источника;
- адрес целевой ячейки;
- использование надписи;
- связь с исходными данными.

Функция

Атрибут table:function содержит функцию, которая используется для консолидации данных. Возможные функции: auto, average, count, countnums, max, min, product, stdev, stdevp, sum, var и varp.

```
<define name="table-consolidation-attlist" combine="interleave">
    <attribute name="table:function">
        <choice>
            <value>auto</value>
            <value>average</value>
            <value>count</value>
            <value>countnums</value>
            <value>max</value>
            <value>min</value>
            <value>product</value>
            <value>stdev</value>
            <value>stdevp</value>
            <value>sum</value>
            <value>var</value>
            <value>varp</value>
            <ref name="string"/>
        </choice>
    </attribute>
</define>
```

Адреса диапазона ячеек источника

Атрибут table:source-cell-range-addresses содержит список адресов диапазонов ячеек, определяющих исходные диапазоны данных.

```
<define name="table-consolidation-attlist" combine="interleave">
    <attribute name="table:source-cell-range-addresses">
        <ref name="cellRangeAddressList"/>
    </attribute>
</define>
```

Адрес целевой ячейки

Атрибут table:target-cell-address содержит адрес целевой ячейки.

```
<define name="table-consolidation-attlist" combine="interleave">
    <attribute name="table:target-cell-address">
        <ref name="cellAddress"/>
    </attribute>
</define>
```

Использование надписи

Атрибут table:use-label определяет, должны ли использоваться надписи при консолидации строк, столбцов или и тех, и других. Возможные значения: none, column, row и both. Если для строк и столбцов используются надписи,

математическая функция применяется к ячейкам с одинаковыми надписями строк или столбцов вместо ячеек с одинаковыми относительными адресами.

```
<define name="table-consolidation-attlist" combine="interleave">
    <optional>
        <attribute name="table:use-labels" a:defaultValue="none">
            <choice>
                <value>none</value>
                <value>row</value>
                <value>column</value>
                <value>both</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Связь с исходными данными

Атрибут `table:link-to-source-data` определяет, должны ли данные в диапазоне консолидированной таблицы быть связанными с исходными данными, т. е. автоматически обновляться, если произошли изменения в исходных данных.

```
<define name="table-consolidation-attlist" combine="interleave">
    <optional>
        <attribute name="table:link-to-source-data" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.10 DDE-ссылки

Элемент-контейнер `<table:dde-links>` содержит все DDE-ссылки документа электронной таблицы. Каждая связь содержит источник DDE и данные последнего соединения. Детальную информация приведена в разделе 12.6.3.

```
<define name="table-dde-links">
    <element name="table:dde-links">
        <oneOrMore>
            <ref name="table-dde-link"/>
        </oneOrMore>
    </element>
</define>
```

8.11 Отслеживание изменений в электронных таблицах

В документах электронных таблиц изменения могут отслеживаться. В данном разделе описано, каким образом представляется информация отслеживания изменений.

В текстовых документах отслеживание изменений в таблицах не поддерживается.

8.11.1 Отслеживаемые изменения

Все изменения, внесенные в документ электронной таблицы, сохраняются в списке. Для каждого изменения, произошедшего в документе, в списке содержится элемент. Для отслеживания изменений в документе электронной таблицы должен быть представлен элемент `<table:tracked-changes>`.

```
<define name="table-tracked-changes">
    <element name="table:tracked-changes">
        <ref name="table-tracked-changes-attlist"/>
        <zeroOrMore>
            <choice>
                <ref name="table-cell-content-change"/>
                <ref name="table-insertion"/>
                <ref name="table-deletion"/>
                <ref name="table-movement"/>
            </choice>
        </zeroOrMore>
    </element>
</define>
```

Отслеживание изменений

Атрибут `table:track-changes` определяет, разрешено ли отслеживание изменений.

```
<define name="table-tracked-changes-attlist" combine="interleave">
    <optional>
        <attribute name="table:track-changes" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

8.11.2 Вставка

Элемент `<table:insertion>` содержит информацию, необходимую для обнаружения любой вставки содержимого. Таким содержимым может быть одна строка или более строк, один столбец или более столбцов, или таблица.

```
<define name="table-insertion">
    <element name="table:insertion">
        <ref name="table-insertion-attlist"/>
        <ref name="common-table-change-attlist"/>
        <ref name="office-change-info"/>
        <optional>
            <ref name="table-dependencies"/>
        </optional>
```

```
        <optional>
            <ref name="table-deletions"/>
        </optional>
    </element>
</define>
```

С данным элементом могут быть связаны следующие атрибуты:

- идентификатор (см. раздел 8.11.18);
- состояние принятия (см. раздел 8.11.18);
- идентификатор отклоняемого изменения (см. раздел 8.11.18);
- тип;
- позиция;
- количество;
- таблица.

Тип

Атрибут `table:type` определяет тип вставки. Он может принимать значения `row`, `column` или `table`.

```
<define name="table-insertion-attlist" combine="interleave">
    <attribute name="table:type">
        <choice>
            <value>row</value>
            <value>column</value>
            <value>table</value>
        </choice>
    </attribute>
</define>
```

Позиция

Атрибут `table:position` определяет позицию, в которой была осуществлена вставка. В зависимости от типа вставки он может принимать значения номера строки, столбца или таблицы.

```
<define name="table-insertion-attlist" combine="interleave">
    <attribute name="table:position">
        <ref name="integer"/>
    </attribute>
</define>
```

Количество

Атрибут `table:count` определяет количество вставленных строк, столбцов или таблиц.

```
<define name="table-insertion-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="table:count" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Таблица

Атрибут `table:table` определяет номер таблицы, в которой осуществлена вставка. Этот атрибут существует для вставок столбца или строки.

```
<define name="table-insertion-attlist" combine="interleave">
    <optional>
        <attribute name="table:table">
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

Пример – Вставка текста в ячейку:

```
<table:tracked-changes>
    <table:insertion table:id="c001" table:acceptance-state="pending"
                     table:type="column" table:position="5">
        <office:change-info>
            <dc:creator>Sascha Ballach</dc:creator>
            <dc:date>1999-55-18T12:56:04</dc:date>
        </office:change-info>
    </table:insertion>
</table:tracked-changes>
```

8.11.3 Зависимости

Элемент `<table:dependencies>` содержит информацию о других отслеживаемых изменениях, от которых данное отслеживаемое изменение зависит. Каждый элемент отслеживаемых изменений может содержать элемент `<table:dependencies>`.

```
<define name="table-dependencies">
    <element name="table:dependencies">
        <oneOrMore>
            <ref name="table-dependency"/>
        </oneOrMore>
    </element>
</define>
```

8.11.4 Зависимость

Элемент `<table:dependency>` содержит информацию об одном действии изменения, от которого зависит родительский элемент. Действие изменения определяется по его идентификатору.

```
<define name="table-dependency">
    <element name="table:dependency">
        <attribute name="table:id">
            <ref name="string"/>
        </attribute>
        <empty/>
    </element>
</define>
```

8.11.5 Удаления

Элемент `<table:deletions>` содержит все удаления, произошедшие во время отслеживания одного изменения таблицы.

```
<define name="table-deletions">
    <element name="table:deletions">
        <oneOrMore>
            <choice>
                <ref name="table-cell-content-deletion"/>
                <ref name="table-change-deletion"/>
            </choice>
        </oneOrMore>
    </element>
</define>
```

8.11.6 Удаление содержимого ячейки

Элемент `<table:cell-content-deletion>` определяет, что содержимое ячейки было удалено. Он содержит адрес обработанной ячейки и ее бывшее содержимое. Если представлен атрибут `text:id`, он определяет идентификатор предыдущего отслеженного изменения, которое было удалено текущим изменением.

```
<define name="table-cell-content-deletion">
    <element name="table:cell-content-deletion">
        <optional>
            <attribute name="table:id">
                <ref name="string"/>
            </attribute>
        </optional>
        <optional>
            <ref name="table-cell-address"/>
        </optional>
        <optional>
            <ref name="table-change-track-table-cell"/>
        </optional>
    </element>
</define>
```

8.11.7 Удаление изменений

Элемент `<table:change-deletion>` определяет предыдущее отслеженное изменение, которое было удалено текущим изменением.

```
<define name="table-change-deletion">
    <element name="table:change-deletion">
        <optional>
            <attribute name="table:id">
                <ref name="string"/>
            </attribute>
        </optional>
        <empty/>
    </element>
</define>
```

8.11.8 Удаление

Элемент `<table:deletion>` хранит содержимое, которое было удалено, пока отслеживание изменений было разрешено. Удаленное содержимое ячеек хранится в элементе `<table:dependencies>` или `<table:deletions>`.

```
<define name="table-deletion">
    <element name="table:deletion">
        <ref name="table-deletion-attlist"/>
        <ref name="common-table-change-attlist"/>
        <ref name="office-change-info"/>
        <optional>
            <ref name="table-dependencies"/>
        </optional>
        <optional>
            <ref name="table-deletions"/>
        </optional>
        <optional>
            <ref name="table-cut-offs"/>
        </optional>
    </element>
</define>
```

С данным элементом могут быть связаны следующие атрибуты:

- идентификатор (см. раздел 8.11.18);
- состояние принятия (см. раздел 8.11.18);
- идентификатор отклоняемого изменения (см. раздел 8.11.18);
- тип;
- позиция;
- таблица;
- покрытие множественных удалений.

Тип

Атрибут `table:type` определяет тип удаления. Он может принимать значения `row`, `column` или `table`.

```
<define name="table-deletion-attlist" combine="interleave">
    <attribute name="table:type">
```

```
        <choice>
            <value>row</value>
            <value>column</value>
            <value>table</value>
        </choice>
    </attribute>
</define>
```

Позиция

Атрибут `table:position` определяет позицию, в которой было осуществлено удаление. В зависимости от типа удаления он может принимать значения номера строки, столбца или таблицы.

```
<define name="table-deletion-attlist" combine="interleave">
    <attribute name="table:position">
        <ref name="integer"/>
    </attribute>
</define>
```

Таблица

Атрибут `table:table` определяет номер таблицы, в которой осуществлено удаление. Этот атрибут существует для удалений столбца или строки.

```
<define name="table-deletion-attlist" combine="interleave">
    <optional>
        <attribute name="table:table">
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

Покрытие множественных удалений

Если одновременно было удалено множество столбцов или строк, каждая удаленная строка или столбец получают свой собственный элемент `<table:deletion>`. Элемент удаленной первой строки или столбца в этом случае несет в себе атрибут `table:multi-deletion-spanned`, который определяет общее количество удаленных строк или столбцов.

```
<define name="table-deletion-attlist" combine="interleave">
    <optional>
        <attribute name="table:multi-deletion-spanned">
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

8.11.9 Вырезки

Элемент `<table:cut-offs>` содержит информацию о ранее отслеженных вставках или перемещениях, часть содержимого которых была удалена данной операцией. Например, это может быть перемещенный ранее диапазон ячеек, который перекрывался только что удаленной строкой.

```
<define name="table-cut-offs">
    <element name="table:cut-offs">
        <choice>
            <oneOrMore>
                <ref name="table-movement-cut-off"/>
            </oneOrMore>
            <group>
                <ref name="table-insertion-cut-off"/>
                <zeroOrMore>
                    <ref name="table-movement-cut-off"/>
                </zeroOrMore>
            </group>
        </choice>
    </element>
</define>
```

8.11.10 Вырезка вставки

Элемент `<table:insertion-cut-off>` содержит информацию о том, где произошло удаление вставки и что было удалено.

```
<define name="table-insertion-cut-off">
    <element name="table:insertion-cut-off">
        <ref name="table-insertion-cut-off-attlist"/>
        <empty/>
    </element>
</define>
```

С данным элементом могут быть связаны следующие атрибуты:

- идентификатор (см. раздел 8.11.18);
- позиция.

Идентификатор

Атрибут `table:id` содержит идентификатор вставки, часть которой была удалена.

```
<define name="table-insertion-cut-off-attlist" combine="interleave">
    <attribute name="table:id">
        <ref name="string"/>
    </attribute>
</define>
```

Позиция

Атрибут table:position определяет количество строк или столбцов, которые были удалены внутри вставки.

```
<define name="table-insertion-cut-off-attlist" combine="interleave">
    <attribute name="table:position">
        <ref name="integer"/>
    </attribute>
</define>
```

8.11.11 Вырезка перемещения

Элемент <table:movement-cut-off> содержит информацию о том, где произошло удаление перемещения и что было удалено.

```
<define name="table-movement-cut-off">
    <element name="table:movement-cut-off">
        <ref name="table-movement-cut-off-attlist"/>
        <empty/>
    </element>
</define>
```

С данным элементом могут быть связаны следующие атрибуты:

- идентификатор (см. раздел 8.11.18);
- начальная позиция, конечная позиция, позиция.

Начальная позиция, конечная позиция, позиция

Атрибуты table:start-position, table:end-position и table:position определяют позицию внутри перемещения, которая была удалена. Если удалена одна строка или столбец, то в атрибуте table:position содержится ее номер. Если были удалены несколько строк или столбцов, атрибуты table:start-position и table:end-position содержат номер первой (включительно) и последней (не включительно) удаленных строк или столбцов.

```
<define name="table-movement-cut-off-attlist" combine="interleave">
    <choice>
        <attribute name="table:position">
            <ref name="integer"/>
        </attribute>
        <group>
            <attribute name="table:start-position">
                <ref name="integer"/>
            </attribute>
            <attribute name="table:end-position">
                <ref name="integer"/>
            </attribute>
```

```
        </group>
    </choice>
</define>
```

Пример – Удаление столбца, не имеющего содержимого:

```
<table:tracked-changes>
    <table:deletion table:id="c002" table:acceptance-state="pending"
                    table:type="column" table:position="9">
        <office:change-info>
            <dc:creator>Sascha Ballach</dc:creator>
            <dc:date>1999-05-18T12:56:04</dc:creator>
        </office:change-info>
    </table:deletion>
</table:tracked-changes>
```

8.11.12 Перемещение

Элемент `<table:movement>` содержит информацию, необходимую для распознавания любого перемещения содержимого. Таким содержимым может быть содержимое ячейки или содержимое диапазона ячеек.

```
<define name="table-movement">
    <element name="table:movement">
        <ref name="common-table-change-attlist"/>
        <ref name="table-source-range-address"/>
        <ref name="table-target-range-address"/>
        <ref name="office-change-info"/>
        <optional>
            <ref name="table-dependencies"/>
        </optional>
        <optional>
            <ref name="table-deletions"/>
        </optional>
    </element>
</define>
```

С данным элементом могут быть связаны следующие атрибуты:

- идентификатор (см. раздел 8.11.18);
- состояние принятия (см. раздел 8.11.18);
- идентификатор отклоняемого изменения (см. раздел 8.11.18).

8.11.13 Адрес целевого диапазона, адрес диапазона источника

Элементы `<table:source-range-address>` и `<table:target-range-address>` определяют адреса исходной и целевой ячеек или адреса диапазонов ячеек для перемещения.

```
<define name="table-source-range-address">
    <element name="table:source-range-address">
```

```
5637        <ref name="common-table-range-attlist"/>
5638        <empty/>
5639    </element>
5640 </define>
5641
5642 <define name="table-target-range-address">
5643    <element name="table:target-range-address">
5644        <ref name="common-table-range-attlist"/>
5645        <empty/>
5646    </element>
5647 </define>
5648
5649
5650 <define name="common-table-range-attlist" combine="interleave">
5651    <choice>
5652        <group>
5653            <ref name="common-table-cell-address-attlist"/>
5654        </group>
5655        <group>
5656            <ref name="common-table-cell-range-address-attlist"/>
5657        </group>
5658    </choice>
5659 </define>
```

С обоими элементами могут быть связаны следующие атрибуты:

- столбец, строка и таблица;
- начальный столбец, конечный столбец, начальная строка, конечная строка, начальная таблица, конечная таблица.

Столбец, строка и таблица

Если адрес диапазона — это адрес ячейки, то три атрибута `table:column`, `table:row` и `table:table` определяют номера столбца, строки и таблицы ячейки.

```
5660 <define name="common-table-cell-address-attlist" combine="interleave">
5661    <attribute name="table:column">
5662        <ref name="integer"/>
5663    </attribute>
5664    <attribute name="table:row">
5665        <ref name="integer"/>
5666    </attribute>
5667    <attribute name="table:table">
5668        <ref name="integer"/>
5669    </attribute>
5670 </define>
```

Начальный столбец, конечный столбец, начальная строка, конечная строка, начальная таблица, конечная таблица

Если адрес является адресом диапазона, а не адресом ячейки, атрибуты `table:start-column`, `table:end-column`, `table:start-row`, `table:end-row`, `table:start-table` и `table:end-table` определяют начальные и конечные

столбцы, строки и таблицы диапазона. И начальные, и конечные номера указываются включительно.

```
<define name="common-table-cell-range-address-attlist" combine="interleave">
    <attribute name="table:start-column">
        <ref name="integer"/>
    </attribute>
    <attribute name="table:start-row">
        <ref name="integer"/>
    </attribute>
    <attribute name="table:start-table">
        <ref name="integer"/>
    </attribute>
    <attribute name="table:end-column">
        <ref name="integer"/>
    </attribute>
    <attribute name="table:end-row">
        <ref name="integer"/>
    </attribute>
    <attribute name="table:end-table">
        <ref name="integer"/>
    </attribute>
</define>
```

Пример – Перемещение ячейки:

```
<table:tracked-changes>
    <table:movement table:id="ct1">
        <table:source-range-address table:column="0" table:row="0"
                                    table:table="0"/>
        <table:target-range-address table:column="1" table:row="1"
                                    table:table="0"/>
        <office:change-info>
            <dc:creator>Michael Brauer</dc:creator>
            <dc:date>2003-12-29T11:46:13,21"</dc:date>
        </office:change-info>
    </table:movement>
</table:tracked-changes>
```

8.11.14 Ячейка отслеживания изменений

Элемент `<table:change-track-table-cell>` содержит всю информацию ячейки таблицы, которая необходима внутри элементов отслеживания изменений. Элемент очень похож на элемент `<table:table-cell>`, но содержит некоторую дополнительную информацию.

```
<define name="table-change-track-table-cell" combine="interleave">
    <element name="table:change-track-table-cell">
        <ref name="table-change-track-table-cell-attlist"/>
        <zeroOrMore>
            <ref name="text-p"/>
        </zeroOrMore>
    </element>
</define>
```

Адрес ячейки

Если ячейка содержит формулу, необходим атрибут `table:cell-address`, определяющий исходный адрес ячейки, используемой в вычислениях.

```
<define name="table-change-track-table-cell-attlist" combine="interleave">
    <optional>
        <attribute name="table:cell-address">
            <ref name="cellAddress"/>
        </attribute>
    </optional>
</define>
```

Покрытие матрицы

Если ячейка является ячейкой матрицы, но не базой матрицы, то необходим атрибут `table:matrix-covered`, значение которого должно быть `true` для обозначения того, что ячейка содержится в матрице.

```
<define name="table-change-track-table-cell-attlist" combine="interleave">
    <optional>
        <attribute name="table:matrix-covered" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Формулы и значения

Ячейки таблицы отслеживания изменений поддерживают атрибуты `table:formula`, `table:number-matrix-rows-spanned`, `table:number-matrix-columns-spanned`, `office:value-type`, `office:value`, `office:date-value`, `office:time-value` и `office:string-value`, описанные в разделе 8.1.3.

```
<define name="table-change-track-table-cell-attlist" combine="interleave">
    <optional>
        <attribute name="table:formula">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="table:number-matrix-columns-spanned">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="table:number-matrix-rows-spanned">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
    <optional>
        <ref name="common-value-and-type-attlist"/>
```

```
    </optional>
</define>
```

8.11.15 Изменение содержимого ячейки

Элемент `<table:cell-content-change>` содержит информацию, необходимую для распознавания изменений содержимого ячейки.

```
<define name="table-cell-content-change">
    <element name="table:cell-content-change">
        <ref name="common-table-change-attlist"/>
        <ref name="table-cell-address"/>
        <ref name="office-change-info"/>
        <optional>
            <ref name="table-dependencies"/>
        </optional>
        <optional>
            <ref name="table-deletions"/>
        </optional>
        <ref name="table-previous"/>
    </element>
</define>
```

С данным элементом могут быть связаны следующие атрибуты:

- идентификатор (см. раздел 8.11.18);
- состояние принятия (см. раздел 8.11.18);
- идентификатор отклоняемого изменения (см. раздел 8.11.18).

8.11.16 Адрес ячейки

Элемент `<table:cell-address>` содержит адрес ячейки, которая была изменена. В отличие от других адресов ячеек, этот адрес содержит номера строки, столбца и таблицы ячейки. Это позволяет определять адреса, выходящие за пределы действительного диапазона адресов ячеек, например, имеющие отрицательный номер столбца.

```
<define name="table-cell-address">
    <element name="table:cell-address">
        <ref name="common-table-cell-address-attlist"/>
        <empty/>
    </element>
</define>
```

С данным элементом могут быть связаны следующие атрибуты:

- номер столбца, строки и таблицы (см. раздел 8.11.13).

8.11.17 Предыдущее содержимое ячейки

Элемент `table:previous` содержит предыдущее содержимое ячейки, которое перезаписано текущим изменением. Если атрибут `text:id` представлен, он определяет ID ранее отслеженного изменения для ячейки, которая изменена снова текущим изменением.

```
<define name="table-previous">
    <element name="table:previous">
        <optional>
            <attribute name="table:id">
                <ref name="string"/>
            </attribute>
        </optional>
        <ref name="table-change-track-table-cell"/>
    </element>
</define>
```

8.11.18 Общие атрибуты отслеживания изменений

Идентификатор

Атрибут `table:id` определяет идентификационный номер отслеженного изменения.

```
<define name="common-table-change-attlist" combine="interleave">
    <attribute name="table:id">
        <ref name="string"/>
    </attribute>
</define>
```

Состояние принятия

Атрибут `table:acceptance-state` определяет, принято или отклонено отслеженное изменение, или находится в состоянии ожидания принятия или отклонения.

```
<define name="common-table-change-attlist" combine="interleave">
    <optional>
        <attribute name="table:acceptance-state" a:defaultValue="pending">
            <choice>
                <value>accepted</value>
                <value>rejected</value>
                <value>pending</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Идентификатор отклоняемого изменения

Если атрибут `table:rejecting-change-id` представлен, текущее изменение должно быть выполнено как отмена другого ранее отслеженного изменения. Значением атрибута является идентификатор ранее отслеженного изменения, которое должно быть отклонено.

```
<define name="common-table-change-attlist" combine="interleave">
    <optional>
        <attribute name="table:rejecting-change-id">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

9 Графическое содержимое

В этом разделе предоставлена спецификация для базовых элементов графических приложений, таких как приложения для рисования и подготовки презентаций, и для графических объектов, содержащихся в неграфических приложениях, таких как текстовые процессоры и электронные таблицы.

9.1 Расширенные свойства страниц для графических приложений

9.1.1 Мастер-страница тезисов

Для приложений, которые поддерживают печать тезисов, этот элемент является шаблоном для автоматической генерации страниц тезисов. Элемент `<style:handout-master>` может содержать любые типы векторных фигур. Наиболее полезная векторная фигура — это `<draw:page-thumbnail>`, которая заменяется текущими страницами из документа. Элемент `<style:handout-master>` содержится в элементе `<office:master-styles>`. Элемент `<office:master-styles>` не должен содержать более одного элемента `<style:handout-master>`.

```
<define name="style-handout-master">
    <element name="style:handout-master">
        <ref name="common-presentation-header-footer-attlist"/>
        <ref name="style-handout-master-attlist"/>
        <zeroOrMore>
            <ref name="shape"/>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<style:handout-master>` могут быть связаны следующие атрибуты:

- разметка страницы презентации (объекты заполнения);
- разметка страницы (размер страницы, поля и т. д.);
- стиль страницы;
- объявление верхнего колонтитула;
- объявление нижнего колонтитула;

- объявление даты и времени.

Разметка страницы презентации

Атрибут `presentation:presentation-page-layout-name` ссылается на элемент `<style:presentation-page-layout>` (см. раздел 14.15 для получения информации об элементе разметки страницы презентации). Это необязательный атрибут.

```
<define name="style-handout-master-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:presentation-page-layout-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Разметка страницы

Атрибут `style:page-layout-name` определяет разметку страницы, которая состоит из размеров, обрамления и ориентации страницы мастер-страницы тезисов (см. раздел 14.3 для подробностей о разметке страницы).

```
<define name="style-handout-master-attlist" combine="interleave">
    <attribute name="style:page-layout-name">
        <ref name="styleNameRef"/>
    </attribute>
</define>
```

Стиль страницы

Атрибут `draw:style-name` назначает дополнительные атрибуты форматирования для мастер-страницы тезисов путем присваивания стиля графической страницы. Это необязательный атрибут. Фиксированным семейством для стилей страницы является `drawing-page`.

```
<define name="style-handout-master-attlist" combine="interleave">
    <optional>
        <attribute name="draw:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Объявление верхнего колонтитула

Атрибут `presentation:use-header-name` определяет имя объявления поля верхнего колонтитула (см. раздел 9.11.2), которое используется для всех

полей верхнего колонтитула (см. раздел 9.10.1), отображаемых на мастер-странице тезисов (см. также раздел 9.1.4).

Объявление нижнего колонтитула

Атрибут `presentation:use-footer-name` определяет имя объявления поля нижнего колонтитула (см. раздел 9.11.3), которое используется для всех полей нижнего колонтитула (см. раздел 9.10.2), отображаемых на мастер-странице тезисов (см. также раздел 9.1.4).

Объявление даты и времени

Атрибут `presentation:use-date-time-name` определяет имя объявления поля даты и времени (см. раздел 9.11.4), которое используется для всех полей даты и времени (см. раздел 9.10.3), отображаемых на мастер-странице тезисов (см. также раздел 9.1.4).

9.1.2 Множества слоев

Элемент `<draw:layer-set>` может содержаться в мастер-стилях графического приложения. Он определяет множество слоев. Слои группируют графические объекты. Графические объекты могут быть отнесены к этим слоям с помощью атрибута `draw:layer-name`.

```
<define name="draw-layer-set">
    <element name="draw:layer-set">
        <zeroOrMore>
            <ref name="draw-layer"/>
        </zeroOrMore>
    </element>
</define>
```

9.1.3 Слой

Элемент `<draw:layer>` определяет одиночный слой.

```
<define name="draw-layer">
    <element name="draw:layer">
        <ref name="draw-layer-attlist"/>
        <empty/>
    </element>
</define>
```

Имя

Каждый элемент `<draw:layer>` определен и на него можно сослаться по его имени, которое содержится в атрибуте `draw:name`. Каждый графический объект внутри графического документа или документа презентации может быть отнесен к слою. Слои виртуально группируют объекты. Каждый объект, отнесенный к слою, наследует настройки слоя.

```
<define name="draw-layer-attlist" combine="interleave">
    <attribute name="draw:name">
        <ref name="string"/>
    </attribute>
</define>
```

Защита

Атрибут `draw:protected` определяет, защищен ли от модификации графический объект, содержащийся в слое.

```
<define name="draw-layer-attlist" combine="interleave">
    <optional>
        <attribute name="draw:protected" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Отображение

Атрибут `draw:display` определяет, будет ли графический объект, содержащийся в слое, видимым на экране и/или при печати.

```
<define name="draw-layer-attlist" combine="interleave">
    <optional>
        <attribute name="draw:display" a:defaultValue="always">
            <choice>
                <value>always</value>
                <value>screen</value>
                <value>printer</value>
                <value>none</value>
            </choice>
        </attribute>
    </optional>
</define>
```

9.1.4 Графические страницы

Элемент `<draw:page>` является контейнером для содержимого в графическом документе или документе презентации. Графические страницы используются для следующего:

- формы (см. раздел 11.1);
- графические изображения (см. раздел 9.2);
- фреймы (см. раздел 9.3);
- анимация презентации (см. раздел 9.7);
- заметки презентации (см. раздел 9.1.5).

Мастер-страница должна быть назначена для каждой графической страницы.

```
<define name="draw-page">
    <element name="draw:page">
        <ref name="common-presentation-header-footer-attlist"/>
        <ref name="draw-page-attlist"/>
        <optional>
            <ref name="office-forms"/>
        </optional>
        <zeroOrMore>
            <ref name="shape"/>
        </zeroOrMore>
        <optional>
            <choice>
                <ref name="presentation-animations"/>
                <ref name="animation-element"/>
            </choice>
        </optional>
        <optional>
            <ref name="presentation-notes"/>
        </optional>
    </element>
</define>
```

С элементом `<draw:page>` могут быть связаны следующие атрибуты:

- имя страницы;
- стиль страницы;
- мастер-страница;
- разметка страницы презентации;
- объявление верхнего колонтитула;
- объявление нижнего колонтитула;
- объявление даты и времени;
- идентификатор.

Элементы, которые могут быть включены в элемент `<draw:page>`:

- формы;
- векторные фигуры;
- анимация;
- заметки презентации.

Имя страницы

Атрибут `draw:name` определяет имя графической страницы. Это необязательный атрибут; если он используется, имя должно быть уникальным. Если он не используется, приложение может генерировать уникальное имя.

```
<define name="draw-page-attlist" combine="interleave">
    <optional>
        <attribute name="draw:name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Стиль страницы

Атрибут `draw:style-name` назначает графической странице дополнительные атрибуты форматирования путем указания стиля графической страницы. Это необязательный атрибут. Заданное семейство стилей для станицы — `drawing-page`.

Для страниц внутри документа презентации также могут быть использованы атрибуты страницы презентации (см. раздел 15.36).

```
<define name="draw-page-attlist" combine="interleave">
    <optional>
        <attribute name="draw:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Мастер-страница

Каждая графическая страница должна иметь одну мастер-страницу, связанную с ней. Мастер-страница:

- определяет свойства, такие как размеры и обрамление графической страницы;
- служит контейнером для векторных фигур, которые используются как общий фон.

Атрибут `draw:master-page-name` определяет имя мастер-страницы, связанной с графической страницей. Это обязательный атрибут.

```
<define name="draw-page-attlist" combine="interleave">
    <attribute name="draw:master-page-name">
        <ref name="styleNameRef"/>
    </attribute>
```

```
</define>
```

Разметка страницы презентации

Если графическая страница была создана с использованием разметки страницы презентации, атрибут `presentation:presentation-page-layout-name` ссылается на соответствующий элемент `<style:presentation-page-layout>` (см. раздел 14.15 для получения информации об элементе разметки страницы презентации). Это необязательный атрибут.

```
<define name="draw-page-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:presentation-page-layout-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Объявление верхнего колонтитула

Атрибут `presentation:use-header-name` определяет имя объявления поля верхнего колонтитула (см. раздел 9.11.2), которое используется для всех полей верхнего колонтитула (см. раздел 9.10.1), отображаемых на странице.

```
<define name="common-presentation-header-footer-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:use-header-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Объявление нижнего колонтитула

Атрибут `presentation:use-footer-name` определяет имя объявления поля нижнего колонтитула (см. раздел 9.11.3), которое используется для всех полей нижнего колонтитула (см. раздел 9.10.2), отображаемых на странице.

```
<define name="common-presentation-header-footer-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:use-footer-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Объявление даты и времени

Атрибут `presentation:use-date-time-name` определяет имя объявления поля даты и времени (см. раздел 9.11.4), которое используется для всех полей даты и времени (см. раздел 9.10.3), отображаемых на странице.

```
<define name="common-presentation-header-footer-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:use-date-time-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Идентификатор

Атрибут `draw:id` присваивает уникальный идентификатор графической странице.

```
<define name="draw-page-attlist">
    <optional>
        <attribute name="draw:id">
            <ref name="ID"/>
        </attribute>
    </optional>
</define>
```

9.1.5 Заметки презентации

Каждый элемент графической страницы в презентации может иметь дополнительные заметки презентации, которые содержат краткое описание соответствующей графической страницы и дополнительных графических фигур. Страница заметок описывается элементом `<presentation:notes>`, который может содержаться в элементе `<draw:page>` (см. раздел 14.4.2 для получения дополнительной информации об этом элементе).

Пример – Графическая страница:

```
<office:automatic-styles>
    <style:style style:name="gg3434" style:family="drawing-page">
        <style:drawing-page-properties presentation:page-duration="5s">
    </style:style>
    <style:style style:name="titledia"
                style:family="presentation-page-layout">
        <presentation:placeholder presentation:object="title"
                    svg:x="20%" svg:y="10%"
                    svg:width="80%" svg:height="10%"/>
        <presentation:placeholder presentation:object="subtitle"
                    svg:x="20%" svg:y="30%"
                    svg:width="80%" svg:height="60%" />
    </style:style>
</office:automatic-styles>
```

```
...
<office:body>
    <draw:page office:name="Page 1"     draw:style-name="gg3434"
            draw:master-page-name="home"
            presentation:page-layout-name="titledia">
        <draw:rect .../>
        presentation:notes>
            <draw:text ...>this is a note</draw:text>
        </presentation:notes>
    </draw:page>
</office:body>
```

9.2 Векторные фигуры

Этот раздел описывает векторные фигуры, которые могут встретиться внутри любых приложений.

```
<define name="shape">
    <choice>
        <ref name="draw-rect"/>
        <ref name="draw-line"/>
        <ref name="draw-polyline"/>
        <ref name="draw-polygon"/>
        <ref name="draw-regular-polygon"/>
        <ref name="draw-path"/>
        <ref name="draw-circle"/>
        <ref name="draw-ellipse"/>
        <ref name="draw-g"/>
        <ref name="draw-page-thumbnail"/>
        <ref name="draw-frame"/>
        <ref name="draw-measure"/>
        <ref name="draw-caption"/>
        <ref name="draw-connector"/>
        <ref name="draw-control"/>
        <ref name="dr3d-scene"/>
        <ref name="draw-custom-shape"/>
    </choice>
</define>
```

9.2.1 Прямоугольник

Элемент `<draw:rect>` представляет прямоугольную векторную фигуру.

```
<define name="draw-rect">
    <element name="draw:rect">
        <ref name="draw-rect-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <ref name="draw-text"/>
    </element>
</define>
```

С элементом `<draw:rect>` могут быть связаны следующие атрибуты:

- положение, размер, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- скругленные вершины.

Скругленные вершины

Атрибут `draw:corner-radius` определяет радиус окружности, используемой для скругления вершин прямоугольника.

```
<define name="draw-rect-attlist" combine="interleave">
    <optional>
        <attribute name="draw:corner-radius">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

Пример – Прямоугольная векторная фигура:

```
<draw:rect svg:x="2cm" svg:y="3cm" svg:width="10cm" svg:height="20cm"
svg:transform="rotate(45)" draw:style-name="object-with-shadow">
```

9.2.2 Линия

Элемент `<draw:line>` представляет линию.

```
<define name="draw-line">
    <element name="draw:line">
        <ref name="draw-line-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <ref name="draw-text"/>
    </element>
</define>
```

С элементом `<draw:line>` могут быть связаны следующие атрибуты:

- стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- начальная точка;
- конечная точка.

Начальная точка

Атрибуты начальной точки `svg:x1` и `svg:y1` определяют начальные координаты линии.

```
<define name="draw-line-attlist" combine="interleave">
    <attribute name="svg:x1">
        <ref name="coordinate"/>
    </attribute>
    <attribute name="svg:y1">
        <ref name="coordinate"/>
    </attribute>
</define>
```

Конечная точка

Атрибуты конечной точки `svg:x2` и `svg:y2` определяют конечные координаты линии.

```
<define name="draw-line-attlist" combine="interleave">
    <attribute name="svg:x2">
        <ref name="coordinate"/>
    </attribute>
    <attribute name="svg:y2">
        <ref name="coordinate"/>
    </attribute>
</define>
```

9.2.3 Ломаная

Элемент `<draw:polyline>` представляет векторную фигуру ломаную.

Некоторые реализации могут игнорировать атрибут размера, а вместо него определять размер векторной фигуры исключительно из данных векторной фигуры (т. е. вершин многоугольника).

```
<define name="draw-polyline">
    <element name="draw:polyline">
        <ref name="common-draw-points-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-viewbox-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <ref name="draw-text"/>
    </element>
</define>
```

С элементом `<draw:polyline>` могут быть связаны следующие атрибуты:

- положение, размер, область отображения, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- точки.

Точки

Атрибут `svg:points` хранит последовательность точек, которые соединены прямыми линиями. Каждая точка имеет две координаты. Координаты разделяются запятыми, а точки разделяются пробельными символами.

```
<define name="common-draw-points-attlist">
    <attribute name="draw:points">
        <ref name="points"/>
    </attribute>
</define>
```

9.2.4 Многоугольник

Элемент `<draw:polygon>` представляет многоугольник (полигон). Многоугольник — это замкнутое множество соединенных прямых линий.

Некоторые реализации могут игнорировать атрибут размера, а вместо него определять размер векторной фигуры исключительно из данных векторной фигуры (т. е. вершин многоугольника).

```
<define name="draw-polygon">
    <element name="draw:polygon">
        <ref name="common-draw-points-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-viewbox-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <ref name="draw-text"/>
    </element>
</define>
```

С элементом `<draw:polygon>` могут быть связаны следующие атрибуты:

- положение, размер, область отображения, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- точки (см. раздел 9.2.3).

9.2.5 Правильный многоугольник

Элемент `<draw:regular-polygon>` представляет правильный многоугольник. Правильный многоугольник — это многоугольник, который определяется числом его граней (которое равно числу его вершин) вместо отдельных точек.

```
<define name="draw-regular-polygon">
    <element name="draw:regular-polygon">
        <ref name="draw-regular-polygon-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <ref name="draw-text"/>
    </element>
</define>
```

С элементом `<draw:polygon>` могут быть связаны следующие атрибуты:

- положение, размер, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- выпуклость;
- вершины;
- заостренность.

Выпуклость

Атрибут `draw:concave` определяет, является ли многоугольник выпуклым или невыпуклым. В выпуклом многоугольнике вершины расположены на единственном эллипсе, центр которого совпадает с центром многоугольника. В невыпуклом многоугольнике требуются два таких эллипса, и последовательные вершины расположены на разных эллипсах. Примером выпуклого многоугольника является шестиугольник. Примером невыпуклого многоугольника является звезда. Для невыпуклых многоугольников дополнительный атрибут `draw:sharpness` является обязательным.

```
<define name="draw-regular-polygon-attlist" combine="interleave">
    <choice>
        <attribute name="draw:concave">
```

```
            <value>false</value>
        </attribute>
        <group>
            <attribute name="draw:concave">
                <value>true</value>
            </attribute>
            <ref name="draw-regular-polygon-sharpness-attlist"/>
        </group>
    </choice>
</define>
```

Вершины

Атрибут `draw:corners` определяет число вершин многоугольника.

```
<define name="draw-regular-polygon-attlist" combine="interleave">
    <attribute name="draw:corners">
        <ref name="positiveInteger"/>
    </attribute>
</define>
```

Заостренность

Для атрибутов невыпуклых многоугольников атрибут `draw:sharpness` определяет радиус эллипса, на котором расположены внутренние вершины многоугольника. Значение атрибута задается в процентах: 0 % означает, что все вершины расположены на одном эллипсе, а 100 % означает, что внутренние вершины расположены в центральной точке многоугольника. В общем, если *r* — радиус многоугольника и *s* — заостренность, внутренние вершины расположены на эллипсе, радиус которого составляет *r(100-s)/100*.

```
<define name="draw-regular-polygon-sharpness-attlist">
    <attribute name="draw:sharpness">
        <ref name="percent"/>
    </attribute>
</define>
```

9.2.6 Траектория

Элемент `<draw:path>` представляет траекторию. Траектория — это векторная фигура с определенным пользователем контуром. Векторная фигура строится путем использования множества действий, таких как:

- *moveto* – установить новую текущую точку;
- *lineto* – нарисовать соединительную линию;
- *curveto* – нарисовать кубическую кривую Безье;
- *arc* – нарисовать эллиптическую или круговую дугу;

- *closepath* – замкнуть текущую векторную фигуру, нарисовав линию к последнему *moveto.*

Составные траектории — это траектории с вложенными траекториями, каждая вложенная траектория состоит из одного *moveto* с последующей одной или несколькими операциями рисования линии или кривой. Составные траектории могут быть использованы для таких эффектов как отверстия внутри объектов.

Некоторые реализации могут игнорировать атрибут размера, а вместо него определять размер векторной фигуры исключительно из данных векторной фигуры (т. е. вершин многоугольника).

```
<define name="draw-path">
    <element name="draw:path">
        <ref name="common-draw-path-data-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-viewbox-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <ref name="draw-text"/>
    </element>
</define>
```

С элементом `<draw:path>` могут быть связаны следующие атрибуты:

- положение, размер, область отображения, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- данные траектории.

Данные траектории

Синтаксис атрибута `svg:d` описан в § 8 *Scalable Vector Graphics (SVG) 1.1 Specification* [SVG].

Некоторые реализации могут поддерживать только подмножество спецификаций траекторий SVG, например не поддерживать сочетание открытых и закрытых кривых для одной векторной фигуры или не поддерживать команду эллиптической дуги.

```
<define name="common-draw-path-data-attlist">
    <attribute name="svg:d">
        <ref name="pathData"/>
```

```
    </attribute>
</define>
```

9.2.7 Окружность

Элемент `<draw:circle>` представляет круглую векторную фигуру.

```
<define name="draw-circle">
    <element name="draw:circle">
        <ref name="draw-circle-attlist"/>
        <ref name="common-draw-circle-ellipse-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <ref name="draw-text"/>
    </element>
</define>
```

С элементом `<draw:circle>` могут быть связаны следующие атрибуты:

- положение, размер, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- центр;
- радиус;
- тип;
- начальный угол;
- конечный угол.

Центр

Атрибуты центра `svg:cx` и `svg:cy` определяют координаты центра окружности. Если эти необязательные атрибуты не установлены, для создания окружности используются атрибуты положения и размера.

```
<define name="common-draw-circle-ellipse-attlist" combine="interleave">
    <optional>
        <attribute name="svg:cx">
            <ref name="coordinate"/>
        </attribute>
        <attribute name="svg:cy">
            <ref name="coordinate"/>
        </attribute>
    </optional>
</define>
```

Радиус

Атрибут `svg:r` определяет радиус окружности. Если этот необязательный атрибут не используется, для создания окружности используются атрибуты положения и размера.

```
<define name="draw-circle-attlist" combine="interleave">
    <optional>
        <attribute name="svg:r">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

Тип

Атрибут `draw:kind` определяет внешний вид окружности.

- `full` — определяет полную окружность или эллипс, например ◯ ;
- `section` — определяет сектор круга или эллипса, например ◔ ;
- `cut` — определяет окружность или эллипс с обрезкой, например ◯ ;
- `arc` — определяет дугу окружности или эллипса, например ◡ .

```
<define name="common-draw-circle-ellipse-attlist" combine="interleave">
    <optional>
        <attribute name="draw:kind" a:defaultValue="full">
            <choice>
                <value>full</value>
                <value>section</value>
                <value>cut</value>
                <value>arc</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Начальный угол

Для окружностей, у которых атрибут `draw:kind` установлен как `section`, `cut` или `arc`, атрибут `svg:start-angle` определяет начальный угол сегмента круга, обрезанного круга или дуги окружности.

```
<define name="common-draw-circle-ellipse-attlist" combine="interleave">
    <optional>
        <attribute name="draw:start-angle">
            <ref name="double"/>
        </attribute>
    </optional>
</define>
```

Конечный угол

Для окружностей, у которых атрибут `draw:kind` установлен как `section`, `cut` или `arc`, атрибут `svg:end-angle` определяет конечный угол сегмента круга, обрезанного круга или дуги окружности.

```
6146 <define name="common-draw-circle-ellipse-attlist" combine="interleave">
6147     <optional>
6148         <attribute name="draw:end-angle">
6149             <ref name="double"/>
6150         </attribute>
6151     </optional>
6152 </define>
```

9.2.8 Эллипс

Элемент `<draw:ellipse>` представляет эллипс.

```
6153 <define name="draw-ellipse">
6154     <element name="draw:ellipse">
6155         <ref name="common-draw-circle-ellipse-attlist"/>
6156         <ref name="draw-ellipse-attlist"/>
6157         <ref name="common-draw-position-attlist"/>
6158         <ref name="common-draw-size-attlist"/>
6159         <ref name="common-draw-shape-with-text-and-styles-attlist"/>
6160         <optional>
6161             <ref name="office-event-listeners"/>
6162         </optional>
6163         <zeroOrMore>
6164             <ref name="draw-glue-point"/>
6165         </zeroOrMore>
6166         <ref name="draw-text"/>
6167     </element>
6168 </define>
```

С элементом `<draw:ellipse>` могут быть связаны следующие атрибуты:

- положение, размер, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- центр, тип, начальный угол, конечный угол (см. раздел 9.2.7);
- радиус.

Радиус

Атрибуты `svg:rx` и `svg:rx` определяют горизонтальный и вертикальный радиус эллипса. Если эти необязательные атрибуты не установлены, для создания эллипса используются атрибуты положения и размера.

```
6169 <define name="draw-ellipse-attlist" combine="interleave">
6170     <optional>
```

```
6171        <attribute name="svg:rx">
6172            <ref name="length"/>
6173        </attribute>
6174        <attribute name="svg:ry">
6175            <ref name="length"/>
6176        </attribute>
6177    </optional>
6178 </define>
```

9.2.9 Соединительный элемент

Элемент `<draw:connector>` представляет множество линий, которые соединены с точками привязки двух других векторных фигур.

```
6179 <define name="draw-connector">
6180     <element name="draw:connector">
6181         <ref name="draw-connector-attlist"/>
6182         <ref name="common-draw-shape-with-text-and-styles-attlist"/>
6183         <optional>
6184             <ref name="office-event-listeners"/>
6185         </optional>
6186         <zeroOrMore>
6187             <ref name="draw-glue-point"/>
6188         </zeroOrMore>
6189         <ref name="draw-text"/>
6190     </element>
6191 </define>
```

С элементом `<draw:connector>` могут быть связаны следующие атрибуты:

- стиль, слой, Z-индекс, идентификатор (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- тип;
- начальное положение;
- начальная векторная фигура;
- начальная точка привязки;
- конечное положение;
- конечная векторная фигура;
- конечная точка привязки;
- наклон линии.

Тип

Атрибут `draw:type` определяет, каким образом формируется изображение соединения между двумя точками. Значениями данного атрибута могут быть `standard`, `lines`, `line`, или `curve` :

- `standard` — стандартный соединительный элемент, выходящий из двух соединенных объектов с прямыми линиями и соединяющий их прямой перпендикулярной линией;
- `lines` — многолинейный соединительный элемент, выходящий из двух соединенных объектов с прямыми линиями и соединяющей их прямой линией (не обязательно перпендикулярной);
- `line` — линейный соединительный элемент, нарисованный одной прямой линией между двумя точками перехода соединенных объектов;
- `curve` — дуговой соединительный элемент, нарисованный одиночной дугой между двумя точками перехода соединенных объектов.

```
<define name="draw-connector-attlist" combine="interleave">
    <optional>
        <attribute name="draw:type" a:defaultValue="standard">
            <choice>
                <value>standard</value>
                <value>lines</value>
                <value>line</value>
                <value>curve</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Начальное положение

Атрибуты начального положения `svg:x1` и `svg:y1` определяют начальное положение соединительного элемента.

Если начальная позиция соединена с векторной фигурой, эти атрибуты необязательны, поскольку начальная позиция по умолчанию относится к соответствующей точке привязки на целевой векторной фигуре.

```
<define name="draw-connector-attlist" combine="interleave">
    <optional>
        <attribute name="svg:x1">
            <ref name="coordinate"/>
        </attribute>
        <attribute name="svg:y1">
            <ref name="coordinate"/>
        </attribute>
    </optional>
</define>
```

Начальная векторная фигура

Атрибут `draw:start-shape` определяет векторную фигуру, на которую по имени ссылается начало соединительного элемента.

Если векторная фигура соединена с началом соединительного элемента, начальная позиция определяется по умолчанию как точка привязки на выбранной векторной фигуре.

```
<define name="draw-connector-attlist" combine="interleave">
    <optional>
        <attribute name="draw:start-shape">
            <ref name="IDREF"/>
        </attribute>
    </optional>
</define>
```

Начальная точка привязки

Атрибут `draw:start-glue-point` идентифицирует точку привязки в начальной векторной фигуре соединительного элемента по ее номеру (см. раздел 9.2.19 для получения дополнительной информации о точках привязки).

Если этот атрибут не установлен и начало соединительного элемента связано с векторной фигурой, приложение само может выбрать точку привязки. Если начало соединительного элемента не связано с векторной фигурой, то этот атрибут игнорируется.

```
<define name="draw-connector-attlist" combine="interleave">
    <optional>
        <attribute name="draw:start-glue-point">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Конечное положение

Атрибуты конечного положения `svg:x2` и `svg:y2` определяют конечное положение соединительного элемента.

Если конечное положение связано с векторной фигурой, эти атрибуты необязательны, поскольку конечное положение по умолчанию устанавливается как соответствующая точка привязки выбранной векторной фигуры.

```
<define name="draw-connector-attlist" combine="interleave">
    <optional>
        <attribute name="svg:x2">
            <ref name="coordinate"/>
        </attribute>
        <attribute name="svg:y2">
            <ref name="coordinate"/>
        </attribute>
    </optional>
</define>
```

Конечная векторная фигура

Атрибут `draw:end-shape` идентифицирует по имени векторную фигуру, с которой связан конец соединительного элемента.

Если векторная фигура связана с концом соединительного элемента, конечное положение устанавливается по умолчанию как соответствующая точка привязки на выбранной векторной фигуре.

```
<define name="draw-connector-attlist" combine="interleave">
    <optional>
        <attribute name="draw:end-shape">
            <ref name="IDREF"/>
        </attribute>
    </optional>
</define>
```

Конечная точка привязки

Атрибут `draw:end-glue-point` идентифицирует точку привязки в конечной векторной фигуре соединительного элемента по ее номеру (см. раздел 9.2.19 для получения дополнительной информации о точках привязки).

Если этот атрибут не установлен и конец соединительного элемента связан с векторной фигурой, приложение может само выбрать точку привязки. Если конец соединительного элемента не привязан к векторной фигуре, этот атрибут игнорируется.

```
<define name="draw-connector-attlist" combine="interleave">
    <optional>
        <attribute name="draw:end-glue-point">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Наклон линии

Атрибут `draw:line-skew` контролирует формирование линий, которые соединяют начало и конец соединительного элемента. В зависимости от типа соединительного элемента, он может изменяться от одного до трех дистанций, сдвигая соединительный элемент относительно его нормального положения.

```
<define name="draw-connector-attlist" combine="interleave">
    <optional>
        <attribute name="draw:line-skew">
            <list>
                <ref name="length"/>
                <optional>
```

```
                    <ref name="length"/>
                    <optional>
                        <ref name="length"/>
                    </optional>
                </optional>
            </list>
        </attribute>
    </optional>
</define>
```

9.2.10 Подпись

Элемент `<draw:caption>` представляет прямоугольную векторную фигуру с дополнительным набором линий. Он может быть использован как описание для точки внутри рисунка.

```
<define name="draw-caption">
    <element name="draw:caption">
        <ref name="draw-caption-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <ref name="draw-text"/>
    </element>
</define>
```

С элементом `<draw:caption>` могут быть связаны следующие атрибуты:

- положение, размер, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- точка подписи;
- скругленные вершины.

Точка подписи

Атрибуты точки подписи `draw:caption-point-x` и `draw:caption-point-y` определяют положение названной точки. Множество линий отображаются из области заголовка.

```
<define name="draw-caption-attlist" combine="interleave">
    <optional>
        <attribute name="draw:caption-point-x">
            <ref name="coordinate"/>
        </attribute>
        <attribute name="draw:caption-point-y">
```

```
            <ref name="coordinate"/>
        </attribute>
    </optional>
</define>
```

Скругленные вершины

Атрибут `draw:corner-radius` определяет радиус окружности, которая используется для скругления вершин подписи.

```
<define name="draw-caption-attlist" combine="interleave">
    <optional>
        <attribute name="draw:corner-radius">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

9.2.11 Измерение

Элемент `<draw:measure>` представляет векторную фигуру, которая используется для измерения расстояний на изображениях.

```
<define name="draw-measure">
    <element name="draw:measure">
        <ref name="draw-measure-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <ref name="draw-text"/>
    </element>
</define>
```

С элементом `<draw:measure>` могут быть связаны следующие атрибуты:

- стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- начальное положение;
- конечное положение.

Начальное положение

Атрибуты `svg:x1` и `svg:y1` определяют начальную точку измеряемой дистанции.

```
<define name="draw-measure-attlist" combine="interleave">
```

```
    <attribute name="svg:x1">
        <ref name="coordinate"/>
    </attribute>
    <attribute name="svg:y1">
        <ref name="coordinate"/>
    </attribute>
</define>
```

Конечное положение

Атрибуты `svg:x2` и `svg:y2` определяют конечную точку измеряемой дистанции.

```
<define name="draw-measure-attlist" combine="interleave">
    <attribute name="svg:x2">
        <ref name="coordinate"/>
    </attribute>
    <attribute name="svg:y2">
        <ref name="coordinate"/>
    </attribute>
</define>
```

9.2.12 Элемент управления

Элемент `<draw:control>` представляет векторную фигуру, которая связана с элементом управления внутри элемента `<office:forms>` (см. раздел 11.1).

```
<define name="draw-control">
    <element name="draw:control">
        <ref name="draw-control-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<draw:control>` могут быть связаны следующие атрибуты:

- положение, размер, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- элемент управления.

Элемент управления

Атрибут `draw:control` определяет элемент управления внутри формы (см. раздел 11.5.2), который привязан к векторной фигуре элемента управления.

```
<define name="draw-control-attlist" combine="interleave">
```

```
    <attribute name="draw:control">
        <ref name="IDREF"/>
    </attribute>
</define>
```

9.2.13 Уменьшенное изображение страницы

Элемент `<draw:page-thumbnail>` представляет прямоугольную область, отображающую уменьшенное изображение страницы страницы рисования.

```
<define name="draw-page-thumbnail">
    <element name="draw:page-thumbnail">
        <ref name="draw-page-thumbnail-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="presentation-shape-attlist"/>
        <ref name="common-draw-shape-with-styles-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<draw:page-thumbnail>` могут быть связаны следующие атрибуты:

- положение, размер, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- класс презентации (см. раздел 9.6.1);
- номер страницы.

Номер страницы

Атрибут `draw:page-number` определяет номер страницы, которая будет отображаться в уменьшенном изображении страницы. Для уменьшенных изображений страниц примечаний значение данного атрибута прикреплено к графической странице страницы заметок. Для уменьшенных изображений страниц мастер-страниц тезисов значением данного атрибута является очередность, в которой страницы просматриваются на странице тезисов. Например, на странице тезисов с 4 уменьшенными изображениями страницы, уменьшенное изображение с наименьшим номером страницы отображает первую страницу, когда печатается первая страница тезисов, и пятую, когда печатается вторая страница тезисов, и так далее.

```
<define name="draw-page-thumbnail-attlist">
    <optional>
        <attribute name="draw:page-number">
```

```
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

9.2.14 Группировка

Элемент `<draw:g>` представляет группу векторных фигур.

```
<define name="draw-g">
    <element name="draw:g">
        <ref name="draw-g-attlist"/>
        <ref name="common-draw-z-index-attlist"/>
        <ref name="common-draw-name-attlist"/>
        <ref name="common-draw-id-attlist"/>
        <ref name="common-draw-style-name-attlist"/>
        <ref name="common-text-spreadsheet-shape-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="shape"/>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<draw:g>` могут быть связаны следующие атрибуты:

- стиль, Z-индекс и идентификатор (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- положение.

Положение

Для сгрупп ированной векторной фигуры, которая содержится в текстовом документе и привязывается как символ, атрибут `svg:y` определяет вертикальное положение векторной фигуры.

```
<define name="draw-g-attlist" combine="interleave">
    <optional>
        <attribute name="svg:y">
            <ref name="coordinate"/>
        </attribute>
    </optional>
</define>
```

9.2.15 Общие атрибуты векторных фигур

Атрибуты, описанные в данном разделе, являются общими для всех векторных фигур.

Имя

Атрибут `draw:name` назначает имя векторной фигуре.

```
<define name="common-draw-name-attlist" combine="interleave">
    <optional>
        <attribute name="draw:name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Положение

Атрибуты положения `svg:x` и `svg:y` определяют координаты x и y начального положения векторной фигуры.

```
<define name="common-draw-position-attlist">
    <optional>
        <attribute name="svg:x">
            <ref name="coordinate"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="svg:y">
            <ref name="coordinate"/>
        </attribute>
    </optional>
</define>
```

Размер

Атрибуты `svg:width` и `svg:height` определяют ширину и высоту векторной фигуры.

```
<define name="common-draw-size-attlist">
    <optional>
        <attribute name="svg:width">
            <ref name="length"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="svg:height">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

Преобразование

Атрибут `draw:transform` определяет список преобразований, которые могут быть применены к векторной фигуре.

Значением атрибута является список определений преобразования, которые применяются к векторной фигуре в порядке их перечисления в списке. Определения преобразований в списке должны быть разделены пробельными символами и/или запятыми. Типы допустимых определений преобразований включают:

- matrix(<a> <b> <c> <d> <e> <f>) — определяет преобразование в виде матричного преобразования шести значений, matrix(a,b,c,d,e,f) эквивалентно применению матрицы преобразования [a b c d e f];
- translate(<tx> [<ty>]) — определяет перемещение на tx и ty;
- scale(<sx> [<sy>]) — определяет операцию масштабирования на sx и sy, если <sy> не представлен, он принимается равным <sx>;
- rotate(<rotate-angle>) — определяет вращение на <rotate-angle> относительно начала системы координат векторных фигур;
- skewX(<skew-angle>) — определяет наклонное преобразование вдоль оси X;
- skewY(<skew-angle>) — определяет наклонное преобразование вдоль оси Y.

```
<define name="common-draw-transform-attlist">
    <optional>
        <attribute name="draw:transform">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Область отображения

Атрибут svg:viewBox устанавливает пользовательскую систему координат внутри физической системы координат векторной фигуры, определенной атрибутами положения и размера. Пользовательская система координат используется атрибутом svg:points и элементом <draw:path>.

Синтаксис использования этого атрибута такой же, как и синтаксис в [SVG]. Значением атрибута являются четыре числа, разделенных пробельными символами, которые определяют левое, верхнее, правое и нижнее измерения пользовательской системы координат.

Некоторые реализации могут игнорировать атрибут области отображения. Предполагаемая система координат в таком случае имеет начало в левой верхней вершине векторной фигуры без какого-либо масштабирования относительно векторной фигуры.

```
<define name="common-draw-viewbox-attlist">
    <attribute name="svg:viewBox">
        <list>
            <ref name="integer"/>
            <ref name="integer"/>
            <ref name="integer"/>
            <ref name="integer"/>
        </list>
    </attribute>
</define>
```

Стиль

Атрибуты draw:style-name и presentation:style-name определяют стиль для векторной фигуры. Если используется атрибут draw:style-name, векторная фигура является обычной векторной фигурой. Если используется атрибут presentation:style-name, векторная фигура является векторной фигурой презентации, как описано в разделе 9.6.

Значением обоих атрибутов является имя элемента <style:style>. Если используется атрибут draw:style-name, стиль должен принимать значение из группы graphic. Если используется атрибут presentation:style-name, стиль должен принимать значение из группы presentation. Свойства форматирования определенного стиля и его необязательного родительского стиля используются для форматирования векторной фигуры (см. также раздел 14.13.1).

Атрибуты draw:class-names и presentation:class-names принимают список разделенных пробельными символами либо имен графических стилей, либо имен стилей презентации. Указанные стили применяются в том порядке, в каком они содержатся в списке. Если представлены атрибуты draw:style-name и draw:class-names, или атрибуты presentation:style-name и presentation:class-names, то стиль, указанный атрибутом style-name, трактуется как первый стиль в списке стилей атрибута class-names. Соответствующее приложение должно поддерживать атрибут class-names, а также сохранять его во время редактирования.

```
<define name="common-draw-style-name-attlist">
```

```
    <choice>
        <group>
            <optional>
                <attribute name="draw:style-name">
                    <ref name="styleNameRef"/>
                </attribute>
            </optional>
            <optional>
                <attribute name="draw:class-names">
                    <ref name="styleNameRefs"/>
                </attribute>
            </optional>
        </group>
        <group>
            <optional>
                <attribute name="presentation:style-name">
                    <ref name="styleNameRef"/>
                </attribute>
            </optional>
            <optional>
                <attribute name="presentation:class-names">
                    <ref name="styleNameRefs"/>
                </attribute>
            </optional>
        </group>
    </choice>
</define>
```

Стиль текста

Атрибут `draw:text-style-name` определяет стиль векторной фигуры, используемый для форматирования текста, который может быть добавлен к этой векторной фигуре.

Значением этого атрибута является имя элемента `<style:style>` со значением из группы `paragraph`.

```
<define name="common-draw-text-style-name-attlist">
    <optional>
        <attribute name="draw:text-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Слой

Атрибут `draw:layer` может привязать каждую векторную фигуру к слою. Значением данного атрибута должно быть имя слоя из множества доступных слоев внутри документа.

```
<define name="common-draw-layer-name-attlist">
    <optional>
        <attribute name="draw:layer">
            <data type="string"/>
```

```
        </attribute>
    </optional>
</define>
```

Идентификатор

Атрибут `draw:id` присваивает уникальный идентификатор к векторной фигуре, который может быть использован для ссылки на эту векторную фигуру.

```
<define name="common-draw-id-attlist">
    <optional>
        <attribute name="draw:id">
            <ref name="ID"/>
        </attribute>
    </optional>
</define>
```

Z-индекс

Векторные фигуры отображаются в определенном порядке. Вообще, векторные фигуры отображаются в том порядке, в каком они появляются в XML-документе. Чтобы изменить этот порядок, используется атрибут `svg:z-index`.

Это необязательный атрибут.

```
<define name="common-draw-z-index-attlist">
    <optional>
        <attribute name="draw:z-index">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

9.2.16 Общие атрибуты векторных фигур для текстовых документов и документов электронных таблиц

Атрибуты, описанные в этом разделе, являются общими для всех векторных фигур, содержащихся в текстовом документе или документе электронных таблиц.

Конечное положение

Если векторная фигура включена в документ электронных таблиц и привязка векторной фигуры осуществлена к ячейке, то атрибуты `table:end-cell-address`, `table:end-x` и `table:end-y` определяют конечное положение векторной фигуры, и атрибуты размера игнорируются. Конечное положение определяется с использованием адреса ячейки, в которой находится конечное

положение, и координат `x` и `y` конечного положения относительно верхней и левой граней ячейки.

```
<define name="common-text-spreadsheet-shape-attlist" combine="interleave">
    <optional>
        <attribute name="table:end-cell-address">
            <ref name="cellAddress"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="table:end-x">
            <ref name="coordinate"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="table:end-y">
            <ref name="coordinate"/>
        </attribute>
    </optional>
</define>
```

Фон таблицы

Если векторная фигура включена в документ электронной таблицы, то атрибут `table:table-background` определяет, будет ли содержаться векторная фигура в фоне таблицы. Если атрибут отсутствует, то векторная фигура помещается на передний план таблицы.

```
<define name="common-text-spreadsheet-shape-attlist" combine="interleave">
    <optional>
        <attribute name="table:table-background">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Привязка

Внутри текстового документа атрибут типа привязки `text:anchor-type` определяет, каким образом фрейм обрамляет текстовый документ. Положение привязки — точка, в которой фрейм обрамляет текстовый документ. Положение привязки зависит от типа привязки, как это описано в следующей таблице:

Значение атрибута `text:anchor-type` ...	*Положение привязки...*	*Элемент векторной фигуры отобразится ...*	*Примечания*
page	Если к элементу векторной фигуры присоединен атрибут	В начале основного содержимого документа, снаружи любого абзаца	Физический номер страницы — это номер,

Значение атрибута text:anchor-type ...	*Положение привязки...*	*Элемент векторной фигуры отобразится ...*	*Примечания*
	text:anchor-page-number, положение привязки — страница, которая имеет такой же физический номер, что и значение атрибута. Если атрибут text:anchor-page-number отсутствует, положение привязки — это страница, на которой появляется символ, стоящий следом за элементом графического объекта	или фрейма, указанного атрибутом text:anchor-page-number . Внутри любого элемента абзаца, который не содержится внутри верхнего колонтитула, нижнего колонтитула, сноски или текстового окна, если не указан атрибут text:anchor-page-number	присвоенный странице, если все страницы в документе пронумерованы, начиная с номера 1
frame	Родительский текстовый блок, в котором содержится текущий элемент векторной фигуры	В элементе, представляющем текстовый блок, которым обрамлен графический объект. Например, если изображение обрамлено текстовым блоком, элемент изображения расположен в элементе текстового блока	
paragraph	Абзац, в котором содержится текущий элемент векторной фигуры	В начале элемента абзаца	
char	Символ после элемента векторной фигуры	Прямо перед символом	
as-char	В данном случае положение привязки отсутствует. Векторная	На позиции, где в документе отображается символ	

Значение атрибута text:anchor-type ...	*Положение привязки...*	*Элемент векторной фигуры отобразится ...*	*Примечания*
	фигура ведет себя как символ		

```
<define name="common-text-spreadsheet-shape-attlist" combine="interleave">
    <ref name="common-text-anchor-attlist"/>
</define>

<define name="common-text-anchor-attlist" combine="interleave">
    <optional>
        <attribute name="text:anchor-type">
            <choice>
                <value>page</value>
                <value>frame</value>
                <value>paragraph</value>
                <value>char</value>
                <value>as-char</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Номер страницы привязки

Внутри текстовых документов, атрибут `text:anchor-page-number` определяет физический номер страницы привязки, если графический объект обрамляет страницу.

```
<define name="common-text-anchor-attlist" combine="interleave">
    <optional>
        <attribute name="text:anchor-page-number">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

9.2.17 Общее содержимое векторных фигур

Большинство векторных фигур может иметь текстовое содержимое. Текстовое содержимое может включать абзацы (см. раздел 4.1.2), а также списки (см. раздел 4.3).

```
<define name="draw-text">
    <zeroOrMore>
        <choice>
            <ref name="text-p"/>
            <ref name="text-list"/>
        </choice>
    </zeroOrMore>
```

```
</define>
```

9.2.18 Общие группы атрибутов векторных фигур

Последующие определенные атрибуты — общие для всех векторных фигур, поддерживающих стили и не поддерживающих текст.

```
<define name="common-draw-shape-with-styles-attlist">
    <ref name="common-draw-z-index-attlist"/>
    <ref name="common-draw-id-attlist"/>
    <ref name="common-draw-layer-name-attlist"/>
    <ref name="common-draw-style-name-attlist"/>
    <ref name="common-draw-transform-attlist"/>
    <ref name="common-draw-name-attlist"/>
    <ref name="common-text-spreadsheet-shape-attlist"/>
</define>
```

Последующие определенные атрибуты — общие для всех векторных фигур, поддерживающих стили и текст.

```
<define name="common-draw-shape-with-text-and-styles-attlist">
    <ref name="common-draw-shape-with-styles-attlist"/>
    <ref name="common-draw-text-style-name-attlist"/>
</define>
```

9.2.19 Точки привязки

Точки привязки — это специально назначенные точки на графическом объекте, с которыми может соединяться векторная фигура соединительного элемента. Большинство графических объектов имеют четыре стандартных точки привязки, расположенных на четырех ребрах объекта. Дополнительные точки привязки могут быть добавлены к графическому объекту путем вставки одного или более элементов `<draw:glue-point>` в элемент графического объекта. Элемент `<draw:glue-point>` создает одиночную определяемую пользователем точку привязки, если он помещается внутри элемента графического объекта, например, внутри элемента `<draw:rectangle>`.

```
<define name="draw-glue-point">
    <element name="draw:glue-point">
        <ref name="draw-glue-point-attlist"/>
        <empty/>
    </element>
</define>
```

Идентификатор

Атрибут `draw:id` содержит идентификатор точки привязки. Идентификатор — это номер, который используется внутри атрибутов `draw:start-glue-point` и

draw:end-glue-point элемента <draw:connector>. Идентификаторы от 0 до 3 зарезервированы для четырех стандартных точек привязки, которые имеет большинство графических объектов. Точки привязки нумеруются по часовой стрелке, начиная с левой верхней вершины векторной фигуры.

```
<define name="draw-glue-point-attlist" combine="interleave">
    <attribute name="draw:id">
        <ref name="nonNegativeInteger"/>
    </attribute>
</define>
```

Положение

Атрибуты svg:x и svg:y определяют положение точки привязки. Координаты являются или процентными значениями относительно центра графической фигуры, или, если также определен атрибут draw:align, абсолютными значениями дистанций до вершины, определенной атрибутом draw:align.

```
<define name="draw-glue-point-attlist" combine="interleave">
    <attribute name="svg:x">
        <choice>
            <ref name="distance"/>
            <ref name="percent"/>
        </choice>
    </attribute>
    <attribute name="svg:y">
        <choice>
            <ref name="distance"/>
            <ref name="percent"/>
        </choice>
    </attribute>
</define>
```

Выравнивание

Атрибут draw:align определяет поведение выравнивания точки привязки в случае, когда изменяются размеры графического объекта и грань векторной фигуры, к которой относится положение точки привязки. Пропущенные значения атрибутов горизонтального или вертикального положения подразумевают, что точка привязки горизонтально или вертикально центрируется.

```
<define name="draw-glue-point-attlist" combine="interleave">
    <attribute name="draw:align">
        <choice>
            <value>top-left</value>
            <value>top</value>
            <value>top-right</value>
            <value>left</value>
            <value>center</value>
```

```
            <value>right</value>
            <value>bottom-left</value>
            <value>bottom-right</value>
        </choice>
    </attribute>
</define>
```

Направление выхода

Атрибут `draw:escape-direction` определяет направление, в котором соединительная линия выходит из графического объекта, если соединительный элемент связан с точкой привязки. Значение `horizontal` подразумевает возможность выхода соединительной линии `left` (влево) или `right` (вправо), значение `vertical` подразумевает возможность выхода соединительной линии `up` (вверх) или `down` (вниз).Значение `auto` подразумевает возможность выхода соединительной линии в любом из четырех направлений.

```
<define name="draw-glue-points-attlist" combine="interleave">
    <attribute name="draw:escape-direction">
        <choice>
            <value>auto</value>
            <value>left</value>
            <value>right</value>
            <value>up</value>
            <value>down</value>
            <value>horizontal</value>
            <value>vertical</value>
        </choice>
    </attribute>
</define>
```

9.2.20 Датчики событий

Векторные фигуры могут иметь присоединенные датчики событий. Датчики событий, присоединенные, например, к текстовому блоку или изображению, представлены элементами событий, описанными в разделе 12.4. Этот элемент располагается внутри элемента графического объекта, например, внутри элемента `<draw:text-box>` или `<draw:image>`.

9.3 Фреймы

Фрейм — это прямоугольный контейнер, который может включать содержимое с расширенными возможностями, такое как текстовые блоки, изображения или объекты. Фреймы очень похожи на обычные векторные фигуры, но в отличие от них поддерживают некоторые функции, которые недоступны для

обычных векторных фигур, такие как контуры, сенсорные изображения и гиперссылки. В частности, фреймы позволяют иметь множество преобразованных копий объекта. Например, фрейм может содержать объект наравне с изображением. В этом случае, приложение может выбирать содержимое, которое поддерживается им наилучшим образом. Если приложение поддерживает типы объектов, содержащихся в фрейме, оно, вероятно, будет отображать объект. Если приложение не поддерживает объект, оно будет отображать изображение.

В общем случае приложение не должно отображать больше одного элемента содержимого, расположенного во фрейме. Порядок элементов содержимого определяется предпочтениями автора документов относительно визуализации, где первый элемент-потомок является наиболее предпочтительным. Это означает, что приложение будет отображать дочерний элемент, который оно поддерживает. Фрейм должен включать как минимум один элемент содержимого. Включение различных элементов содержимого не обязательно. Приложения могут сохранять элементы содержимого, которые они не отображают, но не обязаны этого делать.

Внутри текстовых документов фреймы также используются для расположения содержимого за пределами обычной последовательности текста в документе.

Фреймы могут содержать:

- текстовые блоки;
- объекты, представленные или в формате OpenDocument, или в специальном бинарном формате объекта;
- изображения;
- апплеты;
- плагины;
- плавающие фреймы.

Так же, как и свойства форматирования векторных фигур, свойства форматирования фреймов хранятся в стилях, принадлежащих семейству `graphic`. Способ размещения фрейма в документе такой же, как и для векторных фигур.

```
<define name="draw-frame">
    <element name="draw:frame">
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
```

```
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-rel-size-attlist"/>
        <ref name="presentation-shape-attlist"/>
        <ref name="draw-frame-attlist"/>
        <zeroOrMore>
            <choice>
                <ref name="draw-text-box"/>
                <ref name="draw-image"/>
                <ref name="draw-object"/>
                <ref name="draw-object-ole"/>
                <ref name="draw-applet"/>
                <ref name="draw-floating-frame"/>
                <ref name="draw-plugin"/>
            </choice>
        </zeroOrMore>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <optional>
            <ref name="draw-image-map"/>
        </optional>
        <optional>
            <ref name="svg-desc"/>
        </optional>
        <optional>
            <choice>
                <ref name="draw-contour-polygon"/>
                <ref name="draw-contour-path"/>
            </choice>
        </optional>
    </element>
</define>
```

С элементом `<draw:frame>` могут быть связаны следующие атрибуты:

- положение, размер (относительные размеры, см. ниже), стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- класс презентации (см. раздел 9.6.1);
- копирование фреймов.

Следующие элементы могут быть расположены внутри элемента изображения:

- датчики событий (см. раздел 12.4);
- точки привязки (см. раздел 9.2.19);
- сенсорное изображение (см. раздел 9.3.11);
- альтернативный текст (см. раздел 9.3.9);
- контуры (см. раздел 9.3.8).

Относительные размеры

Для фреймов ширина и высота графического объекта может быть установлена как относительное значение с использованием атрибутов `style:rel-width` и `style:rel-height`. Относительное значение — это либо значение в процентах, либо специальное значение `scale`, либо специальное значение `scale-min`.

Интерпретация относительных значений зависит от привязки графического объекта. Если привязка для графического объекта в ячейке таблицы, то процентное значение берется относительно окружающего табличного блока. Если привязка графического объекта в текстовом блоке, процентное значение берется относительно окружающего текстового блока. В остальных случаях процентное значение берется относительно ширины страницы или окна.

Значение `scale` для ширины означает, что ширина должна вычисляться в зависимости от высоты, с тем чтобы пропорции оригинального объекта или изображения сохранялись.

Значение `scale` для высоты подразумевает, что высота должна вычисляться в зависимости от ширины, с тем чтобы пропорции оригинального объекта или изображения сохранялись.

Значение `scale-min` эквивалентно значению `scale`, за исключением того, что вычисленные ширина или высота — это минимальные величины, а не абсолютные значения.

Для поддержки приложений, которые не поддерживают относительные ширину и высоту, приложения, которые сохраняют атрибуты `style:rel-width` или `style:rel-height`, должны также предоставлять реальную ширину и высоту в атрибутах `svg:width` и `svg:height/fo:min-height`.

```
6650 <define name="common-draw-rel-size-attlist">
6651     <ref name="common-draw-size-attlist"/>
6652     <optional>
6653         <attribute name="style:rel-width">
6654             <choice>
6655                 <ref name="percent"/>
6656                 <value>scale</value>
6657                 <value>scale-min</value>
6658             </choice>
6659         </attribute>
6660     </optional>
6661     <optional>
6662         <attribute name="style:rel-height">
6663             <choice>
```

```
                <ref name="percent"/>
                <value>scale</value>
                <value>scale-min</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Копирование фреймов

Многочисленные фреймы могут быть установлены для отображения одного и того же содержимого: например, для логотипа компании, который должен встречаться в любом месте каждой страницы, не являясь частью верхнего или нижнего колонтитула.

Фрейм может быть установлен на отображение содержимого другого фрейма, указанного атрибутом `draw:copy-of`. Это не воздействует на стиль и положение. Фрейм, имеющий атрибут `draw:copy-of`, имеет свои собственные стиль и положение и не использует их из указанного фрейма.

```
<define name="draw-frame-attlist" combine="interleave">
    <optional>
        <attribute name="draw:copy-of">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

9.3.1 Текстовый блок

Элемент `<draw:text-box>` представляет текстовый блок. Текстовый блок может быть использован для помещения текста в контейнер, который находится за пределами текстовой полосы документа.

```
<define name="draw-text-box">
    <element name="draw:text-box">
        <ref name="draw-text-box-attlist"/>
        <zeroOrMore>
            <ref name="text-content"/>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<draw:text-box>` могут быть связаны следующие атрибуты:

- сцепка;
- скругленные вершины;
- минимальные высота и ширина;
- максимальные высота и ширина.

Текстовые блоки не поддерживают контуры, описанные в разделе 9.3.8, и альтернативный текст, описанный в разделе 9.3.9.

Сцепка

Текстовые блоки могут быть сцеплены, другими словами, если содержимое текстового блока выходит за его пределы, содержимое перетекает в следующий текстовый блок в последовательности. Чтобы сцепить текстовые блоки, используется атрибут `draw:chain-next-name`. Значением этого атрибута является имя следующего текстового блока в цепочке. Сцепленные текстовые блоки обычно поддерживаются только текстовыми документами.

```
<define name="draw-text-box-attlist" combine="interleave">
    <optional>
        <attribute name="draw:chain-next-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Скругленные вершины

Атрибут `draw:corner-radius` определяет радиус окружности, используемой для скругления вершин текстового блока.

```
<define name="draw-text-box-attlist" combine="interleave">
    <optional>
        <attribute name="draw:corner-radius">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

Минимальные высота и ширина

Атрибуты `fo:min-height` и `fo:min-width` определяют минимальную высоту или ширину для текстового блока. Если они существуют, то они перезаписывают высоту или ширину текстового блока, определенную атрибутами `svg:height` и `svg:width` окружающего элемента `<draw:frame>`. Их значения могут быть либо в единицах длины, либо в процентах. Если привязка текстового блока применена к ячейке таблицы, то процентное значение берется относительно окружающего табличного блока. Если привязка текстового блока применена к текстовому блоку, то процентное значение берется относительно

окружающего текстового блока. В остальных случаях процентное значение берется относительно высоты страницы или окна.

```
<define name="draw-text-box-attlist" combine="interleave">
    <optional>
        <attribute name="fo:min-height">
            <choice>
                <ref name="length"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:min-width">
            <choice>
                <ref name="length"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

Максимальные высота и ширина

Если ширина или высота текстового блока определена как минимальная ширина и высота (с использованием атрибутов `fo:min-width` или `fo:min-height`), то атрибуты `fo:max-width` и `fo:max-height` определяют максимальную ширину и высоту текстового блока. Когда достигаются эти максимальные значения, текстовый блок больше не увеличивается в размерах. Значения атрибутов могут быть либо в единицах длины, либо в процентах. Если привязка текстового блока применена к ячейке таблицы, то процентное значение берется относительно размеров окружающей ячейки таблицы. Если привязка текстового блока применена к текстовому блоку, то процентное значение берется относительно размеров окружающего текстового блока. В остальных случаях процентное значение берется относительно ширины или высоты страницы или окна.

```
<define name="draw-text-box-attlist" combine="interleave">
    <optional>
        <attribute name="fo:max-height">
            <choice>
                <ref name="length"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:max-width">
            <choice>
                <ref name="length"/>
```

```
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

9.3.2 Изображение

Элемент `<draw:image>` представляет изображение. Изображение может быть:

- содержащимся в документе как ссылка на внешний ресурс;
- внедренным в документ.

Этот элемент может быть [XLink], тогда случае элемент содержит некоторые атрибуты с фиксированными значениями, которые описывают семантику связи.

Поскольку данные изображения могут иметь произвольный формат, рекомендуется чтобы векторная графика сохранялась в формате [SVG], а растровая графика в формате [PNG].

```
<define name="draw-image">
    <element name="draw:image">
        <ref name="draw-image-attlist"/>
        <choice>
            <ref name="common-draw-data-attlist"/>
            <ref name="office-binary-data"/>
        </choice>
        <ref name="draw-text"/>
    </element>
</define>
```

С элементом `<draw:image>` могут быть связаны следующие атрибуты:

- данные изображения;
- имя фильтра.

Как и большинство других векторных фигур, фигуры изображения могут иметь текстовое содержимое. Оно отображается в дополнении к данным изображения.

Данные изображения

Данные изображения могут быть сохранены одним из следующих способов.

- Данные изображения хранятся во внешнем файле. Используется `xlink:href` и связанные атрибуты, описанные ниже, для связи с внешним файлом.

- Данные изображения хранятся в элементе `<draw:image>`. В этом случае элемент `<draw:image>` содержит элемент `<office:binary-data>`, который состоит из данных изображения в кодировке BASE64 (определенный в [RFC2045]). В этой ситуации атрибут `xlink:href` не требуется.

```
<define name="common-draw-data-attlist" combine="interleave">
    <group>
        <attribute name="xlink:href">
            <ref name="anyURI"/>
        </attribute>
        <optional>
            <attribute name="xlink:type" a:defaultValue="simple">
                <choice>
                    <value>simple</value>
                </choice>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:show" a:defaultValue="embed">
                <choice>
                    <value>embed</value>
                </choice>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:actuate" a:defaultValue="onLoad">
                <choice>
                    <value>onLoad</value>
                </choice>
            </attribute>
        </optional>
    </group>
</define>

<define name="office-binary-data">
    <element name="office:binary-data">
        <ref name="base64Binary"/>
    </element>
</define>
```

Имя фильтра

Если требуется, атрибут `draw:filter-name` может представлять имя фильтра изображения. Этот атрибут содержит имя внутреннего фильтра, который используется офисным приложением для загрузки графики.

```
<define name="draw-image-attlist" combine="interleave">
    <optional>
        <attribute name="draw:filter-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

9.3.3 Объекты

Документ в формате OpenDocument может содержать два типа объектов.

- Объекты, которые имеют представление OpenDocument. К ним относятся:
  - формулы (представленные как [MathML]);
  - диаграммы;
  - электронные таблицы;
  - текстовые документы;
  - графические изображения;
  - презентации.
- Объекты, которые не имеют XML-представления. Эти объекты имеют только бинарное представление. Примером таких объектов могут служить OLE-объекты (см. [OLE]).

Элемент `<draw:object>` представляет объект, который имеет XML-представление. Элемент `<draw:object-ole>` представляет объект который имеет только бинарное представление.

```
<define name="draw-object">
    <element name="draw:object">
        <ref name="draw-object-attlist"/>
        <choice>
            <ref name="common-draw-data-attlist"/>
            <ref name="office-document"/>
            <ref name="math-math"/>
        </choice>
    </element>
</define>

<define name="draw-object-ole">
    <element name="draw:object-ole">
        <ref name="draw-object-ole-attlist"/>
        <choice>
            <ref name="common-draw-data-attlist"/>
            <ref name="office-binary-data"/>
        </choice>
    </element>
</define>
```

С элементами `<draw:object>` и `<draw:object-ole>` могут быть связаны следующие атрибуты:

- данные объекта;
- уведомление об изменении таблицы;
- идентификатор класса.

Объекты не поддерживают преобразований, описанных в разделе 9.2.15.

Данные объекта

Данные объекта могут быть вызваны одним из следующих способов.

- Атрибут `xlink:href` ссылается на представление объекта, как описано ниже:
  - — для объектов, имеющих XML-представление, ссылка указывает на вложенный пакет объекта, причем объект, содержащийся внутри вложенной страницы, такой же, как если бы он содержался внутри документа;
  - — для объектов, не имеющих XML-представления, ссылка указывает на вложенный поток пакета, содержащего бинарное представление объекта.

  Приложения, поддерживающие объекты, должны поддерживать связь с объектами, которые содержатся внутри того же пакета. Они могут также поддерживать связь с объектами, расположенными за пределами пакета.
- Данные объекта содержатся в элементах `<draw:object>` или `<draw:object-ole>`:
  - — элемент `<draw:object>` содержит XML-представление объекта, например, элемент `<office:document>` или элемент `<math:math>`;
  - — элемент `<draw:object-ole>` содержит элемент `<office:binary-data>`, содержащий бинарные данные для объекта в кодировке BASE64.

  В этих ситуациях атрибут `xlink:href` не требуется.

Атрибут `xlink:href` описан в разделе 9.3.2.

Рекомендуется включать представление изображения объекта во фрейм, в дополнение к самому объекту.

Уведомление об изменении таблицы

Некоторые объекты, особенно диаграммы, могут требовать уведомления, когда таблица в документе изменяется. Для включения этих уведомлений используется атрибут `draw:notify-on-change-of-table`, который содержит имя таблицы. Этот атрибут может быть связан с элементом `<draw:object>`.

```
6807 <define name="draw-object-attlist" combine="interleave">
6808     <optional>
```

```
        <attribute name="draw:notify-on-update-of-ranges">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Идентификатор класса

Атрибут `draw:class-id` содержит идентификатор класса OLE-объекта (см. также [OLE]).

```
<define name="draw-object-ole-attlist" combine="interleave">
    <optional>
        <attribute name="draw:class-id"/>
    </optional>
</define>
```

9.3.4 Апплеты

Апплет — это небольшая программа на языке Java, включенная в документ. Элемент `<draw:applet>` основывается на теге `<applet>` из [HTML4]. Этот элемент должен содержать либо атрибут `draw:code`, либо атрибут `draw:object`.

```
<define name="draw-applet">
    <element name="draw:applet">
        <ref name="draw-applet-attlist"/>
        <optional>
            <ref name="common-draw-data-attlist"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-param"/>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<draw:applet>` могут быть связаны следующие атрибуты:

- кодовая база;
- код;
- объект;
- архив;
- разрешение скрипта.

Единственный элемент, который может быть включен в элемент `<draw:applet>`:

- параметр (см. раздел 9.3.6).

Апплеты не поддерживают преобразований, описанных в разделе 9.2.15.

Кодовая база

Кодовая база определяет базовый IRI для апплета. Если этот атрибут не определен, тогда по умолчанию устанавливается такой же IRI, как и для текущего документа. Кодовая база представляется атрибутами стандарта [XLink] `xlink:href`, `xlink:type`, `xlink:show` и `xlink:actuate`. Атрибут `xlink:href` описан в разделе 9.3.2.

Код

Атрибут `draw:code` определяет одно из нижеперечисленного:

- имя файла класса, который содержит скомпилированный подкласс апплета;
- путь к классу, включая сам файл класса.

Обязательно наличие либо этого атрибута, либо атрибута `draw:object`. Значение данного атрибута интерпретируется в зависимости от кодовой базы апплета.

```
<define name="draw-applet-attlist" combine="interleave">
    <optional>
        <attribute name="draw:code"/>
    </optional>
</define>
```

Объект

Атрибут `draw:object` определяет ресурс, который содержит сериализированное представление состояния апплета. Сериализированные данные содержат имя класса апплета, но не его реализацию. Значение данного атрибута интерпретируется в зависимости от кодовой базы апплета.

```
<define name="draw-applet-attlist" combine="interleave">
    <optional>
        <attribute name="draw:object"/>
    </optional>
</define>
```

Архив

Атрибут `draw:archive` определяет список разделенных запятыми адресов URL для архивов, содержащих классы и другие предварительно загружаемые ресурсы.

```
<define name="draw-applet-attlist" combine="interleave">
    <optional>
        <attribute name="draw:archive"/>
    </optional>
</define>
```

Разрешение скрипта

Атрибут `draw:mayscript` определяет, может ли апплет управляться скриптом.

```
<define name="draw-applet-attlist" combine="interleave">
    <optional>
        <attribute name="draw:may-script" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

9.3.5 Плагины

Плагины — это бинарные объекты, которые вставляются в документ для отображения медиа-содержимого, которое обычно не обрабатывается офисными приложениями самостоятельно. Плагины представлены элементом `<draw:plugin>`.

```
<define name="draw-plugin">
    <element name="draw:plugin">
        <ref name="draw-plugin-attlist"/>
        <ref name="common-draw-data-attlist"/>
        <zeroOrMore>
            <ref name="draw-param"/>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<draw:plugin>` могут быть связаны следующие атрибуты:

- тип MIME;
- источник.

Единственный элемент, который может содержаться внутри элемента `<draw:plugin>`, это:

- параметр (см. раздел 9.3.6).

Плагины не поддерживают преобразований, описанных в разделе 9.2.15.

Тип Mime

Атрибут `draw:mimetype` определяет тип MIME, с которым данный плагин будет зарегистрирован.

```
<define name="draw-plugin-attlist" combine="interleave">
    <optional>
        <attribute name="draw:mime-type"/>
    </optional>
</define>
```

Источник

Атрибуты [XLink] `xlink:href`, `xlink:type`, `xlink:show` и `xlink:actuate` определяют источник плагина. Атрибут `xlink:href` описан в разделе 9.3.2.

9.3.6 Параметры

Элемент `<draw:param>` содержит параметры, которые передаются апплету или плагину при инициализации.

```
<define name="draw-param">
    <element name="draw:param">
        <ref name="draw-param-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<draw:param>` могут быть связаны следующие атрибуты:

- имя;
- значение.

Имя

Атрибут `draw:name` определяет имя параметра времени выполнения.

```
<define name="draw-param-attlist" combine="interleave">
    <optional>
        <attribute name="draw:name"/>
    </optional>
</define>
```

Значение

Атрибут `draw:value` определяет значение параметра времени выполнения, заданного именем.

```
<define name="draw-param-attlist" combine="interleave">
    <optional>
        <attribute name="draw:value"/>
    </optional>
</define>
```

9.3.7 Плавающие фреймы

Плавающие фреймы — это фреймы, включенные в документ, которые могут содержать, например, текстовый документ или электронную таблицу. Плавающий фрейм представлен элементом `<draw:floating-frame>`.

```
<define name="draw-floating-frame">
    <element name="draw:floating-frame">
        <ref name="draw-floating-frame-attlist"/>
        <ref name="common-draw-data-attlist"/>
    </element>
</define>
```

С элементом `<draw:floating-frame>` могут быть связаны следующие атрибуты:

- источник;
- имя фрейма.

Плавающие фреймы не поддерживают преобразований, описанных в разделе 9.2.15.

Источник

Атрибуты [XLink] `xlink:href`, `xlink:type`, `xlink:show` и `xlink:actuate` определяют источник плавающего фрейма. Атрибут `xlink:href` описан в разделе 9.3.2.

Имя фрейма

Атрибут `draw:frame-name` определяет имя фрейма. Это имя может быть использовано как целевой фрейм внутри гиперссылки.

```
<define name="draw-floating-frame-attlist" combine="interleave">
    <optional>
        <attribute name="draw:frame-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

9.3.8 Контур

Элементы `<draw:contour-polygon>` и `<draw:contour-path>` могут содержаться внутри следующих элементов:

- `<draw:image>`
- `<draw:object>`

- `<draw:object-ole>`
- `<draw:applet>`
- `<draw:plugin>`
- `<draw:floating-frame>`

Эти элементы описывают контур изображения или объекта.

```
<define name="draw-contour-polygon">
    <element name="draw:contour-polygon">
        <ref name="common-contour-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-viewbox-attlist"/>
        <ref name="common-draw-points-attlist"/>
        <empty/>
    </element>
</define>

<define name="draw-contour-path">
    <element name="draw:contour-path">
        <ref name="common-contour-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-viewbox-attlist"/>
        <ref name="common-draw-path-data-attlist"/>
        <empty/>
    </element>
</define>
```

Элементы аналогичны элементам `<draw:polygon>` (см. раздел 9.2.4) и `<draw:path>` (см. раздел 9.2.6) за исключением того, что они определяют контур, а не векторную фигуру. Поддерживаемые ими атрибуты совпадают для размера, окна просмотра, точек (только для контура-многоугольника) и траектории (только для контура-траектории).

В отличие от остальных элементов, атрибуты `svg:width` и `svg:height` могут иметь значение длины в пикселях (т. е. 20 px) (так же, как традиционно, например, 2 см).

Перерисовка при редактировании

Атрибут `draw:recreate-on-edit` определяет, будет ли контур изображения или объекта автоматически перерисовываться, когда изображение или объект редактируется.

```
<define name="common-contour-attlist" combine="interleave">
    <attribute name="draw:recreate-on-edit">
        <ref name="boolean"/>
    </attribute>
</define>
```

9.3.9 Альтернативный текст

Элемент `<svg:desc>` определяет альтернативный текст, как описано в § 5.4 [SVG]. Он может использоваться со следующими элементами:

- `<draw:image>`
- `<draw:object>`
- `<draw:object-ole>`
- `<draw:applet>`
- `<draw:floating-frame>`
- `<draw:plugin>`
- `<draw:object-ole>`

```
<define name="svg-desc">
    <element name="svg:desc">
        <text/>
    </element>
</define>
```

9.3.10 Гиперссылки

Фреймы могут вести себя как гиперссылки. Такие гиперссылки представлены элементом `<draw:a>`, где содержимым элемента является фрейм, содержащий источник ссылки.

Этот элемент является [XLink] и имеет некоторые атрибуты с фиксированными значениями и описывает семантику ссылки.

```
<define name="draw-a">
    <element name="draw:a">
        <ref name="draw-a-attlist"/>
        <ref name="draw-frame"/>
    </element>
</define>
```

С элементом `<draw:a>` могут быть связаны следующие атрибуты:

- адрес ссылки;
- целевой фрейм ссылки;
- имя;
- серверное сенсорное изображение.

Адрес ссылки

Атрибут `xlink:href` определяет целевой адрес ссылки.

```
6930 <define name="draw-a-attlist" combine="interleave">
6931     <attribute name="xlink:href">
6932         <ref name="anyURI"/>
6933     </attribute>
6934     <optional>
6935         <attribute name="xlink:type" a:defaultValue="simple">
6936             <value>simple</value>
6937         </attribute>
6938     </optional>
6939     <optional>
6940         <attribute name="xlink:actuate" a:defaultValue="onRequest">
6941             <choice>
6942                 <value>onRequest</value>
6943             </choice>
6944         </attribute>
6945     </optional>
6946 </define>
```

Целевой фрейм ссылки

Атрибут `office:target-frame` определяет целевой фрейм ссылки.

Атрибут может принимать одно из следующих значений:

- `_self` — документ по ссылке заменяет содержимое текущего фрейма;
- `_blank` — документ по ссылке открывается в новом фрейме;
- `_parent` — документ по ссылке открывается в родительском по отношению к текущему фрейму фрейме;
- `_top` — документ по ссылке отображается в самом верхнем фрейме, это фрейм, который содержит текущей фрейм в качестве дочернего, но который не содержится внутри никакого другого фрейма;
- имя фрейма — документ по ссылке отображается в заданном фрейме. Если фрейм с таким именем отсутствует, будет создан новый фрейм с заданным именем.

Для соответствия спецификации [XLink] добавлен дополнительный атрибут `xlink:show` для элемента `<draw:a>`. Если значением атрибута `xlink:show` является `_blank`, то значением атрибута `xlink:show` будет `new`. Если значением атрибута `xlink:show` является одно из других возможных, значением атрибута `xlink:show` будет `replace`.

```
6947 <define name="draw-a-attlist" combine="interleave">
6948     <optional>
6949         <attribute name="office:target-frame-name">
6950             <ref name="targetFrameName"/>
6951         </attribute>
6952     </optional>
6953     <optional>
6954         <attribute name="xlink:show">
```

```
            <choice>
                <value>new</value>
                <value>replace</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Имя

Гиперссылка может иметь имя, но оно не является обязательным. Атрибут `office:name` определяет имя ссылки. Имя может служить целевым объектом для других гиперссылок. Имя не должно быть уникальным.

Этот атрибут определен только для совместимости со стандартом [HTML4], в котором элемент `<a>` может выступать и как источник ссылки, и как целевой объект одновременно. Мы настоятельно рекомендуем не использовать этот атрибут для каких-либо целей, кроме представления гиперссылки в оригинальном виде, как она представлена в HTML-документе.

```
<define name="draw-a-attlist" combine="interleave">
    <optional>
        <attribute name="office:name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Серверное сенсорное изображение

Ссылка может быть серверным сенсорным изображением. Если указан атрибут `office:server-map`, координаты курсора мыши в позиции нажатия графической фигуры добавляются к IRI ссылки. Координаты могут использоваться сервером для определения, какую именно ссылку внутри сенсорного изображения активировать.

```
<define name="draw-a-attlist" combine="interleave">
    <optional>
        <attribute name="office:server-map" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

9.3.11 Клиентские сенсорные изображения

Клиентские сенсорные изображения — это коллекция гиперссылок, связанных с графическими элементами. Сенсорное изображение — это

последовательность элементов сенсорных изображений. Каждый элемент сенсорного изображения связывает гиперссылку с определенной областью. Эта область может быть одной из следующих форм:

- прямоугольник;
- круг;
- многоугольник.

Элемент `<draw:image-map>` представляет сенсорное изображение.

```
<define name="draw-image-map">
    <element name="draw:image-map">
        <zeroOrMore>
            <choice>
                <ref name="draw-area-rectangle"/>
                <ref name="draw-area-circle"/>
                <ref name="draw-area-polygon"/>
            </choice>
        </zeroOrMore>
    </element>
</define>
```

Элемент `<draw:image-map>` может содержать три типа элементов сенсорных изображений, которые представляют три типа областей сенсорных изображений:

- прямоугольные элементы сенсорного изображения;
- круглые элементы сенсорного изображения;
- многоугольные элементы сенсорного изображения.

Элементы сенсорных изображений описываются в терминах абсолютного положения. Офисное приложение при загрузке XML-файла должно отобразить сенсорное изображение в связанный с ним графический элемент, например, изображение в оригинальном размере. Затем приложение должно масштабировать сенсорное изображение таким образом, чтобы оно соответствовало текущему размеру изображения, но в формате файла всегда сохраняется в немасштабируемом виде, приведенном в соответствие с размерами немасштабированного изображения.

Прямоугольные области сенсорного изображения

Элемент `<draw:area-rectangle>` описывает прямоугольную область сенсорного изображения путем задания положения x, y (атрибуты `svg:x` и `svg:y`), а также ширины и высоты (атрибуты `svg:width` и `svg:height`). Это обязательные атрибуты. В дополнение к этому, атрибуты, описанные в

нижеследующем разделе *Общие атрибуты и элементы сенсорных изображений,* устанавливаются по выбору.

```
<define name="draw-area-rectangle">
    <element name="draw:area-rectangle">
        <ref name="common-draw-area-attlist"/>
        <attribute name="svg:x">
            <ref name="coordinate"/>
        </attribute>
        <attribute name="svg:y">
            <ref name="coordinate"/>
        </attribute>
        <attribute name="svg:width">
            <ref name="length"/>
        </attribute>
        <attribute name="svg:height">
            <ref name="length"/>
        </attribute>
        <optional>
            <ref name="svg-desc"/>
        </optional>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
    </element>
</define>
```

Круглые области сенсорного изображения

Элемент `<draw:area-circle>` описывает круглую область сенсорного изображения. Дополнительные атрибуты для круглых областей сенсорного изображения описаны ниже, в разделе общих атрибутов.

Обязательные атрибуты `svg:cx` и `svg:cy` определяют центр окружности. Обязательный атрибут `svg:r` определяет радиус окружности.

Атрибуты, описанные в разделе *Общие атрибуты и элементы сенсорных изображений,* являются необязательными.

```
<define name="draw-area-circle">
    <element name="draw:area-circle">
        <ref name="common-draw-area-attlist"/>
        <attribute name="svg:cx">
            <ref name="coordinate"/>
        </attribute>
        <attribute name="svg:cy">
            <ref name="coordinate"/>
        </attribute>
        <attribute name="svg:r">
            <ref name="length"/>
        </attribute>
        <optional>
            <ref name="svg-desc"/>
        </optional>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
```

```
    </element>
</define>
```

Многоугольные области сенсорного изображения

Элемент `<draw:area-polygon>` описывает многоугольную область сенсорного изображения. Многоугольная область сенсорного изображения включает в себя следующие компоненты.

- **Обрамляющая рамка.**

  Обрамляющая рамка, которая представляется так же, как и прямоугольная область сенсорного изображения с использованием атрибутов `svg:x`, `svg:y`, `svg:width`, и `svg:height`, устанавливает фрейм, на который ссылаются область отображения и последовательность точек многоугольника. Указанный фрейм делает доступными координаты, которые будут преобразовываться в абсолютные координаты.

- **Область отображения.**

  Атрибут области отображения `svg:viewBox` устанавливает систему координат для последовательности точек. Область отображения позволяет избежать необходимость записывать каждую точку из последовательности абсолютными координатами с длинами и единицами измерения.

- **Последовательность точек** в координатах области отображения, указанная в атрибуте `svg:points`.

Для получения дополнительной информации о задании многоугольника см. раздел 9.2.4.

Описанные выше атрибуты являются обязательными. Атрибуты, описанные в разделе *Общие атрибуты и элементы сенсорных изображений,* являются необязательными.

```
<define name="draw-area-polygon">
    <element name="draw:area-polygon">
        <ref name="common-draw-area-attlist"/>
        <attribute name="svg:x">
            <ref name="coordinate"/>
        </attribute>
        <attribute name="svg:y">
            <ref name="coordinate"/>
        </attribute>
        <attribute name="svg:width">
```

```
            <ref name="length"/>
        </attribute>
        <attribute name="svg:height">
            <ref name="length"/>
        </attribute>
        <ref name="common-draw-viewbox-attlist"/>
        <ref name="common-draw-points-attlist"/>
        <optional>
            <ref name="svg-desc"/>
        </optional>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
    </element>
</define>
```

Пример – Многоугольная область сенсорного изображения.

Элемент, показанный в следующем примере, определяет треугольник, который расположен в центре изображения размером 2 см на 2 см. Обрамляющая рамка покрывает область размером 2 см на 1,5 см. Одна единица области отображения соответствует 0,01 мм.

```
<draw:area-polygon ...
    svg:x="0" svg:y="0" svg:width="2.0cm" svg:height="2.0cm"
    svg:viewBox="0 0 2000 2000"
    svg:points="400,1500 1600,1500 1000,400"/>
```

Общие атрибуты и элементы сенсорных изображений

В дополнение к атрибутам векторной фигуры, каждый элемент сенсорного изображения может содержать следующую информацию:

- ссылка, включающая IRI и целевой фрейм ссылки;
- имя;
- флаг активности;
- описание (используется дочерний элемент `<svg:desc>`, как описано в разделе 9.3.9);
- события, связанные с областью (используется дочерний элемент `<office:event-listeners>`, как описано в разделе 12.4).

Остальные атрибуты сенсорных изображений взяты из HTML-представления сенсорного изображения.

Каждый элемент сенсорного изображения идентифицирует гиперссылку и использует атрибуты стандарта [XLink] `href`, `type`, и `show`, а также атрибут `office:target-frame-name` для описания ссылки.

```
<define name="common-draw-area-attlist" combine="interleave">
    <optional>
```

```
7057        <attribute name="xlink:href">
7058            <ref name="anyURI"/>
7059        </attribute>
7060    </optional>
7061    <optional>
7062        <attribute name="xlink:type" a:defaultValue="simple">
7063            <choice>
7064                <value>simple</value>
7065            </choice>
7066        </attribute>
7067    </optional>
7068    <optional>
7069        <attribute name="office:target-frame-name">
7070            <ref name="targetFrameName"/>
7071        </attribute>
7072    </optional>
7073    <optional>
7074        <attribute name="xlink:show">
7075            <choice>
7076                <value>new</value>
7077                <value>replace</value>
7078            </choice>
7079            </attribute>
7080    </optional>
7081 </define>
```

Атрибут `office:name` присваивает имя каждому элементу сенсорного изображения.

```
7082 <define name="common-draw-area-attlist" combine="interleave">
7083     <optional>
7084         <attribute name="office:name">
7085             <ref name="string"/>
7086         </attribute>
7087     </optional>
7088 </define>
```

Атрибут `draw:nohref` объявляет о том, что элемент сенсорного изображения и связанная с ним область не активна. Содержащийся внутри элемента сенсорного изображения IRI не используется.

```
7089 <define name="common-draw-area-attlist" combine="interleave">
7090     <optional>
7091         <attribute name="draw:nohref">
7092             <choice>
7093                 <value>nohref</value>
7094             </choice>
7095         </attribute>
7096     </optional>
7097 </define>
```

9.4 Трехмерные фигуры

9.4.1 Сцена

Элемент `<dr3d:scene>` является единственным элементом, который может содержать трехмерные фигуры. Сцена подобна группе, но кроме того она устанавливает проекцию, освещение и другие детали формирования изображения для фигур внутри сцены.

```
<define name="dr3d-scene">
    <element name="dr3d:scene">
        <ref name="dr3d-scene-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-style-name-attlist"/>
        <ref name="common-draw-z-index-attlist"/>
        <ref name="common-draw-id-attlist"/>
        <ref name="common-draw-layer-name-attlist"/>
        <ref name="common-text-spreadsheet-shape-attlist"/>
        <ref name="common-dr3d-transform-attlist"/>
        <zeroOrMore>
            <ref name="dr3d-light"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="shapes3d"/>
        </zeroOrMore>
    </element>
</define>

<define name="shapes3d">
    <choice>
        <ref name="dr3d-scene"/>
        <ref name="dr3d-extrude"/>
        <ref name="dr3d-sphere"/>
        <ref name="dr3d-rotate"/>
        <ref name="dr3d-cube"/>
    </choice>
</define>
```

С данным элементом могут быть связаны следующие атрибуты:

- положение, размер, стиль, слой, Z-индекс, идентификатор (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- векторы камеры;
- проекция;
- дистанция;
- фокусное расстояние;
- наклон тени;
- режим затенения;

- окружающий свет;
- режим освещения.

Векторы камеры

Векторы камеры определяют видимый объем. Атрибут `dr3d:vrp` определяет начало координат, атрибут `dr3d:vpn` указывает направление к строящемуся объекту, атрибут `dr3d:vup` определяет верхний вектор.

```
<define name="dr3d-scene-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:vrp">
            <ref name="vector3D"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="dr3d:vpn">
            <ref name="vector3D"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="dr3d:vup">
            <ref name="vector3D"/>
        </attribute>
    </optional>
</define>
```

Проекция

Атрибут `dr3d:projection` определяет проекцию. Проекция может быть аксонометрической или параллельной. В аксонометрическом режиме объекты выглядят меньше на расстоянии.

```
<define name="dr3d-scene-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:projection">
            <choice>
                <value>parallel</value>
                <value>perspective</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Дистанция

Атрибут `dr3d:distance` определяет расстояние между камерой и объектом.

```
<define name="dr3d-scene-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:distance">
            <ref name="length"/>
        </attribute>
```

```
    </optional>
</define>
```

Фокусное расстояние

Атрибут `dr3d:focal-length` определяет фокусное расстояние для виртуальной камеры данной сцены.

```
<define name="dr3d-scene-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:focal-length">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

Наклон тени

Атрибут `dr3d:shadow-slant` определяет угол между трехмерной сценой и виртуальной поверхностью, на которую падает тень.

```
<define name="dr3d-scene-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:shadow-slant">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Режим затенения

Режим затенения определяет, каким образом рассчитывается освещение при прорисовке поверхностей

- `flat` — освещение рассчитывается по одной нормали поверхности;
- `phong` — освещение рассчитывается путем интерполирования нормалей поверхности по всей поверхности;
- `gouraud` — освещение рассчитывается путем интерполирования цвета с нормалями поверхности на каждой грани;
- `draft` — поверхность не освещается и отображается только ее каркас.

```
<define name="dr3d-scene-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:shade-mode">
            <choice>
                <value>flat</value>
                <value>phong</value>
                <value>gouraud</value>
                <value>draft</value>
            </choice>
        </attribute>
```

```
    </optional>
</define>
```

Окружающий свет

Атрибут `dr3d:ambient-color` определяет цвет окружающего освещения. Окружающее освещение — это освещение, которое выглядит как исходящее со всех направлений.

```
<define name="dr3d-scene-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:ambient-color">
            <ref name="color"/>
        </attribute>
    </optional>
</define>
```

Режим освещения

Атрибут `dr3d:lighting-mode` включает или выключает использование света на трехмерной сцене.

```
<define name="dr3d-scene-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:lighting-mode">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Трехмерное преобразование

Значением атрибута `dr3d:transform` является список определений преобразований, которые применяются в том порядке, в котором они перечислены в списке. Одиночные определения преобразований разделяются пробельными символами. Допустимые типы определений преобразований включают:

- `matrix (<a> <b> <c> <d> <e> <f> <g> <h> <i> <j> <k> <l>)` — определяет преобразование в виде матрицы преобразований шести значений. matrix(a,b,c,d,e,f,g,h,i,j,k,l) эквивалентно применению матрицы преобразований [a b c d e f g h i j k l];
- `translate (<tx> <ty> <tz>)` — определяет перемещение на tx, ty и tz;
- `scale (<sx> <sy> <sz>)` — определяет операцию масштабирования на sx, sy и sz;

- `rotatex (<rotate-angle>)` — определяет вращение на <rotate-angle> градусов вокруг оси x;
- `rotatey (<rotate-angle>)`, который определяет вращение на <rotate-angle> градусов вокруг оси y.
- `rotatez (<rotate-angle>)` — определяет вращение на <rotate-angle> градусов вокруг оси z.

```
<define name="common-dr3d-transform-attlist">
    <optional>
        <attribute name="dr3d:transform"/>
    </optional>
</define>
```

9.4.2 Источник света

Элемент `<dr3d:light>` представляет источник света в сцене.

Этот элемент должен быть первым элементом, содержащимся в элементе `<dr3d:scene>`. Может быть указано несколько источников света, но приложения могут поддерживать только ограниченное их количество внутри одной сцены. Обычное ограничение — 8 источников света на одну сцену.

```
<define name="dr3d-light">
    <element name="dr3d:light">
        <ref name="dr3d-light-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<dr3d:light>` могут быть связаны следующие атрибуты:

- рассеянный;
- направление;
- включенный;
- отраженный.

Рассеянный

Атрибут `dr3d:diffuse-color` определяет базовый цвет распространяемого света.

```
<define name="dr3d-light-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:diffuse-color">
            <ref name="color"/>
        </attribute>
    </optional>
</define>
```

Направление

Атрибут `dr3d:direction` определяет направление, в котором распространяется свет.

```
<define name="dr3d-light-attlist" combine="interleave">
    <attribute name="dr3d:direction">
        <ref name="vector3D"/>
    </attribute>
</define>
```

Включенный

Атрибут `dr3d:enabled` определяет, включен или нет источник света. Если источник света не включен, он не распространяет никакого света.

```
<define name="dr3d-light-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:enabled">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Отраженный

Атрибут `dr3d:specular` определяет, будет ли свет от источника создавать отражение на объектах. Приложения могут рассчитать этот атрибут только для первого источника в сцене.

```
<define name="dr3d-light-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:specular">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

9.4.3 Куб

Элемент `<dr3d:cube>` представляет трехмерную кубическую фигуру.

```
<define name="dr3d-cube">
    <element name="dr3d:cube">
        <ref name="dr3d-cube-attlist"/>
        <ref name="common-draw-z-index-attlist"/>
        <ref name="common-draw-id-attlist"/>
        <ref name="common-draw-layer-name-attlist"/>
        <ref name="common-draw-style-name-attlist"/>
        <ref name="common-dr3d-transform-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<dr3d:cube>` могут быть связаны следующие атрибуты:

- стиль, слой, Z-индекс и идентификатор (см. раздел 9.2.15);
- минимальная и максимальная грань.

Минимальная и максимальная грань

Атрибуты `dr3d:min-edge` и `dr3d:max-edge` определяют минимальную и максимальную грань куба в трехмерном пространстве.

```
<define name="dr3d-cube-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:min-edge">
            <ref name="vector3D"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="dr3d:max-edge">
            <ref name="vector3D"/>
        </attribute>
    </optional>
</define>
```

9.4.4 Сфера

Элемент `<dr3d:sphere>` представляет трехмерную сферическую фигуру.

```
<define name="dr3d-sphere">
    <element name="dr3d:sphere">
        <ref name="dr3d-sphere-attlist"/>
        <ref name="common-draw-z-index-attlist"/>
        <ref name="common-draw-id-attlist"/>
        <ref name="common-draw-layer-name-attlist"/>
        <ref name="common-draw-style-name-attlist"/>
        <ref name="common-dr3d-transform-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<dr3d:sphere>` могут быть связаны следующие атрибуты:

- стиль, слой, Z-индекс и идентификатор (см. раздел 9.2.15);
- центр;
- размер.

Центр

Атрибут `dr3d:center` определяет центр сферы в трехмерном пространстве.

```
<define name="dr3d-sphere-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:center">
            <ref name="vector3D"/>
        </attribute>
```

```
    </optional>
</define>
```

Размер

Атрибут `dr3d:size` определяет размер сферы в трехмерном пространстве.

```
<define name="dr3d-sphere-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:size">
            <ref name="vector3D"/>
        </attribute>
    </optional>
</define>
```

9.4.5 Выдавливание

Элемент `<dr3d:extrude>` представляет трехмерное выдавливание, основанное на многоугольнике.

```
<define name="dr3d-extrude">
    <element name="dr3d:extrude">
        <ref name="common-draw-path-data-attlist"/>
        <ref name="common-draw-viewbox-attlist"/>
        <ref name="common-draw-id-attlist"/>
        <ref name="common-draw-z-index-attlist"/>
        <ref name="common-draw-layer-name-attlist"/>
        <ref name="common-draw-style-name-attlist"/>
        <ref name="common-dr3d-transform-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<dr3d:extrude>` могут быть связаны следующие атрибуты:

- область отображения, стиль, слой, Z-индекс, идентификатор (см. раздел 9.2.15);
- данные траектории (см. раздел 9.2.6).

9.4.6 Вращение

Элемент `<dr3d:rotate>` представляет трехмерную фигуру вращения, основанную на многоугольнике.

```
<define name="dr3d-rotate">
    <element name="dr3d:rotate">
        <ref name="common-draw-viewbox-attlist"/>
        <ref name="common-draw-path-data-attlist"/>
        <ref name="common-draw-z-index-attlist"/>
        <ref name="common-draw-id-attlist"/>
        <ref name="common-draw-layer-name-attlist"/>
        <ref name="common-draw-style-name-attlist"/>
        <ref name="common-dr3d-transform-attlist"/>
        <empty/>
```

```
    </element>
</define>
```

С элементом `<dr3d:rotate>` могут быть связаны следующие атрибуты:

- область отображения, стиль, слой, Z-индекс, идентификатор (см. раздел 9.2.15);
- данные траектории (см. раздел 9.2.6).

9.5 Специальные фигуры

Элемент `<draw:custom-shape>` представляет векторную фигуру, допускающую прорисовку множества образов. Элемент предлагает работу со шрифтом и выдавливанием. Специальная фигура может иметь геометрию, влияющую на ее векторные фигуры. Эта геометрия может быть отображена в пользовательском интерфейсе офисных приложений, например, отображением интерактивных обработчиков, представляющих простой путь изменения геометрии.

```
<define name="draw-custom-shape">
    <element name="draw:custom-shape">
        <ref name="draw-custom-shape-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <ref name="draw-glue-point"/>
        </zeroOrMore>
        <ref name="draw-text"/>
        <optional>
            <ref name="draw-enhanced-geometry"/>
        </optional>
    </element>
</define>
```

С элементом `<draw:custom shape>` могут быть связаны следующие атрибуты:

- положение, размер, стиль, слой, Z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- механизм рисования;
- данные рисования.

Механизм рисования

Необязательный атрибут draw:engine определяет имя механизма отображения, который будет использоваться для построения изображения специфической фигуры. Значением атрибута является токен пространства имен, имеющий префикс пространства имен XML, как и любой атрибут или элемент этой спецификации. Механизм рисования может либо получать данные из атрибута draw:data, либо вычислять дочерний элемент <draw:enhanced-geometry>.

Если атрибут draw:engine пропущен, будет использован стандартный механизм рисования офисного приложения для отображения специальных фигур. Этот механизм получает данные о геометрии только от элемента <draw:enhanced-geometry>.

```
7328 <define name="draw-custom-shape-attlist" combine="interleave">
7329     <optional>
7330         <attribute name="draw:engine">
7331             <ref name="namespacedToken"/>
7332         </attribute>
7333     </optional>
7334 </define>
```

Данные рисования

Атрибут draw:data содержит специфические данные механизма прорисовки, которые описывают геометрию специфической фигуры. Этот атрибут вычисляется только в случае установки в атрибуте draw:engine значения не по умолчанию.

```
7335 <define name="draw-custom-shape-attlist" combine="interleave">
7336     <optional>
7337         <attribute name="draw:data">
7338             <ref name="string"/>
7339         </attribute>
7340     </optional>
7341 </define>
```

9.5.1 Расширенная геометрия

Элемент <draw:enhanced-geometry> содержит геометрию для элемента <draw:custom-shape> в том случае, если его атрибут draw:engine пропущен.

```
7342 <define name="draw-enhanced-geometry">
7343     <element name="draw:enhanced-geometry">
7344         <ref name="draw-enhanced-geometry-attlist"/>
7345         <zeroOrMore>
7346             <ref name="draw-equation"/>
7347         </zeroOrMore>
```

```
        <zeroOrMore>
            <ref name="draw-handle"/>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<draw:enhanced-geometry>` могут быть связаны следующие атрибуты:

- тип;
- область отображение;
- отражение;
- угол поворота текста;
- допустимость выдавливания;
- допустимость текстовой траектории;
- допустимость концентрической градиентной заливки;
- расширенная геометрия: атрибуты выдавливания (см. раздел 9.5.2);
- расширенная геометрия: атрибуты траектории (см. раздел 9.5.3);
- расширенная геометрия: атрибуты траектории текста (см. раздел 9.5.4);
- расширенная геометрия: уравнение (см. раздел 9.5.5);
- расширенная геометрия: атрибуты меток-манипуляторов (см. раздел 9.5.6).

Тип

Атрибут `draw:type` содержит имя типа фигуры. Это имя может быть использовано для предоставления специализированного пользовательского интерфейса для определенного типа, например для стрелок, смайлов и т.п.

Тип фигуры зависит от механизма рисования и никак не влияет на геометрию фигуры. Если значением атрибута `draw:type` является `non-primitive`, то тип фигуры не определен.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:type" a:defaultValue="non-primitive">
            <ref name="custom-shape-type"/>
        </attribute>
    </optional>
</define>

<define name="custom-shape-type">
    <choice>
        <value>non-primitive</value>
        <ref name="string"/>
```

```
    </choice>
</define>
```

Область отображения

Атрибут `svg:viewBox` устанавливает пользовательскую систему координат внутри физической системы координат фигуры, заданной путем определения атрибутов положения и размера. Пользовательская система координат используется элементом `<draw:enhanced-path>`.

Синтаксис использования данного атрибута такой же, как и синтаксис стандарта [SVG]. Значениями данного атрибута являются четыре числа, разделенные пробельными символами. Они определяют левое, верхнее, правое и нижние измерения пользовательской системы координат.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="svg:viewBox">
            <list>
                <ref name="integer"/>
                <ref name="integer"/>
                <ref name="integer"/>
                <ref name="integer"/>
            </list>
        </attribute>
    </optional>
</define>
```

Отражение

Атрибуты `draw:mirror-vertical` и `draw:mirror-horizontal` определяют, будет ли геометрия фигуры отображаться зеркально.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:mirror-vertical" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="draw:mirror-horizontal" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Угол поворота текста

Атрибут draw:text-rotate-angle определяет угол, на который будет повернут текст внутри специальной фигуры, в дополнение к вращению, включенному в атрибут draw:transform.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:text-rotate-angle" a:defaultValue="0">
            <ref name="double"/>
        </attribute>
    </optional>
</define>
```

Допустимость выдавливания

Атрибут draw:extrusion-allowed определяет, будет ли у фигуры возможность отображаться как объекту выдавливания.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-allowed" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Допустимость текстовой траектории

Атрибут draw:text-path-allowed определяет, может ли фигура отображаться как объект текстовых эффектов. Текст объекта текстовых эффектов отличается от обычных текстовых объектов возможностью отображать текст вдоль или между линий, определенных атрибутом draw:enhanced-path. Объекты текстовых эффектов способны поддерживать стандартные графические атрибуты, такие как заливка, тень или стиль линий.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:text-path-allowed" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Допустимость концентрической градиентной заливки

Атрибут draw:concentric-gradient-fill-allowed определяет, может ли фигура отображаться с концентрической градиентной заливкой, использующей траекторию специальной фигуры.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:concentric-gradient-fill-allowed"
                   a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

9.5.2 Расширенная геометрия: атрибуты выдавливания

Выдавливание

Атрибут `draw:extrusion` определяет, будет ли отображаться выдавливание.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Яркость выдавливания

Атрибут `draw:extrusion-brightness` определяет яркость сцены.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-brightness" a:defaultValue="33%">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

Глубина выдавливания

Атрибут `draw:extrusion-depth` определяет глубину выдавливания. Он берет два значения, разделенных пробельными символами. Первое значение определяет глубину выдавливания, второе значение определяет фрагмент выдавливания, лежащий перед фигурой. Он должен быть в диапазоне [0,1]. Значение 0 устанавливается по умолчанию.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-depth" a:defaultValue="36pt 0">
            <list>
                <ref name="length"/>
                <ref name="double"/>
            </list>
        </attribute>
```

```
    </optional>
</define>
```

Диффузия выдавливания

Величина диффузии, отражаемой фигурой, определяется атрибутом `draw:extrusion-diffusion`.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-diffusion" a:defaultValue="0%">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

Количество сегментов линии выдавливания

Атрибут `draw:extrusion-number-of-line-segments` определяет количество сегментов линии, которые будут использоваться для отображения кривых поверхностей. Чем больше число, тем больше сегментов используется.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-number-of-line-segments"
                   a:defaultValue="30">
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

Фронтальное освещение выдавливания

Атрибут `draw:extrusion-light-face` определяет, реагирует ли передняя грань выдавливания на изменение освещения.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-light-face" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Резкость первого источника света выдавливания

Атрибут `draw:extrusion-first-light-harsh` определяет, является ли первый источник света резким.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-first-light-harsh"
```

```
                    a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Резкость второго источника света выдавливания

Атрибут `draw:extrusion-second-light-harsh` определяет, является ли второй источник света резким.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-second-light-harsh"
                    a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Интенсивность первого источника света выдавливания

Атрибут `draw:extrusion-first-light-level` определяет интенсивность первого источника света.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-first-light-level"
                    a:defaultValue="66%">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

Интенсивность второго источника света выдавливания

Атрибут `draw:extrusion-second-light-level` определяет интенсивность второго источника света.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-second-light-level"
                    a:defaultValue="66%">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

Направление первого источника света выдавливания

Атрибут `draw:extrusion-first-light-direction` определяет направление первого источника света.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-first-light-direction"
                   a:defaultValue="(5 0 1)">
            <ref name="vector3D"/>
        </attribute>
    </optional>
</define>
```

Направление второго источника света выдавливания

Атрибут `draw:extrusion-second-light-direction` определяет направление второго источника света.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-second-light-direction"
                   a:defaultValue="(-5 0 1)">
            <ref name="vector3D"/>
        </attribute>
    </optional>
</define>
```

Металлическое выдавливание

Атрибут `draw:extrusion-metal` определяет, будет ли поверхность выдавливания выглядеть как металлическая.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-metal" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Режим затенения выдавливания

Атрибут `dr3d:shade-mode` определяет, каким образом будет рассчитываться освещение при прорисовке поверхностей:

- `flat` — освещение рассчитывается по одной нормали поверхности;
- `phong` — освещение рассчитывается путем интерполяции нормалей поверхности по всей поверхности;
- `gouraud` — освещение рассчитывается путем интерполяции цвета с нормалями поверхности на каждой грани;
- `draft` — поверхность не освещается и отображается только ее каркас.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:shade-mode" a:defaultValue="flat">
```

```
            <choice>
                <value>flat</value>
                <value>phong</value>
                <value>gouraud</value>
                <value>draft</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Угол вращения выдавливания

Первое значение атрибута `draw:extrusion-rotation-angle` определяет вращение вокруг оси x. Второе значение атрибута `draw:extrusion-rotation-angle` определяет вращение вокруг оси y. Вращение вокруг оси z определяется углом вращения атрибута `draw:transform`.

Порядок вращения: вокруг оси z, вокруг оси y, затем вокруг оси x.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-rotation-angle" a:defaultValue="0 0">
            <list>
                <ref name="double"/>
                <ref name="double"/>
            </list>
        </attribute>
    </optional>
</define>
```

Центр вращения выдавливания

Атрибут `draw:extrusion-rotation-center` определяет положение центральной точки вращения в терминах размеров фрагментов фигуры. Если этот атрибут пропущен, то используется геометрический центр фигуры.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-rotation-center">
            <ref name="vector3D"/>
        </attribute>
    </optional>
</define>
```

Глянцевость выдавливания

Атрибут `draw:extrusion-shininess` определяет глянцевость отражения.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-shininess" a:defaultValue="50%">
            <ref name="percent"/>
        </attribute>
    </optional>
```

```
</define>
```

Наклон выдавливания

Атрибут `draw:extrusion-skew` определяет величину наклона и угол наклона выдавливания. Настройки наклона применяются только в том случае, если значением атрибута `dr3d:projection` является `parallel`.

Первый параметр определяет величину наклона в процентах, второй параметр определяет угол наклона выдавливания.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-skew" a:defaultValue="50 45">
            <list>
                <ref name="double"/>
                <ref name="double"/>
            </list>
        </attribute>
    </optional>
</define>
```

Отражение выдавливания

Атрибут `draw:extrusion-specularity` определяет отражение объекта выдавливания.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-specularity" a:defaultValue="0%">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

Режим проецирования выдавливания

Атрибут `dr3d:projection` определяет режим проецирования: аксонометрический или параллельный.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:projection" a:defaultValue="parallel">
            <choice>
                <value>parallel</value>
                <value>perspective</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Точка обзора выдавливания

Атрибут `draw:extrusion-viewpoint` определяет точку обзора наблюдателя как точку в трехмерном пространстве. Синтаксис значения атрибута похож на синтаксис vector3D, только единица измерения следует за каждым параметром. Пример 3D-точки: (1cm 1cm 0m).

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-viewpoint"
                   a:defaultValue="3.5cm -3.5cm 25cm">
            <ref name="point3D"/>
        </attribute>
    </optional>
</define>

<define name="point3D">
    <data type="string"/>
</define>
```

Начало координат выдавливания

Атрибут `draw:extrusion-origin` определяет начало координат выдавливания внутри обрамляющей рамки в частях фигуры.

Первый параметр представляет точку отсчета по горизонтали, значение «минус 0.5» представляет левый край фигуры, значение «0» представляет центр фигуры, значение «0.5» представляет правый край фигуры.

Второй параметр представляет точку отсчета по вертикали, значение «минус 0.5» представляет верхний край фигуры, значение «0» представляет центр фигуры, значение «0.5» представляет нижний край фигуры.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-origin" a:defaultValue="0.5 -0.5">
            <list>
                <ref name="double"/>
                <ref name="double"/>
            </list>
        </attribute>
    </optional>
</define>
```

Цвет выдавливания

Атрибут `draw:extrusion-color` определяет, используется ли цвет выдавливания. Цвет выдавливания затем задается атрибутом `draw:secondary-fill-color`, определенном в графическом стиле специальной фигуры.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:extrusion-color" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

9.5.3 Расширенная геометрия: атрибуты траектории

Расширенная траектория

Атрибут `draw:enhanced-path` определяет траекторию аналогично атрибуту `svg:d` элемента `<svg:path>`. Инструкции, такие как `moveto`, `lineto`, `arcto` и другие, вместе с их параметрами описывают геометрию фигуры, которая может быть заполнена цветом и/или заштрихована. Относительные команды не поддерживаются.

Синтаксис атрибута `draw:enhanced-path` представлен ниже.

- Инструкции представляются в виде одного символа (например moveto представляется как `M`).
- Используется префиксная нотация, которая означает что за каждой командой следуют ее параметры.
- Излишние пробельные символы и разделители, такие как запятые, могут быть пропущены (например, `"M 10 10 L 20 20 L 30 20"` можно записать в виде: `"M10 10L20 20L30 20"` .
- Если команда повторяется несколько раз, требуется только первая команда (например `"M 10 10 L 20 20 L 30 20"` можно записать в виде `"M 10 10 L 20 20 30 20"`.
- Могут использоваться числа с плавающей точкой, поэтому единственный доступный десятичный разделитель — это точка («.»).

Приведенные выше правила совпадают с определенными для элемента `<svg:path>`.

Параметр может иметь одно из следующих расширений.

- Символ «?» для отметки начала имени формулы. В этом случае результат атрибута элемента `draw:formula` используется как значение параметра.

- Если символ «$» предшествует целому значению, значение является индексом атрибута `draw:modifiers`. В этом случае значение соответствующего модификатора используется как значение параметра.

Следующие обозначения используются в таблице ниже:

- () — группировка параметров;
- + — требуется 1 или более параметров.

***Пример** — для специальной фигуры, использующей draw:enhanced-path для описания круговой диаграммы, из которой убран сегмент правой верхней четверти:*

```
<draw:custom-shape
    svg:width="10cm" svg:height="10cm" svg:x="0cm" svg:y="0cm">
    <draw:enhanced-geometry svg:viewBox="0 0 10 10"
        draw:enhanced-path="V 0 0 10 10 10 5 5 0 L 5 5 Z N">
    </draw:enhanced-geometry>
</draw:custom-shape>
```

Поддерживаются следующие команды:

Команда	***Имя***	***Параметры***	***Описание***
M	moveto	(x y) +	Начинает новую вложенную траекторию с координат (x,y). Если за moveto следует множество пар координат, они интерпретируются как lineto
L	lineto	(x y) +	Рисует линию от текущей точки до точки (x, y). Если следует неоднократное указание пар координат, они все интерпретируются как lineto
C	curveto	(x1 y1 x2 y2 x y) +	Рисует кубическую кривую Безье от текущей точки к точке (x,y), используя (x1,y1) как контрольную точку в начале кривой и (x2,y2) как контрольную точку в конце кривой
Z	closepath	(none)	Закрывает текущую вложенную траекторию путем рисования прямой линии между текущей точкой и начальной точкой текущей вложенной траектории
N	endpath	(none)	Заканчивает текущее множество вложенных траекторий. Вложенные траектории будут закрашены с использованием eofill. Следующие вложенные траектории будут закрашиваться независимо
F	nofill	(none)	Определяет, что текущее множество вложенных траекторий не будет закрашено цветом
S	nostroke	(none)	Определяет, что текущее множество вложенных

Команда	Имя	Параметры	Описание
			траекторий не будет заштриховано
T	angle-ellipseto	(x y w h t0 t1) +	Рисует сегмент эллипса. Эллипс определяется его центром (x, y), размером(w, h), начальным углом t0 и конечным углом t1
U	angle-ellipse	(x y w h t0 t1) +	То же, что и команда «T», за исключением включенной moveto к стартовой точке
A	arcto	(x1 y1 x2 y2 x3 y3 x y) +	(x1, y1) и (x2, y2) определяют обрамляющую рамку эллипса. Линия рисуется от текущей точки к начальному углу дуги, который определяется радиус-вектором точки (x3, y3), а затем против часовой стрелки к конечному углу, определенному точкой (x4, y4)
B	arc	(x1 y1 x2 y2 x3 y3 x y) +	То же, что и команда «A», за исключением включенной moveto к стартовой точке
W	clockwise-arcto	(x1 y1 x2 y2 x3 y3 x y) +	То же, что и команда «A», за исключением того, что дуга рисуется по часовой стрелке
V	clockwise-arc	(x1 y1 x2 y2 x3 y3 x y)+	То же, что и команда «A», за исключением включенной moveto к стартовой точке и рисования дуги по часовой стрелке
X	elliptical-quatrantx	(x y) +	Рисует четверть эллипса, чей начальный сегмент направлен по касательной к оси y, рисуется от текущей точки до (x, y)
Y	elliptical-quadranty	(x y) +	Рисует четверть эллипса, чей начальный сегмент направлен по касательной к оси x, рисуется от текущей точки до (x, y)
Q	quadratic-curveto	(x1 y1 x y)+	Рисует квадратичную кривую Безье от текущей точки к (x, y), используя (x1, y1) как контрольную точку. (x, y) становится новой текущей точкой в конце выполнения команды

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:enhanced-path">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Точки растяжения траектории

Атрибуты `draw:path-stretchpoint-x` и `draw:path-stretchpoint-y` определяют точки растяжения фигуры.

```
7620 <define name="draw-enhanced-geometry-attlist" combine="interleave">
7621     <optional>
7622         <attribute name="draw:path-stretchpoint-x" a:defaultValue="0">
7623             <ref name="double"/>
7624         </attribute>
7625     </optional>
7626     <optional>
7627         <attribute name="draw:path-stretchpoint-y" a:defaultValue="0">
7628             <ref name="double"/>
7629         </attribute>
7630     </optional>
7631 </define>
```

Текстовые области

Атрибут `draw:text-areas` определяет список текстовых областей. Текстовые области используются для расположения и выравнивания текста. Если текстовая область не указана, то используется область самой фигуры. Если доступна вторая текстовая область, то она используется для вертикального текста.

Область состоит из четырех параметров:

первый параметр определяет левый край текстовой области;

второй параметр определяет верхний край текстовой области;

третий параметр определяет правый край текстовой области;

четвертый параметр определяет нижний край текстовой области.

Параметр может также иметь одно из расширений, приведенных ниже.

- Символ «?» используется для отметки начала имени формулы. В этом случае в качестве значения параметра используется значение атрибута элемента `draw:formula`.
- Если символ «$» предшествует целочисленному значению, значением является индексированный атрибут `draw:modifiers`. В этом случае значение соответствующего модификатора используется как значение параметра.

Пример —

атрибут `draw:text-areas`, определяющий две текстовых области, включающий модификатор и уравнение, может быть таким:

`draw:text-areas="0 0 100 100 ?Formula1 $1 200 200"`

```
7632 <define name="draw-enhanced-geometry-attlist" combine="interleave">
7633     <optional>
7634         <attribute name="draw:text-areas">
7635             <ref name="string"/>
```

```
        </attribute>
    </optional>
</define>
```

Точки привязки

Атрибут draw:glue-points определяет список определенных объектом точек привязки. В отличие от определяемых пользователем точек привязки, которые определяются во вложенном элементе <draw:glue-point>, определенные объектом точки привязки могут использовать уравнения и модификаторы:

первый параметр определяет горизонтальное положение точки привязки;

второй параметр определяет вертикальное положение точки привязки.

Каждый параметр может быть числом с плавающей точкой, а также может иметь одно из расширений приведенных ниже.

- Символ «?» используется для отметки начала имени формулы. В этом случае в качестве значения параметра используется значение атрибута элемента draw:formula.
- Если символ «$» предшествует целочисленному значению, значением является индексированный атрибут draw:modifiers. В этом случае значение соответствующего модификатора используется как значение параметра.

Пример —

атрибут draw:glue-points, определяющий две точки привязки, включающий модификатор и уравнение, может быть таким: draw:glue-points="0 ?Formula1 100 $1"

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:glue-points">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Тип точки привязки

Атрибут draw:glue-point-type определяет тип точки привязки. Если атрибут draw:glue-points также доступен, то данный атрибут игнорируется:

- none — точки привязки специального объекта отсутствуют;

- `segments` — соединительный элемент будет соединен с каждой точкой атрибута `draw:enhanced-path` ;
- `rectangle` — середина каждой стороны обрамляющего фигуру прямоугольника определяет точку привязки специального объекта.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:glue-point-type" a:defaultValue="none">
            <choice>
                <value>none</value>
                <value>segments</value>
                <value>rectangle</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Направление выхода точки привязки

Атрибут `draw:glue-point-leaving-directions` состоит из списка значений углов в градах, разделенных запятыми. Значение угла может быть числом с плавающей точкой. Позиция в списке имеет то же значение, что и указанная точка привязки атрибута `draw:glue-points`.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:glue-point-leaving-directions"/>
    </optional>
</define>
```

9.5.4 Расширенная геометрия: атрибуты траектории текста

Траектория текста

Атрибут `draw:text-path` определяет, будет ли использоваться траектория текста для отображения текста.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:text-path" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Режим траектории текста

Атрибут `draw:text-path-mode` определяет, как будет выводиться текст:

- `normal` — текст выводится вдоль траектории без масштабирования;

- path — текст согласовывается с траекторией;
- shape — текст согласовывается с обрамляющей рамкой фигуры.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:text-path-mode" a:defaultValue="normal">
            <choice>
                <value>normal</value>
                <value>path</value>
                <value>shape</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Масштабирование траектории текста

Атрибут draw:text-path-scale определяет масштабирование траектории текста.

- path — масштабирование текста определяется длиной траектории из атрибута draw:enhanced-path ;
- shape — масштабирование текста определяется шириной фигуры.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:text-path-scale" a:defaultValue="path">
            <choice>
                <value>path</value>
                <value>shape</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Одинаковая высота символов траектории текста

Атрибут draw:text-path-same-letter-heights определяет, будут ли все символы специальной фигуры иметь одинаковую высоту.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:text-path-same-letter-heights"
                   a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Модификаторы

Атрибут draw:modifiers содержит список значений модификатора. Модификатор может принимать дробное значение. В большинстве случаев

атрибут draw:modifiers используется атрибутом draw:handle-position для сохранения позиции метки-манипулятора.

```
<define name="draw-enhanced-geometry-attlist" combine="interleave">
    <optional>
        <attribute name="draw:modifiers">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

9.5.5 Расширенная геометрия: уравнение

Уравнение

Элемент <draw:equation> может быть использован в метках-манипуляторах, текстовых областях, точках привязки и расширенных траекториях для вычисления значений, которые зависят от значений модификатора. Поскольку значения модификатора могут интерактивно изменяться, это удобный способ интегрирования динамических значений в геометрию фигуры.

```
<define name="draw-equation">
    <element name="draw:equation">
        <ref name="draw-equation-attlist"/>
        <empty/>
    </element>
</define>
```

Имя

Атрибут draw:name определяет имя уравнения. В имени не допускаются пробелы.

```
<define name="draw-equation-attlist" combine="interleave">
    <optional>
        <attribute name="draw:name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Формула

Атрибут draw:formula определяет уравнение, по которому рассчитывается значение. Формула может использовать другие формулы или значения модификаторов в функциях и/или ссылках модификаторов.

```
number_digit = '0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'

number = number number_digit | number_digit
```

```
identifier = 'pi'|'left'|'top'|'right'|'bottom'|'xstretch'|'ystretch'|
             'hasstroke'|'hasfill'|'width'|'height'|'logwidth'|'logheight'

unary_function = 'abs'|'sqrt'|'sin'|'cos'|'tan'|'atan'|'atan2'
binary_function = 'min'|'max'
ternary_function = 'if'

function_reference = '?' 'a-z,A-Z,0-9' ' '
modifier_reference = '$' '0-9' ' '

basic_expression =
    number |
    identifier |
    function_reference |
    unary_function '(' additive_expression ')' |
    binary_function '(' additive_expression ',' additive_expression ')' |
    ternary_function '(' additive_expression ',' additive_expression ',
                         ' additive_expression ')' | '(' additive_expression ')'

unary_expression = '-' basic_expression

multiplicative_expression =
                        basic_expression |
                        multiplicative_expression '*' basic_expression |
                        multiplicative_expression '/' basic_expression

additive_expression =
                    multiplicative_expression |
                    additive_expression '+' multiplicative_expression |
                    additive_expression '-' multiplicative_expression
```

Идентификатор	***Описание***
left	Используется значение левой границы атрибута `svg:viewBox`
top	Используется значение верхней границы атрибута `svg:viewBox`
right	Используется значение правой границы атрибута `svg:viewBox`
bottom	Используется значение нижней границы атрибута `svg:viewBox`
xstretch	Используется значение `draw:path-stretchpoint-x`
ystretch	Используется значение `draw:path-stretchpoint-y`
hasstroke	Если фигура имеет стиль линий, используется значение 1
hasfill	Если фигура имеет стиль заполнения, используется значение 1
width	Используется значение ширины `svg:viewBox`
height	Используется значение высоты `svg:viewBox`
logwidth	Используется значение ширины `svg:viewBox` в 1/100 долях мм
logheight	Используется значение высоты `svg:viewBox` в 1/100 долях мм

Пример — использование атрибута `draw:formula`:

draw:formula="width+10-$0". Если значение первого модификатора равно 100 и ширина svg:viewbox равна 10000, то результат приведенной формулы будет 10000 + 10 – 100 = 9910.

```
<define name="draw-equation-attlist" combine="interleave">
    <optional>
        <attribute name="draw:formula">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

9.5.6 Расширенная геометрия: атрибуты меток-манипуляторов

Метка-манипулятор

Элемент <draw:handle> определяет одиночную метку-манипулятор.

```
<define name="draw-handle">
    <element name="draw:handle">
        <ref name="draw-handle-attlist"/>
        <empty/>
    </element>
</define>
```

Вертикальное отражение метки-манипулятора

Атрибут draw:handle-mirror-vertical определяет, является ли положение метки-манипулятора зеркально отраженным по оси x.

```
<define name="draw-handle-attlist" combine="interleave">
    <optional>
        <attribute name="draw:handle-mirror-vertical" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Горизонтальное отражение метки-манипулятора

Атрибут draw:handle-mirror-horizontal определяет, является ли положение метки-манипулятора зеркально отраженным по оси y.

```
<define name="draw-handle-attlist" combine="interleave">
    <optional>
        <attribute name="draw:handle-mirror-horizontal" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Переключатель метки-манипулятора

Атрибут draw:handle-switched определяет, будут ли изменяться направления метки-манипулятора, если высота фигуры больше, чем ее ширина.

```
7745 <define name="draw-handle-attlist" combine="interleave">
7746     <optional>
7747         <attribute name="draw:handle-switched" a:defaultValue="false">
7748             <ref name="boolean"/>
7749         </attribute>
7750     </optional>
7751 </define>
```

Положение метки-манипулятора

Атрибут draw:handle-position определяет положение метки-манипулятора и состоит из двух параметров.

Каждый параметр может быть числом с плавающей точкой, или он может иметь одно из следующих расширений.

- Символ «?» используется для отметки начала имени формулы. В этом случае в качестве значения параметра используется значение атрибута элемента draw:formula.
- Если символ «$» предшествует целочисленному значению, значением является индексированный атрибут draw:modifiers. В этом случае значение соответствующего модификатора используется как значение параметра.
- В зависимости от номера, параметр также может иметь значение одной из нижеперечисленных констант:

Константа	*Описание*
left	Используется значение атрибута draw:coordinate-origin-x
top	Используется значение атрибута draw:coordinate-origin-y
right	Используется значение атрибута draw:coordinate-origin-x + значение атрибута draw:coordinate-width
bottom	Используется значение атрибута draw:coordinate-origin-y + значение атрибута draw:coordinate-height
xstretch	Используется значение атрибута draw:path-stretchpoint-x
ystretch	Используется значение атрибута draw:path-stretchpoint-y
hasstroke	Если фигура имеет стиль линий, используется значение 1
hasfill	Если фигура имеет стиль заполнения цветом, используется значение 1.

Константа	*Описание*
width	Используется значение ширины `svg:viewBox`
height	Используется значение высоты `svg:viewBox`
logwidth	Используется значение ширины `svg:viewBox` в 1/100 долях мм
logheight	Используется значение высоты `svg:viewBox` в 1/100 долях мм

Атрибут `draw:handle-position` определяет положение метки-манипулятора. Если атрибут `draw:handle-polar` не установлен, то первый параметр атрибута `draw:handle-position` определяет горизонтальное положение метки-манипулятора, вертикальное положение метки-манипулятора описывается вторым параметром. Если установлен атрибут `draw:handle-polar`, то метка-указатель является полярной меткой и первый параметр атрибута `draw:handle-position` определяет угол в градах, радиус метки-указателя определяется вторым параметром.

Пример атрибута `draw:handle-position` ***следующий:***

`draw:handle-position = "left $5"`

```
<define name="draw-handle-attlist" combine="interleave">
    <attribute name="draw:handle-position">
        <ref name="string"/>
    </attribute>
</define>
```

Минимальный диапазон перемещения по оси X для манипулятора

Атрибут `draw:handle-range-x-minimum` определяет минимум горизонтального диапазона, внутри которого может перемещаться метка-манипулятор. Синтаксис для атрибута такой же, как и для атрибута `draw:handle-position`, за исключением того, что используется только первый параметр.

Пример объявления атрибута минимального значения из первой формулы:

`draw:handle-range-x-minimum = "?Formula1"`

```
<define name="draw-handle-attlist" combine="interleave">
    <optional>
        <attribute name="draw:handle-range-x-minimum">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Максимальный диапазон перемещения по оси X для манипулятора

Атрибут `draw:handle-range-x-maximum` определяет максимум горизонтального диапазона, внутри которого может перемещаться метка-манипулятор. Синтаксис атрибута такой же, как и для атрибута `draw:handle-range-x-minimum`.

```
<define name="draw-handle-attlist" combine="interleave">
    <optional>
        <attribute name="draw:handle-range-x-maximum">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Минимальный диапазон перемещения по оси Y для манипулятора

Атрибут `draw:handle-range-y-minimum` определяет минимум вертикального диапазона, внутри которого может перемещаться метка-манипулятор. Синтаксис атрибута такой же, как и для атрибута `draw:handle-range-x-minimum`.

```
<define name="draw-handle-attlist" combine="interleave">
    <optional>
        <attribute name="draw:handle-range-y-minimum">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Максимальный диапазон перемещения по оси Y для манипулятора

Атрибут `draw:handle-range-y-maximum` определяет максимум вертикального диапазона, внутри которого может перемещаться метка-манипулятор. Синтаксис атрибута такой же, как и для атрибута `draw:handle-range-x-minimum`.

```
<define name="draw-handle-attlist" combine="interleave">
    <optional>
        <attribute name="draw:handle-range-y-maximum">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Полярная метка-манипулятор

Атрибут draw:handle-polar определяет, является ли метка-манипулятор полярной меткой-манипулятором. Синтаксис для данного атрибута такой же, как и для атрибута draw:handle-position. Первый параметр определяет горизонтальное положение центра, вертикальное положение центра определяется вторым параметром. Если атрибут установлен, атрибуты draw:handle-range-x и draw:handle-range-y игнорируются, вместо них могут быть использованы атрибуты draw:handle-radius-range-minumum и draw:handle-radius-range-maximum.

```
<define name="draw-handle-attlist" combine="interleave">
    <optional>
        <attribute name="draw:handle-polar">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Минимальный диапазон радиуса метки-манипулятора

Если установлен атрибут draw:handle-radius-range-minimum, то он определяет минимум диапазона радиуса, который может быть использован для полярной метки-манипулятора. Синтаксис атрибута такой же, как и для атрибута draw:handle-range-x-minimum.

```
<define name="draw-handle-attlist" combine="interleave">
    <optional>
        <attribute name="draw:handle-radius-range-minimum">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Максимальный диапазон радиуса метки-манипулятора

Если установлен атрибут draw:handle-radius-range-maximum, он определяет максимум диапазона радиуса, который может быть использован для полярной метки-манипулятора. Синтаксис атрибута такой же, как и для атрибута draw:handle-range-x-minimum.

```
<define name="draw-handle-attlist" combine="interleave">
    <optional>
        <attribute name="draw:handle-radius-range-maximum">
            <ref name="string"/>
        </attribute>
    </optional>
```

```
7805 </define>
```

9.6 Фигуры презентации

Фигуры презентации — это специальные фигуры текстовых блоков, изображений, объектов или уменьшенных изображений, содержащиеся в презентации. Фигуры презентации используют стили со значением семейства стилей presentation, в отличие от векторных фигур, которые используют стили со значением семейства стилей graphic. Фигуры презентации могут быть пустыми, действуя как поля для заполнения. Если разметка графической страницы презентации (см. раздел 14.15) изменяется, все фигуры презентации автоматически адаптируются.

Стандартные векторные фигуры также могут быть использованы в презентации. Атрибут presentation:class отличает фигуры презентаций от векторных фигур. В отличие от фигур презентации векторные фигуры не адаптируются, если разметка страницы изменяется.

9.6.1 Общие атрибуты фигур презентации

Атрибуты, описанные в данном разделе, являются общими для всех фигур презентации.

Стиль

Фигуры презентации могут иметь связанные с ними стили из семейства стилей presentation. Фигуру презентации можно отличить от векторной фигуры проверкой наличия атрибута presentation:style-name. Векторные фигуры используют атрибут draw:style-name со стилем из семейства graphic, в то время как фигуры презентации используют атрибут presentation:style-name со стилем из семейства presentation. Это имя ссылается на элемент <style:style> с семейством presentation. Свойства форматирования этого стиля, а также его необязательных родительских стилей, используются для форматирования данной фигуры (см. также раздел 9.2.15).

Класс

Атрибут `presentation:class` классифицирует фигуры презентации по признакам их использования на графической странице (например, заголовок или структура). Существуют следующие классы:

- `title` — заглавия являются стандартными текстовыми фигурами;
- `outline` — структуры являются стандартными текстовыми фигурами;
- `subtitle` — подзаголовки являются стандартными текстовыми фигурами;
- `text` — тексты презентации являются стандартными текстовыми фигурами;
- `graphic` — графические объекты являются стандартными графическими фигурами;
- `object` — объекты презентации являются стандартными фигурами объекта;
- `chart` — диаграммы являются стандартными фигурами объекта;
- `table` — таблицы презентации являются стандартными фигурами объекта;
- `orgchart` — организационные диаграммы являются стандартными фигурами объекта;
- `page` — страницы используются страницами заметок;
- `notes` — заметки презентации используются страницами заметок;
- `handout` — слайды тезисов являются полями для заполнения для графических страниц в странице мастер-слайд тезисов.

Следующие четыре класса могут быть использованы только для векторных фигур, содержащихся на мастер-странице. В зависимости от установок страницы (см. раздел 15.36), они отображаются автоматически на страницах рисования, которые используют мастер-страницу:

- `header` — фигура используется как верхний колонтитул, фигуры верхнего колонтитула являются стандартными текстовыми фигурами;
- `footer` — фигура используется как нижний колонтитул, фигуры нижнего колонтитула являются стандартными текстовыми фигурами;

- `date-time` — фигура используется как фигура времени и/или даты, Фигуры времени и даты являются стандартными текстовыми фигурами;
- `page-number` — фигуры используются как фигуры номеров страниц, фигуры номеров страниц являются стандартными текстовыми фигурами.

```
<define name="presentation-shape-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:class">
            <ref name="presentation-classes"/>
        </attribute>
    </optional>
</define>
<define name="presentation-classes">
    <choice>
        <value>title</value>
        <value>outline</value>
        <value>subtitle</value>
        <value>text</value>
        <value>graphic</value>
        <value>object</value>
        <value>chart</value>
        <value>table</value>
        <value>orgchart</value>
        <value>page</value>
        <value>notes</value>
        <value>handout</value>
        <value>header</value>
        <value>footer</value>
        <value>date-time</value>
        <value>page-number</value>
    </choice>
</define>
```

Поле для заполнения

Атрибут `presentation:placeholder` определяет, является ли фигура полем для заполнения или объектом презентации с фактическим содержимым.

```
<define name="presentation-shape-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:placeholder">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Пользовательское преобразование

Атрибут `presentation:user-transformed` определяет, устанавливается ли пользователем положение и размер фигуры, или же они устанавливаются в соответствии с фигурой презентации на мастер-странице.

```
<define name="presentation-shape-attlist" combine="interleave">
    <optional>
```

```
        <attribute name="presentation:user-transformed">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

9.7 Анимация презентаций

В документе презентации векторные фигуры могут быть анимированы. Каждая страница презентации может иметь необязательный элемент `<presentation:animations>`, который служит контейнером для эффектов анимации. Анимация осуществляется в момент отображения страницы при показе презентации.

Эта спецификация допускает множество эффектов для одной и той же фигуры внутри страницы. Приложения могут иметь ограничения относительно числа и комбинации эффектов, применимых к фигуре, например, поддерживать только один элемент показа и один элемент затухания для фигуры с дополнительным показом и затуханием текста, одним эффектом затемнения и одним звуковым эффектом.

```
<define name="presentation-animations">
    <element name="presentation:animations">
        <zeroOrMore>
            <choice>
                <ref name="presentation-animation-elements"/>
                <ref name="presentation-animation-group"/>
            </choice>
        </zeroOrMore>
    </element>
</define>
<define name="presentation-animation-elements">
    <choice>
        <ref name="presentation-show-shape"/>
        <ref name="presentation-show-text"/>
        <ref name="presentation-hide-shape"/>
        <ref name="presentation-hide-text"/>
        <ref name="presentation-dim"/>
        <ref name="presentation-play"/>
    </choice>
</define>
```

9.7.1 Звук

Элемент `<presentation:sound>` может содержаться во всех элементах анимации презентации, которые поддерживают звук. Звуковой файл, указанный атрибутами XLink, воспроизводится во время выполнения эффекта.

```
<define name="presentation-sound">
    <element name="presentation:sound">
```

```
7869        <ref name="presentation-sound-attlist"/>
7870        <attribute name="xlink:href">
7871            <ref name="anyURI"/>
7872        </attribute>
7873        <optional>
7874            <attribute name="xlink:type" a:defaultValue="simple">
7875                <choice>
7876                    <value>simple</value>
7877                </choice>
7878            </attribute>
7879        </optional>
7880        <optional>
7881            <attribute name="xlink:actuate" a:defaultValue="onRequest">
7882                <choice>
7883                    <value>onRequest</value>
7884                </choice>
7885            </attribute>
7886        </optional>
7887        <optional>
7888            <attribute name="xlink:show">
7889                <choice>
7890                    <value>new</value>
7891                    <value>replace</value>
7892                </choice>
7893            </attribute>
7894        </optional>
7895        <empty/>
7896    </element>
7897 </define>
```

С элементом `<presentation:sound>` может быть связан следующий атрибут:

- воспроизводить полностью.

Воспроизводить полностью

Если значением атрибута `presentation:play-full` является `true`, следующий эффект начнется после завершения воспроизведения звука. Если значение данного атрибута `false`, следующий эффект начнется сразу после завершения текущего эффекта.

```
7898 <define name="presentation-sound-attlist" combine="interleave">
7899     <optional>
7900         <attribute name="presentation:play-full">
7901             <ref name="boolean"/>
7902         </attribute>
7903     </optional>
7904 </define>
```

9.7.2 Показ фигуры

Элемент `<presentation:show-shape>` делает фигуру видимой. Если для фигуры указан элемент`<presentation:show-shape>`, эта фигура автоматически становится невидимой до начала выполнения эффекта.

```
<define name="presentation-show-shape">
    <element name="presentation:show-shape">
        <ref name="common-presentation-effect-attlist"/>
        <optional>
            <ref name="presentation-sound"/>
        </optional>
    </element>
</define>
```

С элементом `<presentation:show-shape>` могут быть связаны следующие атрибуты:

- фигура;
- эффект;
- направление;
- скорость;
- задержка;
- начальный масштаб;
- траектория.

Фигура

Атрибут `draw:shape-id` определяет фигуру для применения эффекта указанием ее идентификатора.

```
<define name="common-presentation-effect-attlist" combine="interleave">
    <attribute name="draw:shape-id">
        <ref name="IDREF"/>
    </attribute>
</define>
```

Эффект

Атрибут `presentation:effect` определяет тип эффекта.

- `none` — эффект не используется;
- `fade` — фигура постепенно изменяется от ее видимого или невидимого состояния до невидимого или видимого состояния соответственно;
- `move` — фигура перемещается из конечного положения или в ее конечное положение;

- stripes — фигура плавно проявляется или исчезает в процессе рисования или стирания горизонтальных или вертикальных полос, изменяющих ее размер;
- open — фигура рисуется или убирается линия за линией либо в горизонтальном, либо в вертикальном направлении, начиная от центра фигуры;
- close — фигура рисуется или убирается линия за линией либо в горизонтальном, либо в вертикальном направлении, начиная с края фигуры;
- dissolve — фигура плавно проявляется или исчезает в процессе рисования или удаления маленьких блоков в произвольном порядке;
- wavyline — фигура плавно проявляется или исчезает в процессе рисования или удаления маленьких блоков «змейкой»;
- random — эффект выбирается случайным образом для плавного появления или исчезновения фигуры;
- lines — фигура рисуется или убирается линия за линией либо в горизонтальном, либо в вертикальном направлении, в произвольном порядке;
- laser — этот эффект применим только для текстовых фигур, символы текста двигаются один за другим от верхнего края экрана к их конечному положению;
- appear — фигура постепенно проявляется в процессе изменения своего состояния от невидимого к видимому;
- hide — фигура постепенно исчезает в процессе изменения своего состояния от видимого к невидимому;
- move-shor — как и эффект move, но перемещение фигуры обрезается ограничивающим прямоугольником во время постепенного изменения;
- checkerboard — фигура плавно проявляется или исчезает в процессе рисования или удаления шахматной доски, блоки которой со временем увеличиваются в размерах;
- rotate — фигура вертикально или горизонтально вращается небольшой промежуток времени;

- `stretch` — фигура плавно проявляется или исчезает в процессе изменения ее размера.

```
7918 <define name="common-presentation-effect-attlist" combine="interleave">
7919     <optional>
7920         <attribute name="presentation:effect" a:defaultValue="none">
7921             <ref name="presentationEffects"/>
7922         </attribute>
7923     </optional>
7924 </define>
7925 <define name="presentationEffects">
7926     <choice>
7927         <value>none</value>
7928         <value>fade</value>
7929         <value>move</value>
7930         <value>stripes</value>
7931         <value>open</value>
7932         <value>close</value>
7933         <value>dissolve</value>
7934         <value>wavyline</value>
7935         <value>random</value>
7936         <value>lines</value>
7937         <value>laser</value>
7938         <value>appear</value>
7939         <value>hide</value>
7940         <value>move-short</value>
7941         <value>checkerboard</value>
7942         <value>rotate</value>
7943         <value>stretch</value>
7944     </choice>
7945 </define>
```

Направление

Атрибут `presentation:direction` определяет направление эффекта. Он нужен только для некоторых эффектов.

```
7946 <define name="common-presentation-effect-attlist" combine="interleave">
7947     <optional>
7948         <attribute name="presentation:direction" a:defaultValue="none">
7949             <ref name="presentationEffectDirections"/>
7950         </attribute>
7951     </optional>
7952 </define>
7953 <define name="presentationEffectDirections">
7954     <choice>
7955         <value>none</value>
7956         <value>from-left</value>
7957         <value>from-top</value>
7958         <value>from-right</value>
7959         <value>from-bottom</value>
7960         <value>from-center</value>
7961         <value>from-upper-left</value>
7962         <value>from-upper-right</value>
7963         <value>from-lower-left</value>
7964         <value>from-lower-right</value>
7965         <value>to-left</value>
7966         <value>to-top</value>
7967         <value>to-right</value>
7968         <value>to-bottom</value>
```

```
        <value>to-upper-left</value>
        <value>to-upper-right</value>
        <value>to-lower-right</value>
        <value>to-lower-left</value>
        <value>path</value>
        <value>spiral-inward-left</value>
        <value>spiral-inward-right</value>
        <value>spiral-outward-left</value>
        <value>spiral-outward-right</value>
        <value>vertical</value>
        <value>horizontal</value>
        <value>to-center</value>
        <value>clockwise</value>
        <value>counter-clockwise</value>
    </choice>
</define>
```

Скорость

Атрибут `presentation:speed` определяет скорость эффекта.

```
<define name="common-presentation-effect-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:speed" a:defaultValue="medium">
            <ref name="presentationSpeeds"/>
        </attribute>
    </optional>
</define>
<define name="presentationSpeeds">
    <choice>
        <value>slow</value>
        <value>medium</value>
        <value>fast</value>
    </choice>
</define>
```

Задержка

Атрибут `presentation:delay` определяет задержку перед началом эффекта презентации после завершения предыдущего эффекта.

```
<define name="common-presentation-effect-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:delay">
            <ref name="duration"/>
        </attribute>
    </optional>
</define>
```

Начальный масштаб

Некоторые эффекты изменяют масштаб фигуры во время своего выполнения. Атрибут `presentation:start-scale` определяет начальный размер фигуры в процентах от ее оригинального размера.

```
<define name="common-presentation-effect-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="presentation:start-scale" a:defaultValue="100%">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

Траектория

Атрибут `presentation:path-id` применяется для эффекта `move`. Атрибут определяет идентификатор фигуры многоугольника. Эффект движется вдоль линий указанного многоугольника. Указанный многоугольник является невидимым во время показа презентации.

```
<define name="common-presentation-effect-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:path-id"/>
    </optional>
</define>
```

9.7.3 Показ текста

Элемент `<presentation:show-text>` делает текст фигуры видимым. Если элемент `<show-text>` указан для фигуры, текст фигуры автоматически становится невидимым до начала выполнения эффекта.

```
<define name="presentation-show-text">
    <element name="presentation:show-text">
        <ref name="common-presentation-effect-attlist"/>
        <optional>
            <ref name="presentation-sound"/>
        </optional>
    </element>
</define>
```

С элементом `<presentation:show-text>` могут быть связаны следующие атрибуты:

- фигура, эффект, направление, скорость, начальный масштаб, траектория (см. раздел 9.7.2).

9.7.4 Скрытие фигуры

Элемент `<presentation:hide-shape>` делает фигуру невидимой.

```
<define name="presentation-hide-shape">
    <element name="presentation:hide-shape">
        <ref name="common-presentation-effect-attlist"/>
        <optional>
            <ref name="presentation-sound"/>
        </optional>
```

```
    </element>
</define>
```

С элементом `<presentation:hide-shape>` могут быть связаны следующие атрибуты:

- фигура, эффект, направление, скорость, начальный масштаб, траектория (см. раздел 9.7.2).

9.7.5 Скрытие текста

Элемент `<presentation:hide-text>` делает текст фигуры невидимым.

```
<define name="presentation-hide-text">
    <element name="presentation:hide-text">
        <ref name="common-presentation-effect-attlist"/>
        <optional>
            <ref name="presentation-sound"/>
        </optional>
    </element>
</define>
```

С элементом `<presentation:hide-text>` могут быть связаны следующие атрибуты:

- фигура, эффект, направление, скорость, начальный масштаб, траектория (см. раздел 9.7.2).

9.7.6 Затемнение

Элемент `<presentation:dim>` закрашивает фигуру одним цветом.

```
<define name="presentation-dim">
    <element name="presentation:dim">
        <ref name="presentation-dim-attlist"/>
        <optional>
            <ref name="presentation-sound"/>
        </optional>
    </element>
</define>
```

С элементом `<presentation:dim>` могут быть связаны следующие атрибуты:

- фигура (см. раздел 9.7.2);
- цвет.

```
<define name="presentation-dim-attlist" combine="interleave">
    <attribute name="draw:shape-id">
        <ref name="IDREF"/>
    </attribute>
</define>
```

Цвет

Атрибут `draw:color` определяет цвет заполнения фигуры, если она затемнена.

```
<define name="presentation-dim-attlist" combine="interleave">
    <attribute name="draw:color">
        <ref name="color"/>
    </attribute>
</define>
```

9.7.7 Воспроизведение

Элемент `<presentation:play>` начинает анимацию фигуры, поддерживающей анимацию.

```
<define name="presentation-play">
    <element name="presentation:play">
        <ref name="presentation-play-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<presentation:play>` могут быть связаны следующие атрибуты:

- идентификатор фигуры и скорость (см. раздел 9.7.2).

```
<define name="presentation-play-attlist" combine="interleave">
    <attribute name="draw:shape-id">
        <ref name="IDREF"/>
    </attribute>
    <optional>
        <attribute name="presentation:speed" a:defaultValue="medium">
            <ref name="presentationSpeeds"/>
        </attribute>
    </optional>
</define>
```

9.7.8 Группы эффектов

Элемент `<presentation:animation-group>` позволяет определить начало нескольких эффектов в один момент времени.

```
<define name="presentation-animation-group">
    <element name="presentation:animation-group">
        <zeroOrMore>
            <ref name="presentation-animation-elements"/>
        </zeroOrMore>
    </element>
</define>
```

9.8 Анимация SMIL-презентаций

В этом разделе описана анимация фигур для документов презентаций, основанная на [SMIL20]. Этот тип анимации может быть использован вместо того, который определяется элементом `<presentation:animations>`, если требуется достичь одно из нижеперечисленного:

- множественная анимация для одной фигуры;
- комбинация анимации, начинающейся по действию пользователя и стартующей автоматически на странице;
- множество эффектов анимации, происходящих в одно и то же время;
- дополнительные эффекты, «программируемые» в XML путем совмещения основных элементов анимации;
- преобразования документа к SVG, включающему [SMIL20].

9.8.1 Рекомендуемое применение SMIL

В следующих разделах описано использование элементов SMIL-анимации, позволяющих офисному приложению показывать пользователю анимированные элементы в виде простого и легкого пользовательского интерфейса. Этот пользовательский интерфейс может содержать одну основную последовательность эффектов и, в дополнение к ней, множество последовательностей эффектов, которые начинаются как взаимодействие с графическими фигурами. *Эффект* — это комбинация одного или более элементов анимации одиночной фигуры и/или абзацев этой фигуры.

Рекомендуется, чтобы в пользовательском интерфейсе эффекты могли создаваться с использованием представлений, имеющих локализованные и легко интерпретируемые имена. В этом случае пользователь не будет работать с иерархией элементов SMIL-анимации, а будет пользоваться одномерным списком эффектов, что гораздо проще для пользователей офисных приложений.

Анимация слайдов

Каждый элемент `<draw:page>` может опционально иметь элемент `<anim:par>`, который задает анимацию страницы во время показа слайдов. Элемент `<anim:par>` должен содержать один элемент `<anim:seq>`, который

является основной последовательностью для эффектов фигуры, и ноль или больше элементов <anim:seq> для задания интерактивных последовательностей фигур, содержащих взаимодействующую анимацию. Элементы анимации запускаются после того, как слайд осуществит свой начальный переход.

Основная последовательность

Основная последовательность — это элемент <anim:seq>, содержащий эффекты, которые должны начаться после того, как слайд выполнил свой начальный переход. Поскольку это последовательный контейнер, его дочерние узлы выполняются один за другим. Если атрибут дочернего узла smil:begin имеет значение indefinite, то выполнение останавливается до тех пор, пока пользователь не продвинет показ слайдов мышью или клавиатурой.

Первый уровень дочерних узлов в главной последовательности должен быть элементами <anim:par>, которые группируют элементы анимации, начинающиеся с одного и того же действия пользователя. Второй уровень дочерних элементов должен быть элементами <anim:par>, которые группируют элементы анимации начинающиеся в одно и то же время. Третий уровень дочерних узлов должен быть элементами <anim:par>, которые группируют элементы для единого эффекта.

Пример —

Следующий пример иллюстрирует основную последовательность с эффектами A, B, C и D. Эффект A начинается по взаимодействию с пользователем, эффект B стартует одновременно с A. Эффект C стартует через 4 секунды после эффектов A и B. Эффект D стартует при следующем взаимодействии с пользователем:

```
<amin:par> <!-- timing root-->
    <anim:seq> <!-- main sequence-->
        <anim:par smil:begin="indefinite">
            <!-- first user interaction -->
            <anim:par smil:begin="0s" smil:dur="4s">
                <!-- first group of effects to execute -->
                <anim:par> <!-- effect a -->
                    <!-- nodes for effect a-->
                </anim:par>
                <anim:par> <!-- effect b -->
                    <!-- nodes for effect b-->
                </anim:par>
            </anim:par>
            <anim:par smil:begin="4s">
                <!-- second group of effects to execute -->
```

```
                <anim:par> <!-- effect c -->
                    <!-- nodes for effect c-->
                </anim:par>
            </anim:par>
        </anim:par>
        <anim:par>
            <!-- second user interaction-->
            <anim:par smil:begin="indefinite">
                <!-- first group of effects to execute -->
                <anim:par> <!-- effect d -->
                    <!--- nodes for effect d-->
                </anim:par>
            </anim:par>
        </anim:par>
    </anim:seq>
</anim:par>
```

Интерактивная последовательность

Интерактивная последовательность — это элемент `<anim:seq>`, который должен иметь такую же структуру как и основная последовательность. Единственным отличием является то, что элемент `<anim:par>` на первом уровне значением атрибута `smil:begin` имеет `[shape-id].click`, где `[shape-id]` идентифицирует графические фигуры по значению их атрибута `draw:id`. Эти элементы анимации включаются, когда пользователь взаимодействует с элементом, определенным `[shape-id]`.

9.8.2 Значения атрибутов SMIL-анимации, зависящие от документа

В этом разделе описаны значения атрибутов, зависящих от типа документа, описанных в разделе 13, если они используются внутри документа презентации.

Целевой элемент повторения

Для документов презентаций атрибут `smil:targetElement` элемента `<anim:iterate>` (см. раздел 13.4.4) может ссылаться на элемент графической фигуры или абзаца. Если атрибуту `anim:sub-item` элемента `<anim:iterate>` установлено значение `whole`, повтор включает в себя и фон фигуры, и ее текст. Если значение атрибута `anim:sub-item` установлено как `text`, повторяется только текст фигуры.

Тип повторения

Для документов презентаций атрибут `anim:iterate-type` элемента `<anim:iterate>` (см. раздел 13.4.4) может принимать следующие значения:

- `by-paragraph` — целевая фигура повторяется по абзацам;
- `by-word` — целевая фигура или абзац повторяются по словам;
- `by-letter` — целевая фигура или абзац повторяются по символам.

Целевой элемент

Для документов презентаций атрибут `smil:targetElement`, описанный в разделе 13.3.1, может ссылаться на фигуры по значению их атрибута `draw:id` и абзацы по значению их атрибута `text:id`.

Целевой атрибут

Для документов презентаций атрибут `smil:attributeName`, описанный в разделе 13.3.1, может принимать следующие значения:

- `x` — анимируется x-положение элемента, значения даются в размерах экрана, где 0 означает левый край, 1 означает правый край;
- `y` — анимируется y-положение элемента, значения даются в размерах экрана, где 0 означает верхний край, 1 означает нижний край;
- `width` — анимируется ширина элемента, значения даются в размерах экрана, где 0 означает отсутствие ширины, 1 означает ширину экрана;
- `height` — анимируется высота элемента, значения даются в размерах экрана, где 0 означает отсутствие высоты, 1 означает высоту экрана;
- `color` — анимируется цвет элемента, анимируются одновременно цвета заливки, линий и символов, значения могут быть RGB или HSL;
- `rotate` — анимируется поворот элемента, анимируются одновременно фигура и текстовая анимация;
- `skewX` — анимируется горизонтальный наклон элемента;
- `fillColor` — анимируется цвет заливки элемента;
- `fillStyle` — анимируется стиль заливки элемента;
- `lineColor` — анимируется цвет линий элемента;
- `lineStyle` — анимируется стиль линий элемента;

- charColor — анимируется цвет символов элемента;
- charWeight — анимируется ширина символов текста элемента;
- charUnderline — анимируется подчеркивание текста элемента;
- charFontName — анимируется шрифт текста элемента;
- charHeight — анимируется высота текста элемента;
- charPosture — анимируется положение текста элемента;
- visibility — анимируется видимость элемента;
- opacity — анимируется непрозрачность элемента.

Подпункты целевого элемента

Для документов презентаций атрибут anim:sub-item, описанный в разделе 13.3.1, может принимать следующие значения:

- whole — анимируется и фигура, и ее текст;
- background — анимируется только фон фигуры, но не ее текст;
- text — анимируется только текст.

Формула

Для документов презентаций атрибут anim:formula, описанный в разделе 13.3.2, может содержать следующие дополнительные идентификаторы:

- e, — эйлерова константа;
- x — левая вершина анимированного элемента в размерах экрана, где 0 означает левый край экрана, 1 означает правый край экрана;
- y — верхняя вершина анимированного элемента в размерах экрана, где 0 означает верхний край экрана, 1 означает нижний край экрана;
- width — ширина анимированного элемента в размерах экрана, где 0 означает отсутствие ширины, 1 означает ширину экрана;
- height — высота анимированного элемента в размерах экрана, где 0 означает отсутствие высоты, 1 означает высоту экрана.

Команда

Для документов презентаций атрибут anim:command элемента <anim:command> (см. раздел 13.6.1) может принимать следующие значения:

- `custom` — команда задается пользователем;
- `verb` — команда нацелена на OLE2-объект, параметр `verb` соответствует номеру команды, которая будет применена к OLE2–объекту;
- `play` — команда нацелена на медиа-объект и начинает его проигрывание, необязательный параметр `media-time` задает время начала проигрывания в секундах, если этот параметр не задан, то проигрывание начнется с последней позиции;
- `toggle-pause` — команда нацелена на медиа-объект и переключает его состояние из проигрывания в паузу и наоборот;
- `stop` — команда нацелена на медиа-объект и останавливает его проигрывание;
- `stop-audio` — команда ни на что не нацелена и останавливает проигрывание всех текущих аудиозаписей.

9.8.3 Атрибуты анимации SMIL-презентации

Атрибуты, описанные в данном разделе, могут быть добавлены к элементам анимации, описанным в разделах 13.4, 13.5 и 13.6, если они используются внутри документа презентации. Они не влияют на фактическое поведение анимации, но помогают пользовательскому интерфейсу офисных приложений в представлении установок эффектов анимации.

Тип узла

Атрибут `presentation:node-type` определяет тип узла для элемента анимации. Этот атрибут не изменяет поведение элемента, однако помогает приложению быстро идентифицировать назначение элемента внутри иерархии элементов анимации. Значением данного атрибута может быть:

- `default` — этот элемент анимации не имеет специального значения для приложения, это установка по умолчанию;
- `on-click` — этот элемент анимации является корневым элементом эффекта, который начинается по пользовательскому щелчку;
- `with-previous` — этот элемент анимации является корневым для эффекта, который начинается вместе с предыдущим эффектом;

- `after-previous` — этот элемент анимации является корневым для эффекта, который начинается после предыдущего эффекта;
- `timing-root` — этот элемент анимации является корневым элементом для анимации страницы;
- `main-sequence` — этот элемент анимации является корневым элементом для главной последовательности эффектов страницы;
- `interactive-sequence` — этот элемент анимации является корневым элементом для последовательности эффектов, которая начинается при взаимодействии пользователя со специальными элементами на странице.

```
<define name="common-anim-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:node-type" a:defaultValue="default">
            <choice>
                <value>default</value>
                <value>on-click</value>
                <value>with-previous</value>
                <value>after-previous</value>
                <value>timing-root</value>
                <value>main-sequence</value>
                <value>interactive-sequence</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Идентификатор предустановки

Атрибут `presentation:preset-id` определяет имя предустановки, которое было использовано для создания этого эффекта анимации.

```
<define name="common-anim-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:preset-id">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Подтип предустановки

Атрибут `presentation:preset-sub-type` определяет подтип предустановки, с помощью которой создавался данный эффект анимации.

```
<define name="common-anim-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:preset-sub-type">
            <ref name="string"/>
        </attribute>
```

```
    </optional>
</define>
```

Класс предустановки

Атрибут `presentation:preset-class` определяет класс предустановки, с помощью которой создавался данный эффект анимации. Значением данного атрибута может быть:

- `custom` — предустановка была определена пользователем, это установка по умолчанию;
- `entrance` — предустановка была эффектом входа;
- `exit` — предустановка была эффектом выхода;
- `emphasis` — предустановка была эффектом выделения;
- `motion-path` — предустановка была траекторией движения;
- `ole-action` — предустановка была OLE-действием;
- `media-call` — предустановка была вызовом медиа-содержимого.

```
<define name="common-anim-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:preset-class" a:defaultValue="custom">
            <choice>
                <value>custom</value>
                <value>entrance</value>
                <value>exit</value>
                <value>emphasis</value>
                <value>motion-path</value>
                <value>ole-action</value>
                <value>media-call</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Мастер-элемент

Атрибут `presentation:master-element` определяет идентификатор элемента анимации. Пользовательский интерфейс офисных приложений может только отображать элементы анимации, которые не имеют атрибута `presentation:master-element`, и может рассматривать элементы, имеющие атрибут `presentation:master-element` как части элементов анимации, на которые осуществляется ссылка.

```
<define name="common-anim-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:master-element">
            <ref name="IDREF"/>
```

```
        </attribute>
    </optional>
</define>
```

Идентификатор группы

Атрибут `presentation:group-id` определяет идентификатор группы. Этот идентификатор может быть использован для группировки элементов анимации в пользовательском интерфейсе, где группа состоит из всех элементов анимации, имеющих одинаковый идентификатор группы. Это может быть использовано, например, для группировки элементов анимации, которые анимируют абзацы одной фигуры.

```
<define name="common-anim-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:group-id">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

9.9 События презентации

Многие объекты внутри документа презентации поддерживают специальные события презентации. Например, пользователь может продвинуть один фрейм презентации, когда он щелкает по объекту с соответствующим событием. События презентации содержатся вместе с таблицей датчиков событий графических объектов (см. раздел 9.2.20 для получения дополнительной информации).

```
<define name="presentation-event-listener">
    <element name="presentation:event-listener">
        <ref name="presentation-event-listener-attlist"/>
        <optional>
            <ref name="presentation-sound"/>
        </optional>
    </element>
</define>
```

Имя события

Атрибут `script:event-name` определяет имя события (см. раздел 12.4.1 для получения подробной информации).

```
<define name="presentation-event-listener-attlist" combine="interleave">
    <attribute name="script:event-name">
        <ref name="string"/>
    </attribute>
</define>
```

Вызов события

Тип осуществляемого действия для свершения события может быть выбран с помощью атрибута presentation:action. Доступны следующие действия:

- none — никаких действий не происходит, когда включается данное событие;
- previous-page — показ презентации перескакивает на предыдущую страницу;
- next-page — показ презентации перескакивает на следующую страницу;
- first-page — показ презентации перескакивает на первую страницу текущего документа;
- last-page — показ презентации перескакивает на последнюю страницу текущего документа;
- hide — объект, содержащий это событие, скрывается, если осуществляется событие;
- stop — если показ слайдов активен, то он будет остановлен;
- execute — стороннее приложение запускается при совершении данного события, приложение может быть задано при помощи Xlink;
- show — целевой адрес URL открывается при совершении данного события, целевой адрес URL может быть задан при помощи Xlink;
- verb — если объект, который содержит данное событие, поддерживает запуск команд [OLE], выполняется команда с идентификатором, установленным в атрибуте presentation:verb ;
- fade-out — объект, содержащий данное событие, обесцвечивается при совершении этого события, атрибуты presentation:effect, presentation:direction, presentation:speed и presentation:start-scale могут быть использованы для задания эффекта;
- sound — звуковой эффект начинается при совершении этого события, звуковой эффект описывается дочерним элементом <presentation:sound>.

```
<define name="presentation-event-listener-attlist" combine="interleave">
    <attribute name="presentation:action">
        <choice>
```

```
            <value>none</value>
            <value>previous-page</value>
            <value>next-page</value>
            <value>first-page</value>
            <value>last-page</value>
            <value>hide</value>
            <value>stop</value>
            <value>execute</value>
            <value>show</value>
            <value>verb</value>
            <value>fade-out</value>
            <value>sound</value>
        </choice>
    </attribute>
</define>
```

Эффект события

См. атрибут `presentation:effect` в разделе 9.7.2.

```
<define name="presentation-event-listener-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:effect" a:defaultValue="none">
            <ref name="presentationEffects"/>
        </attribute>
    </optional>
</define>
```

Направление эффекта

См. атрибут `presentation:direction` в разделе 9.7.2.

```
<define name="presentation-event-listener-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:direction" a:defaultValue="none">
            <ref name="presentationEffectDirections"/>
        </attribute>
    </optional>
</define>
```

Скорость эффекта

См. атрибут `presentation:speed` в разделе 9.7.2.

```
<define name="presentation-event-listener-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:speed" a:defaultValue="medium">
            <ref name="presentationSpeeds"/>
        </attribute>
    </optional>
</define>
```

Начальный масштаб

См. атрибут `presentation:start-scale` в разделе 9.7.2.

```
<define name="presentation-event-listener-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="presentation:start-scale" a:defaultValue="100%">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

Ссылка

В зависимости от действия, выбранного атрибутом `presentation:action`, этот атрибут `xlink:href` либо выбирает закладки документа, либо приложение.

```
<define name="presentation-event-listener-attlist" combine="interleave">
    <optional>
        <attribute name="xlink:href">
            <ref name="anyURI"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="xlink:type" a:defaultValue="simple">
            <choice>
                <value>simple</value>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="xlink:show" a:defaultValue="embed">
            <choice>
                <value>embed</value>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="xlink:actuate" a:defaultValue="onRequest">
            <choice>
                <value>onRequest</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Операция

Операция [OLE], определенная атрибутом `presentation:verb`, выполняется для типа датчиков событий `verb` в объекте, содержащем это событие.

```
<define name="presentation-event-listener-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:verb">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

9.10 Текстовые поля презентаций

Этот раздел описывает текстовые поля, специфические для текста графических страниц, содержащихся в презентации.

9.10.1 Поле верхнего колонтитула

Поля верхнего колонтитула отображают текст верхнего колонтитула, описанный в объявлении поля верхнего колонтитула (см. раздел 9.11.2). Использование конкретного объявления поля верхнего колонтитула задается атрибутом `presentation:use-header-name` графической страницы, на которой встречается поле. Если поле содержится в фигуре презентации внутри мастер-страницы (см. раздел 9.6.1), то используется атрибут `presentation:use-header-name` графической страницы, для которой рисуется фигура (см. раздел 9.1.4).

Это поле используется в основном внутри мастер-страниц. Так как его значение может быть различным для отдельных графических страниц, использующих мастер-страницу, то текущее значение поля не доступно.

```
<define name="paragraph-content" combine="choice">
    <element name="presentation:header">
        <empty/>
    </element>
</define>
```

9.10.2 Поле нижнего колонтитула

Поля нижнего колонтитула отображают текст нижнего колонтитула, описанный в объявлении поля нижнего колонтитула (см. раздел 9.11.3). Использование конкретного объявления поля нижнего колонтитула задается атрибутом `presentation:use-footer-name` графической страницы, на которой встречается поле. Если поле содержится в фигуре презентации внутри мастер-страницы (см. раздел 9.6.1), то используется атрибут `presentation:use-footer-name` графической страницы, для которой рисуется фигура (см. раздел 9.1.4).

Это поле используется в основном внутри мастер-страниц. Т.к. его значение может быть различным для отдельных графических страниц, использующих мастер-страницу, то текущее значение поля недоступно.

```
<define name="paragraph-content" combine="choice">
    <element name="presentation:footer">
        <empty/>
    </element>
</define>
```

9.10.3 Поле даты и времени

Поля даты и времени отображают дату и время, описанные в объявлении поля даты и времени (см. раздел 9.11.4). Какое объявление поля даты и времени используется, определяется атрибутом `presentation:use-date-time-name` графической страницы, в которой определено это поле. Если поле содержится в элементе презентации внутри мастер-страницы (см. раздел 9.6.1), то используется атрибут `presentation:use-date-time-name` графической страницы, для которой отображается фигура (см. раздел 9.1.4).

Это поле используется в основном внутри мастер-страниц. Т. к. его значение может быть различным для отдельных графических страниц, использующих мастер-страницу, то текущее значение поля недоступно.

```
<define name="paragraph-content" combine="choice">
    <element name="presentation:date-time">
        <empty/>
    </element>
</define>
```

9.11 Содержимое документов презентаций

9.11.1 Объявления презентации

Некоторые специальные текстовые поля презентации требуют объявления для каждого документа перед тем, как они могут быть использованы. Например, поля верхнего колонтитула требуют, чтобы текст колонтитула объявлялся отдельно. Эти объявления собраны в начале текстового документа.

```
<define name="presentation-decls">
    <zeroOrMore>
        <ref name="presentation-decl"/>
    </zeroOrMore>
</define>
```

9.11.2 Объявление поля верхнего колонтитула

Элемент `<presentation:header-decl>` определяет текст поля верхнего колонтитула (см. раздел 9.10.1 для получения дополнительной информации).

```
<define name="presentation-decl" combine="choice">
    <element name="presentation:header-decl">
        <ref name="presentation-header-decl-attlist"/>
        <text/>
    </element>
</define>
```

Имя

Атрибут `presentation:name` определяет имя объявления верхнего колонтитула.

```
<define name="presentation-header-decl-attlist" combine="interleave">
    <attribute name="presentation:name">
        <ref name="string"/>
    </attribute>
</define>
```

9.11.3 Объявление поля нижнего колонтитула

Элемент `<presentation:footer-decl>` определяет текст поля нижнего колонтитула (см. раздел 9.10.2 для получения дополнительной информации).

```
<define name="presentation-decl" combine="choice">
    <element name="presentation:footer-decl">
        <ref name="presentation-footer-decl-attlist"/>
        <text/>
    </element>
</define>
```

Имя

Атрибут `presentation:name` определяет имя объявления нижнего колонтитула.

```
<define name="presentation-footer-decl-attlist" combine="interleave">
    <attribute name="presentation:name">
        <ref name="string"/>
    </attribute>
</define>
```

9.11.4 Объявление поля даты и времени

Элемент `<presentation:date-time-decl>` определяет текст поля даты и времени (см. раздел 9.10.3 для получения дополнительной информации).

```
<define name="presentation-decl" combine="choice">
    <element name="presentation:date-time-decl">
        <ref name="presentation-date-time-decl-attlist"/>
        <text/>
    </element>
</define>
```

Имя

Атрибут `presentation:name` определяет имя объявления даты и времени.

```
<define name="presentation-date-time-decl-attlist" combine="interleave">
    <attribute name="presentation:name">
        <ref name="string"/>
    </attribute>
</define>
```

Источник

Атрибут `presentation:source` определяет, будет ли отображаться текущая дата и время или же фиксированное значение поля.

```
<define name="presentation-date-time-decl-attlist" combine="interleave">
    <attribute name="presentation:source">
        <choice>
            <value>fixed</value>
            <value>current-date</value>
        </choice>
    </attribute>
</define>
```

Стиль форматирования даты и времени

Стиль даты, указанный атрибутом `style:data-style-name`, используется для форматирования даты и времени полей `presentation:date-time` в случае, если поля не фиксированы.

```
<define name="presentation-date-time-decl-attlist" combine="interleave">
    <optional>
        <attribute name="style:data-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

9.11.5 Параметры презентации

Параметры для презентации расположены в элементе `<presentation:settings>` внутри элемента `<office:presentation>`. Эти параметры устанавливают поведение, если документ отображается в презентации.

```
<define name="presentation-settings">
    <optional>
        <element name="presentation:settings">
            <ref name="presentation-settings-attlist"/>
            <zeroOrMore>
                <ref name="presentation-show"/>
            </zeroOrMore>
```

```
        </element>
    </optional>
</define>
```

С элементом `<presentation:settings>` могут быть связаны следующие атрибуты:

- начальная страница;
- отображение;
- во весь экран;
- непрерывно;
- пауза;
- отображение логотипа;
- принудительный ручной режим;
- видимость указателя мыши;
- указатель мыши как ручка;
- начать с навигатора;
- анимация;
- перемещение по щелчку;
- оставаться наверху.

Начальная страница

Атрибут `presentation:start-page` определяет имя страницы, с которой начинается показ презентации. Если этот атрибут установлен, он переопределяет значение атрибута `presentation:show`.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:start-page">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Отображение

Атрибут `presentation:show` определяет имя описания демонстрации (см. раздел 9.11.6), которое используется для показа презентации. Если установлено значение атрибута `presentation:start-page`, то оно переопределяет значение данного атрибута.

```
<define name="presentation-settings-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="presentation:show">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Во весь экран

Атрибут `presentation:full-screen` определяет, будет ли презентация отображаться в полноэкранном режиме или в окне.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:full-screen" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Непрерывно

Атрибут `presentation:endless` включает и выключает непрерывный повтор презентации.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:endless" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Пауза

Если презентация повторяется непрерывно, то атрибут `presentation:pause` определяет время отображения экрана паузы перед следующим показом презентации. Если этот атрибут не установлен или имеет значение 0, то экран паузы не отображается в непрерывном режиме. Значение этого атрибута должно соответствовать формату временного периода, описанному в § 3.2.6 [xmlschema-2].

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:pause">
            <ref name="duration"/>
        </attribute>
    </optional>
</define>
```

Отображение логотипа

Атрибут `presentation:show-logo` определяет, будет или нет приложение презентации показывать свой логотип на экране паузы.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:show-logo" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Принудительный ручной режим

Если атрибут `presentation:force-manual` установлен, то он переопределяет все свойства атрибута `presentation:transition-type`, которые определены в странице презентации (см. раздел 15.36.1), и устанавливает их в значение `manual`.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:force-manual" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Видимость указателя мыши

Атрибут `presentation:mouse-visible` определяет, будет ли видимым указатель мыши во время показа презентации.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:mouse-visible" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Указатель мыши как ручка

Атрибут `presentation:mouse-as-pen` определяет, отображается ли указатель мыши как ручка или как указатель. Если он отображается как ручка, пользователь может рисовать эскизы на страницах во время показа презентации.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:mouse-as-pen" a:defaultValue="false">
            <ref name="boolean"/>
```

```
        </attribute>
    </optional>
</define>
```

Начать с навигатора

Атрибут `presentation:start-with-navigator` определяет, будет ли изначально отображаться окно навигатора во время показа презентации.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:start-with-navigator"
                 a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Анимация

Атрибут `presentation:animations` включает или выключает воспроизведение растровой анимации во время показа презентации.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:animations" a:defaultValue="enabled">
            <choice>
                <value>enabled</value>
                <value>disabled</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Перемещение по щелчку

Атрибут `presentation:transition-on-click` включает или отключает ручной переход щелканьем мыши по слайду во время показа презентации.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:transition-on-click"
                 a:defaultValue="enabled">
            <choice>
                <value>enabled</value>
                <value>disabled</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Оставаться наверху

Если атрибут presentation:stay-on-top установлен в значение true, окно презентации будет отображаться выше всех остальных окон во время показа презентации.

```
<define name="presentation-settings-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:stay-on-top" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

9.11.6 Описание демонстрации

Документ презентации может содержать один или более элементов <presentation:show>. Элемент <presentation:show> переопределяет порядок появления страниц во время показа презентации. Он также может быть использован для исключения страниц из презентации или для многократного повторения страницы во время показа.

Это необязательный элемент.

```
<define name="presentation-show">
    <element name="presentation:show">
        <ref name="presentation-show-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом <presentation:show> могут быть связаны следующие атрибуты:

- имя;
- страницы.

Имя

Атрибут presentation:name уникальным образом определяет элемент <presentation:show>.

```
<define name="presentation-show-attlist" combine="interleave">
    <attribute name="presentation:name">
        <ref name="string"/>
    </attribute>
</define>
```

Страницы

Атрибут `presentation:pages` содержит список имен страниц, разделенных запятыми. В презентации, использующей данный показ, страницы отображаются в порядке их появления в списке. Страницы могут включаться неоднократно.

```
<define name="presentation-show-attlist" combine="interleave">
    <attribute name="presentation:pages"/>
</define>
```

10 Содержимое диаграмм

В этом разделе описано XML-представление содержимого диаграммы:

- введение в документы диаграмм;
- диаграмма;
- заголовок, подзаголовок и нижний колонтитул;
- легенда;
- область построения;
- стена;
- основание диаграммы;
- оси координат;
- ряды данных;
- категории;
- точка данных;
- среднее значение;
- индикатор ошибок;
- кривые регрессии.

10.1 Введение в документы диаграмм

Документы диаграмм всегда содержатся внутри других XML-документов. Существует два типа документов, в которых могут содержаться диаграммы:

- документы, которые не предоставляют данные для диаграммы, данные в диаграмме содержатся в элементе `<table:table>` внутри элемента `<chart:chart>`;
- документы, которые предоставляют данные для диаграммы, данные диаграммы могут содержаться в элементе `<table:table>` в родительском документе, например в электронной таблице или текстовом документе.

Данные диаграммы описываются атрибутом `table:cell-range-address` в элементе `<chart:plot-area>`. Элемент `<chart:plot-area>` представляет визуальный контейнер всего ряда данных в диаграмме.

10.2 Диаграмма

Элемент `<chart:chart>` представляет всю диаграмму, включая заголовки, легенду и графический объект, отображающий базовые данные, называемый областью построения. Данные, лежащие в основе диаграммы, представлены элементом таблицы. Этот элемент может также существовать для встроенных диаграмм, которые получают данные из документа-контейнера. В этом случае диаграмма может быть визуализирована без получения данных из контейнерного документа.

```
<define name="chart-chart">
    <element name="chart:chart">
        <ref name="chart-chart-attlist"/>
        <optional>
            <ref name="chart-title"/>
        </optional>
        <optional>
            <ref name="chart-subtitle"/>
        </optional>
        <optional>
            <ref name="chart-footer"/>
        </optional>
        <optional>
            <ref name="chart-legend"/>
        </optional>
        <ref name="chart-plot-area"/>
        <optional>
            <ref name="table-table"/>
        </optional>
    </element>
</define>
```

Класс

Атрибут `chart:class` определяет тип диаграммы. Тип диаграммы представлен токеном из пространства имен, означающим идентификатор, которому предшествует XML-префикс пространства имен, подобно любому имени атрибута или элемента в данной спецификации. Спецификацией определено несколько типов диаграмм в пространстве имен диаграммы (URN: urn:oasis:names:tc:opendocument:xmlns:chart:1.0). Также могут использоваться дополнительные типы диаграмм при условии использования другого пространства имен.

Тип диаграммы может быть указан более точно путем изменения свойств, которые могут быть добавлены к стилям диаграммы. Например, трехмерная горизонтальная гистограмма задана с помощью атрибута класса `chart:bar` путем

добавления свойств трехмерности и горизонтального расположения в соответствующий стиль.

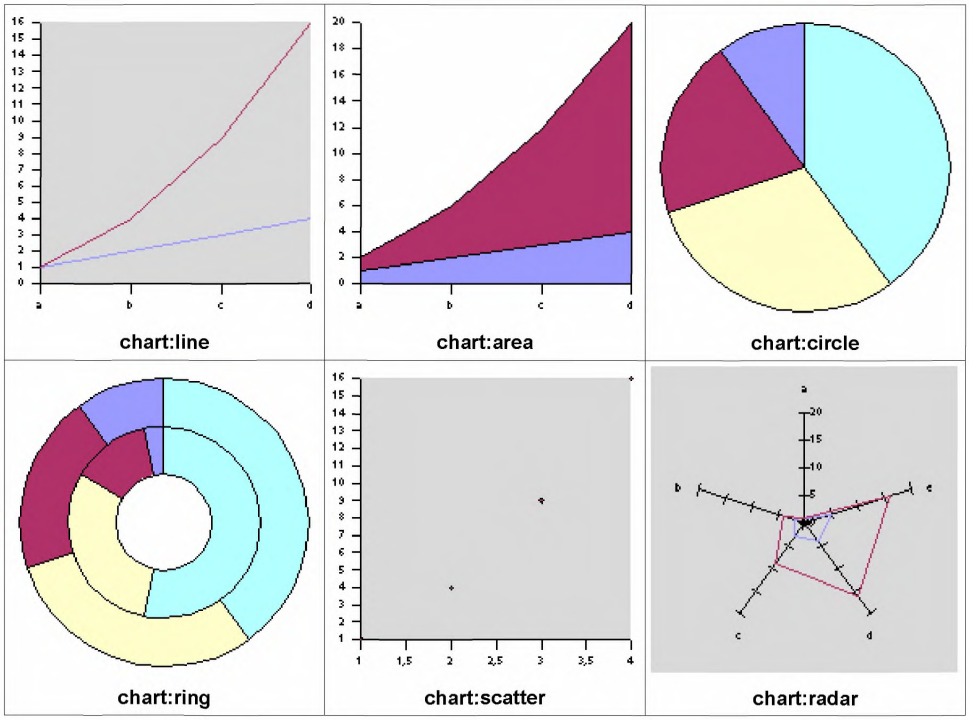
```
<define name="chart-chart-attlist" combine="interleave">
    <attribute name="chart:class">
        <ref name="namespacedToken"/>
    </attribute>
</define>
```

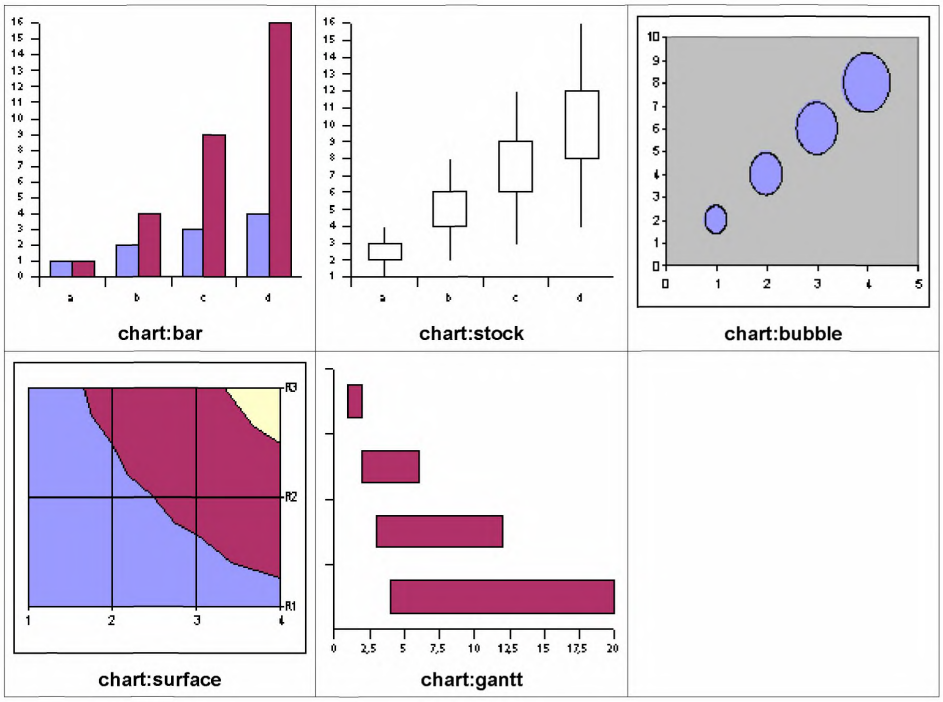
Предопределенные типы диаграмм:

- **линейная** (line) — точки каждого ряда данных соединены линиями;
- **с областями** (area) — область под рядом данных заполнена, а дополнительные ряды данных расположены ярусами;
- **круговая** (circle) — круговая диаграмма поделена на сегменты согласно относительным весам точек данных;
- **кольцевая** (ring) — каждый ряд данных представляет собой концентрические кольца, где каждое кольцо отображено так, будто оно является частью круговой диаграммы;
- **диаграмма XY** (scatter) — пара рядов данных, используемых для определения позиции каждой точки по осям X и Y;
- **лепестковая** (radar) — точки данных в полярной системе координат, где значение каждой точки определяет расстояние от центра диаграммы, причем точки данных серии соединены так, что формируют замкнутую линию вокруг центра;
- **гистограмма** (bar) — каждая точка данных изображена как полоса длиной, пропорциональной значению точки данных;
- **биржевая** (stock) — четыре ряда данных интерпретируются как значение при открытии биржи, минимальное значение, максимальное и значение при закрытии биржи;
- **пузырьковая** (bubble) — первые два или три ряда данных определяют позиции как в диаграмме XY, где размер области каждой точки определяется относительно значения в третьем ряду данных;
- **поверхность** (surface) — точки данных интерпретируются как табулированные данные, где каждое значение определяет высоту в определенном месте сетки, что на графике может быть отображено с помощью цветов для интервалов высот, путем создания цветовых карт подобно географическим картам;

- **диаграмма Ганта** (gannt) — пара рядов данных, которая используется для определения начальной и конечной позиции для горизонтальных полос.

Пример – В следующей таблице показаны примеры предопределенных типов диаграмм. Те диаграммы, которые используют один или два ряда данных, используют два ряда данных со значениями 1;2;3;4 и 1;4;9;16 и метки a;b;c;d. Те типы диаграмм, которые используют больше, чем два ряда данных (биржевая и пузырьковая), используют ряды данных 1;2;3;4 и их комбинации. Лепестковая диаграмма использует два ряда данных с пятью точками.





Размер

Атрибуты `svg:width` и `svg:height` (см. раздел 9.2.15) определяют размер всей диаграммы. Если они опущены, размер диаграммы определяется размером окна, в котором будет отображена диаграмма.

```
<define name="chart-chart-attlist" combine="interleave">
    <ref name="common-draw-size-attlist"/>
</define>
```

Отображение строк и столбцов

Атрибуты `chart:column-mapping` и `chart:row-mapping` содержат, если задан, список индексов рядов данных. Числа определяют переупорядочение данных, которые поступают из документа-контейнера, предоставляющего данные для диаграммы. Нумерация начинается с 1. Упорядоченный по возрастанию список, начинающийся с 1, не вызовет никакого эффекта. Чтобы поменять два ряда, нужно поменять номера в списке. Например, для `1 3 2 4` меняются второй и третий ряды.

Атрибуты chart:column-mapping и chart:row-mapping не должны использоваться одновременно.

```
<define name="chart-chart-attlist" combine="interleave">
    <optional>
        <attribute name="chart:column-mapping">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
<define name="chart-chart-attlist" combine="interleave">
    <optional>
        <attribute name="chart:row-mapping">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Имя стиля

Атрибут chart:style-name ссылается на стиль диаграммы (см. раздел 14.16).

В пределах стиля, применяемого к элементу <chart:chart>, могут использоваться свойства заполнения (описано в разделе 15.14) и свойства контурной линии (описано в разделе 15.13), а также свойство масштабирования текста, описанное в разделе 15.29.1.

```
<define name="chart-chart-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

10.3 Заголовок, подзаголовок и нижний колонтитул

10.3.1 Заголовок

Элемент <chart:title> представляет главный заголовок объекта в документе диаграммы. Этот элемент может содержать фиксированный текст или элемент <table:cell-address>, указывающий на текст, который должен быть отображен как заголовок. Этот элемент может быть подэлементом chart:axis (см. раздел 10.8). В этом случае заголовок будет отображен за пределами осей координат.

```
<define name="chart-title">
    <element name="chart:title">
```

```
        <ref name="chart-title-attlist"/>
        <optional>
            <ref name="text-p"/>
        </optional>
    </element>
</define>
```

Диапазон таблицы

Заголовок диаграммы может быть привязан к ячейке таблицы, поэтому текущее содержимое данной ячейки будет отображено в названии диаграммы.

```
<define name="chart-title-attlist" combine="interleave">
    <optional>
        <attribute name="table:cell-range">
            <ref name="cellAddress"/>
        </attribute>
    </optional>
</define>
```

Положение и размер

Общий атрибут расположения для графических объектов может использоваться в элементах `<chart:title>`.

```
<define name="chart-title-attlist" combine="interleave">
    <ref name="common-draw-position-attlist"/>
</define>
```

Имя стиля

Атрибут `chart:style-name` определяет стиль диаграммы для элемента `<chart:title>`. В данном стиле могут быть использованы свойства заполнения и обрамления. Они применяются к области, окружающей заголовок (см. разделы 15.14 и 15.13 для получения дополнительной информации). В дополнение к этому могут использоваться свойства текста. Они применяются непосредственно к тексту заголовка (см. раздел 15.4).

```
<define name="chart-title-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

10.3.2 Подзаголовок

Элемент `<chart:subtitle>` представляет подзаголовок, который может использоваться для отображения дополнительного заголовка к диаграмме.

Структура элемента `<chart:subtitle>` схожа со структурой элемента `<chart:title>`. С элементом `<chart:subtitle>` могут быть связаны те же атрибуты, что и с элементом `<chart:title>` (см. раздел 10.3.1 для получения дополнительной информации).

```
<define name="chart-subtitle">
    <element name="chart:subtitle">
        <ref name="chart-title-attlist"/>
        <optional>
            <ref name="text-p"/>
        </optional>
    </element>
</define>
```

10.3.3 Нижний колонтитул

Элемент `<chart:footer>` представляет нижний колонтитул под областью построения диаграммы.

Структура элемента подзаголовка та же, что и для элемента заголовка `<chart:title>` (см. раздел 10.3.1 для получения дополнительной информации).

```
<define name="chart-footer">
    <element name="chart:footer">
        <ref name="chart-title-attlist"/>
        <optional>
            <ref name="text-p"/>
        </optional>
    </element>
</define>
```

10.4 Легенда

Элемент `<chart:legend>` определяет, отображается ли легенда на диаграмме. Положение легенды может быть указано как относительное или как абсолютное. Размер легенды вычисляется автоматически и поэтому не может быть установлен через атрибут.

```
<define name="chart-legend">
    <element name="chart:legend">
        <ref name="chart-legend-attlist"/>
        <empty/>
    </element>
</define>
```

Размещение легенды

Легенда может быть помещена автоматически, рядом с областью построения или в одном из углов. Размещение определяется атрибутом

chart:legend-position, который может принимать значения start, end, top, bottom для размещения легенды рядом с областью построения и top-start, bottom-start, top-end или bottom-end для размещения легенды в углах. Если легенда расположена рядом с областью построения в любом из четырех положений start, end, top bottom, дополнительный атрибут выравнивания chart:legend-align определяет, какая граница (start, end) или ось координат (center) легенды и области построения должна быть выровнены.

```
<define name="chart-legend-attlist" combine="interleave">
    <choice>
        <group>
            <attribute name="chart:legend-position">
                <choice>
                    <value>start</value>
                    <value>end</value>
                    <value>top</value>
                    <value>bottom</value>
                </choice>
            </attribute>
            <optional>
                <attribute name="chart:legend-align">
                    <choice>
                        <value>start</value>
                        <value>center</value>
                        <value>end</value>
                    </choice>
                </attribute>
            </optional>
        </group>
        <attribute name="chart:legend-position">
            <choice>
                <value>top-start</value>
                <value>bottom-start</value>
                <value>top-end</value>
                <value>bottom-end</value>
            </choice>
        </attribute>
        <empty/>
    </choice>
</define>
```

Пример – Если chart:legend-position="right", легенда будет размещена справа от области построения диаграммы. Значения chart:legend-align, равные start, center и end, приведут к положению легенды, как изображено зеленым, красным и синим квадратами соответственно.

Положение легенды может быть задано в абсолютных координатах, как для любого отображаемого объекта. Если доступны и положение рисунка, и опции размещения легенды, то размещение легенды имеет приоритет, и позиция будет определена автоматически.

```
<define name="chart-legend-attlist" combine="interleave">
    <ref name="common-draw-position-attlist"/>
</define>
```

Расширение легенды

Для размещения дополнительных пунктов легенда должна быть расширена. Атрибут `style:legend-expansion` определяет, в каком направлении расширяется легенда. Расширение легенды по `wide` и `high` вызывает расширение легенды по горизонтали и вертикали. Расширение `balanced` вызывает расширение в обоих направлениях. Значение расширения `custom` с числовым `style:legend-expansion-aspect-ratio` заставляет легенду расширяться таким образом, что соблюдается отношение между шириной и высотой.

```
<define name="chart-legend-attlist" combine="interleave">
    <choice>
        <attribute name="style:legend-expansion">
            <choice>
                <value>wide</value>
                <value>high</value>
                <value>balanced</value>
            </choice>
        </attribute>
        <group>
            <attribute name="style:legend-expansion">
                <value>custom</value>
            </attribute>
            <attribute name="style:legend-expansion-aspect-ratio">
                <ref name="double"/>
            </attribute>
        </group>
        <empty/>
    </choice>
</define>
```

Стиль легенды

Дополнительная информация о стиле легенды может быть задана через атрибут `chart:style-name`. В стиле могут устанавливаться свойства заполнения и обрамления. Они применимы к легенде (см. разделы 15.14 и 15.13 для получения дополнительной информации). В дополнение к этому, стиль может определить свойства текста. Они применимы к тексту внутри объекта легенды (см. раздел 15.4).

```
<define name="chart-legend-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

10.5 Область построения

Элемент `<chart:plot-area>` является контейнером для графических объектов, которые представляют данные диаграммы. Основное предназначение области построения — быть контейнером для элементов ряда, которые представляют единственный ряд данных и элементы координатной оси.

```
<define name="chart-plot-area">
    <element name="chart:plot-area">
        <ref name="chart-plot-area-attlist"/>
        <zeroOrMore>
            <ref name="dr3d-light"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="chart-axis"/>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="chart-series"/>
        </zeroOrMore>
        <optional>
            <ref name="chart-stock-gain-marker"/>
        </optional>
        <optional>
            <ref name="chart-stock-loss-marker"/>
        </optional>
        <optional>
            <ref name="chart-stock-range-line"/>
        </optional>
        <optional>
            <ref name="chart-wall"/>
        </optional>
        <optional>
            <ref name="chart-floor"/>
        </optional>
    </element>
</define>
```

Позиционирование области построения

Позиция и размер области построения определяются общими атрибутами положения и размера для отображаемых объектов. Если атрибуты положения и размера не указаны, то значения вычисляются визуализирующим приложением.

```
<define name="chart-plot-area-attlist" combine="interleave">
    <ref name="common-draw-position-attlist"/>
    <ref name="common-draw-size-attlist"/>
</define>
```

Стиль области построения

Атрибут `chart:style-name`, который установлен для элемента `<chart:plot-area>`, используется для всех элементов данных, содержащихся в области построения, если не определены дополнительные стили в одном из подэлементов. Эти элементы данных могут быть элементами `<chart:series>` и `<chart:data-point>`.

Если диаграмма является трехмерной, то к области построения могут быть применены трехмерные свойства сцены (см. разделы 15.22—15.26 для получения дополнительной информации).

```
<define name="chart-plot-area-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Атрибуты данных области построения

Если диаграмма встроена в документ, в котором содержатся данные для диаграммы, атрибут `table:cell-range-address` отражает диапазоны, в которых находятся все данные для диаграммы. Заданный здесь диапазон интерпретируется диаграммой как последовательные ряды данных.

```
<define name="chart-plot-area-attlist" combine="interleave">
    <optional>
        <attribute name="table:cell-range-address">
            <ref name="cellRangeAddress"/>
        </attribute>
    </optional>
</define>
```

Если первая строка или столбец, или оба, содержат метки, то это задается атрибутом `chart:data-source-has-labels`.

```
<define name="chart-plot-area-attlist" combine="interleave">
    <optional>
        <attribute name="chart:data-source-has-labels" a:defaultValue="none">
            <choice>
                <value>none</value>
                <value>row</value>
                <value>column</value>
                <value>both</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Свойство форматирования диаграммы chart:series-source, которое описано в разделе 15.34.1, определяет, каким образом расположены данные в таблице: по строкам или по столбцам.

10.5.1 Трехмерная область построения

Область построения может быть отображена как трехмерная сцена, как описано в разделе 9.4.1. Все трехмерные атрибуты, которые могут применяться к элементу <dr3d:scene>, включая атрибут dr3d:transform, могут применяться и к элементу <chart:plot-area>. Он представляет вращение сцены диаграммы, которая является трехмерной областью построения (см. раздел 9.4.1 для получения дополнительной информации). В дополнение к этому, элемент <chart:plot-area> может содержать элемент <dr3d:light>, как описано в разделе 9.4.2

```
<define name="chart-plot-area-attlist" combine="interleave">
    <ref name="dr3d-scene-attlist"/>
    <ref name="common-dr3d-transform-attlist"/>
</define>
```

10.6 Стена

Элемент <chart:wall> может содержаться в элементе <chart:plot-area>. Он устанавливает стены диаграммы. Для двумерных диаграмм стена охватывает всю область построения. Для трехмерных диаграмм стена, как правило, состоит из двух перпендикулярных прямоугольников.

```
<define name="chart-wall">
    <element name="chart:wall">
        <ref name="chart-wall-attlist"/>
        <empty/>
    </element>
</define>
```

Ширина

Атрибут `svg:width` устанавливает ширину стены для трехмерных диаграмм.

```
<define name="chart-wall-attlist" combine="interleave">
    <optional>
        <attribute name="svg:width">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

Стиль

Элемент `<chart:wall>` может иметь атрибут `chart:style-name` для дальнейшего определения стилей оформления, которые могут включать свойства заполнения и обрамления. Для получения более подробной информации см. разделы 15.14 и 15.13.

```
<define name="chart-wall-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

10.7 Основание диаграммы

Элемент `<chart:floor>` может содержаться в элементе `<chart:plot-area>`. Для трехмерных диаграмм элемент `<chart:floor>` представлен в качестве дополнения к элементу `<chart:wall>`.

```
<define name="chart-floor">
    <element name="chart:floor">
        <ref name="chart-floor-attlist"/>
        <empty/>
    </element>
</define>
```

Размер

Размер основания вычисляется на основе размера области построения, которая всегда представлена прямоугольником, служащим в качестве ограничения для трехмерной сцены. Атрибут `svg:width` может использоваться для установления ширины основания.

```
<define name="chart-floor-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="svg:width">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

Стиль

Элемент `<chart:floor>` может иметь атрибут `chart:style-name` для дальнейшего определения стилей оформления. К основанию диаграммы могут быть применены свойства заполнения и обрамления. Для получения более подробной информации см. разделы 15.14 и 15.13.

```
<define name="chart-floor-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

10.8 Оси координат

Элемент `<chart:axis>` в основном содержит данные стиля, в частности — информацию о масштабе. Данные диаграммы обычно структурированы следующим образом:

- несколько рядов данных, состоящих из названий, например названий компании;
- значения, например объем выпуска продукции компанией по годам;
- каждое значение в ряду данных принадлежит определенной категории, например году.

```
<define name="chart-axis">
    <element name="chart:axis">
        <ref name="chart-axis-attlist"/>
        <optional>
            <ref name="chart-title"/>
        </optional>
        <optional>
            <ref name="chart-categories"/>
        </optional>
        <zeroOrMore>
            <ref name="chart-grid"/>
        </zeroOrMore>
    </element>
</define>
```

Измерение

Атрибут chart:dimension определяет, вдоль какой реальной оси на диаграмме, отображаются значения шкалы текущей оси (значения какой шкалы будут отображаться для каждого измерения).

Диаграмма может содержать более одной оси одного и того же измерения. Например, может быть две оси y. Ряды данных могут быть привязаны к каждой из осей. Таким образом, данные могут группироваться по различным измерениям Чтобы привязать специальную ось координат к некоторому ряду данных, для координатной оси должен быть задан элемент <chart:series> с атрибутом chart:axis-name. Если с координатной осью не связан ни один ряд данных, она становится копией существующей оси того же измерения.

Позиция оси на диаграмме определяется визуализирующим приложением и зависит от типа диаграммы. На горизонтальных гистограммах при визуализации обычно отображается ось измерения x в нижней части области построения. Если имеется две оси измерения y, то при формировании изображения вторая ось может быть отображена в верхней части области построения.

```
8702 <define name="chart-axis-attlist" combine="interleave">
8703     <attribute name="chart:dimension">
8704         <choice>
8705             <value>x</value>
8706             <value>y</value>
8707             <value>z</value>
8708         </choice>
8709     </attribute>
8710 </define>
```

Наименование

Атрибут chart:name может использоваться для именования оси, таким образом на него можно ссылаться, например для ряда данных.

```
8711 <define name="chart-axis-attlist" combine="interleave">
8712     <optional>
8713         <attribute name="chart:name">
8714             <ref name="string"/>
8715         </attribute>
8716     </optional>
8717 </define>
```

Стиль

Атрибут chart:style-name может быть связан с осями. Свойство обрамления может применяться к осям (см. раздел 15.13). Эти свойства влияют на

все линии координатной оси. Текстовые свойства также применимы к координатным осям (см. раздел 15.4). Эти свойства влияют на внешний вид всех объектов текстового типа. Также могут быть использованы свойства координатных осей, описанные в разделе 15.31.

Стиль диаграммы, на который ссылается атрибут `chart:style-name` может определять вид отображения данных, используемых для подписей осей. Для получения дополнительной информации см. раздел 14.1

```
<define name="chart-axis-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Пример – Гистограмма.

В этом примере две оси y. Одна из них имеет название `primary-value`. Ряды данных были прикреплены к этой именованной оси. Для второй оси нет прикрепленных данных, поэтому имя оси не было задано, и ось — просто копия первой.

```
<chart:chart chart:class="bar">
    <chart:title>
        <text:p>Title of my chart</text:p>
    </chart:title>
    <chart:plot-area>
        <chart:axis chart:dimension="x"
                    chart:axis-name="x"/>
        <chart:axis chart:dimension="y"
                    chart:axis-name="primary-value"/>
        <chart:axis chart:dimension="y"/>
        <chart:series chart:values-address="Sheet1.A1:.A7"
                    chart:attached-axis="primary-value"/>
    </chart:plot-area>
</chart:chart>
```

10.8.1 Координатная сетка

Элемент `<chart:grid>` может содержаться в элементе `<chart:axis>`. Он добавляет сетку для координатной оси.

```
<define name="chart-grid">
    <element name="chart:grid">
        <ref name="chart-grid-attlist"/>
    </element>
</define>
```

Класс

Атрибут chart:class определяет, используются ли основные или вспомогательные метки деления шкалы. Если используются основные линии сетки, метки деления являются их продолжением. Вспомогательные линии сетки не имеют меток деления.

```
<define name="chart-grid-attlist" combine="interleave">
    <optional>
        <attribute name="chart:class" a:defaultValue="major">
            <choice>
                <value>major</value>
                <value>minor</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Имя стиля

Элемент <chart:grid> может иметь атрибут chart:style-name для дальнейшего определения стилей оформления. Свойства обрамления могут быть применены к координатной сетке, и влияние распространится на линии сетки. Для получения более детальной информации о свойствах обрамления см. раздел 15.13

```
<define name="chart-grid-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

10.9 Ряды данных

Элемент <chart:series> используется для представления рядов данных на диаграмме. Если диаграмма требует больше входных данных, как, например, пузырьковая диаграмма и диаграмма ХУ, то подэлементы <chart:domain> должны быть определены таким образом, чтобы содержать преимущественно атрибуты cell-range-address соответствующих данных.

```
<define name="chart-series">
    <element name="chart:series">
        <ref name="chart-series-attlist"/>
        <zeroOrMore>
            <ref name="chart-domain"/>
        </zeroOrMore>
        <optional>
```

```
            <ref name="chart-mean-value"/>
        </optional>
        <optional>
            <ref name="chart-regression-curve"/>
        </optional>
        <optional>
            <ref name="chart-error-indicator"/>
        </optional>
        <zeroOrMore>
            <ref name="chart-data-point"/>
        </zeroOrMore>
    </element>
</define>
```

Диапазон ячеек

Атрибут `chart:values-cell-range-address` позволяет задать диапазон значений из данного ряда, которые должны быть отображены.

```
<define name="chart-series-attlist" combine="interleave">
    <optional>
        <attribute name="chart:values-cell-range-address">
            <ref name="cellRangeAddress"/>
        </attribute>
    </optional>
</define>
```

Атрибут `chart:label-cell-address` позволяет привязать некоторое имя для ряда данных.

```
<define name="chart-series-attlist" combine="interleave">
    <optional>
        <attribute name="chart:label-cell-address">
            <ref name="cellAddress"/>
        </attribute>
    </optional>
</define>
```

Класс

Атрибут `chart:class` может быть использован для установки типа диаграммы, который будет использован для визуализации данных элемента `<chart:series>`. Атрибут `chart:class` элемента `<chart:series>` переопределяет атрибут `chart:class` для всей диаграммы. Это позволяет создавать диаграммы с множественными субдиаграммами, например одномерная диаграмма с одним или несколькими рядами данных, отображенными в виде линий. Для более подробной информации о доступных классах диаграмм см. раздел 10.2.

```
<define name="chart-series-attlist" combine="interleave">
    <optional>
```

```
8783        <attribute name="chart:class">
8784            <ref name="namespacedToken"/>
8785        </attribute>
8786    </optional>
8787 </define>
```

Привязка координатных осей

Атрибут chart:attached-axis может использоваться для привязки рядов данных к элементу <chart:axis>.

```
8788 <define name="chart-series-attlist" combine="interleave">
8789     <optional>
8790         <attribute name="chart:attached-axis">
8791             <ref name="string"/>
8792         </attribute>
8793     </optional>
8794 </define>
```

Имя стиля

Свойства стиля для рядов данных могут быть назначены при помощи атрибута chart:style-name. Свойства обрамления могут быть применены к элементу <chart:series>. Для получения более детальной информации о свойствах обрамления см. разделы 15.14 и 15.13. Текстовые свойства также применимы к тексту, описывающему ряды данных, расположенному ниже (см. раздел 15.4 для более получения подробной информации).

```
8795 <define name="chart-series-attlist" combine="interleave">
8796     <optional>
8797         <attribute name="chart:style-name">
8798             <ref name="styleNameRef"/>
8799         </attribute>
8800     </optional>
8801 </define>
```

10.9.1 Домен

Для пузырьковых и точечных диаграмм для элемента <chart:series> может быть задан один элемент <chart:domain> или более.

Для диаграмм ХУ один элемент <chart:domain> обязателен. Его атрибут cell-range-address ссылается на значения по координате x для диаграмм ХУ.

Для пузырьковых диаграмм два элемента <chart:domain> обязательны. Их атрибут cell-range-address ссылается на значения по координатам x и y для пузырьковых диаграмм.

Для обоих типов диаграмм должен быть по крайней мере один элемент `<chart:series>` с несколькими субэлементами `<chart:domain>`. Все остальные элементы `<chart:series>` могут не включать вышеперечисленного. В этом случае используется тот домен, который был указан первым.

```
<define name="chart-domain">
    <element name="chart:domain">
    <optional>
        <attribute name="table:cell-range-address">
            <ref name="cellRangeAddress"/>
        </attribute>
    </optional>
    </element>
</define>
```

10.10 Категории

Элемент `<chart:categories>` представляет диапазон адресов ячеек, содержащих подписи к категориям, имеющимbся в каждой последовательности.

Элемент может содержать атрибут `table:cell-range-address`, отмечающий область, из которой были взяты подписи к категориям. Если этот атрибут или элемент `<chart:categories>` пропущены, приложение использует атрибут `chart:data-source-has-labels`.

```
<define name="chart-categories">
    <element name="chart:categories">
    <optional>
        <attribute name="table:cell-range-address">
            <ref name="cellRangeAddress"/>
        </attribute>
    </optional>
    </element>
</define>
```

10.11 Точка данных

Если отдельная точка ряда данных требует особого отображения, то для применения необходимых свойств используется элемент `<chart:data-point>`.

```
<define name="chart-data-point">
    <element name="chart:data-point">
        <ref name="chart-data-point-attlist"/>
        <empty/>
    </element>
</define>
```

Повтор

Атрибут `chart:repeated` служит для упрощения в том случае, если несколько последовательных точек имеют одинаковые свойства. Например, нижеследующие XML-фрагменты имеют одинаковый смысл:

```
<chart:series chart:style-name="ch9">
    <chart:data-point/>
    <chart:data-point/>
    <chart:data-point/>
    <chart:data-point/>
</chart:series>
```

и

```
<chart:series chart:style-name="ch9">
    <chart:data-point chart:repeated="4"/>
</chart:series>
```

```
<define name="chart-data-point-attlist" combine="interleave">
    <optional>
        <attribute name="chart:repeated">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Стиль

Атрибут `chart:style-name` относится к стилю диаграммы. Свойства заполнения и обрамления применяются к каждой точке данных (см. разделы 15.14 и 15.13). Текстовые свойства также могут быть применены к подписи, расположенной непосредственно под значением (см. раздел 15.4).

```
<define name="chart-data-point-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

10.12 Среднее значение

Свойства форматирования линии среднего значения содержатся в элементе `<chart:mean-value>`, который может являться частью элемента `<chart:series>`.

```
<define name="chart-mean-value">
    <element name="chart:mean-value">
        <ref name="chart-mean-value-attlist"/>
        <empty/>
```

```
    </element>
</define>
```

Имя стиля

Атрибут `chart:style-name` указывает на стиль диаграммы, содержащий свойства форматирования для линии среднего значения.

```
<define name="chart-mean-value-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

10.13 Индикатор ошибок

Свойства форматирования индикаторов ошибок содержатся в элементе `<chart:error-indicator>`, который может быть частью рядов данных.

```
<define name="chart-error-indicator">
    <element name="chart:error-indicator">
        <ref name="chart-error-indicator-attlist"/>
        <empty/>
    </element>
</define>
```

Имя стиля

Атрибут `chart:style-name` указывает на стиль диаграммы, содержащий свойства форматирования индикатора ошибок.

```
<define name="chart-error-indicator-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

10.14 Кривые регрессии

Свойства форматирования кривых регрессии содержатся в элементе `<chart:regression-curve>`, который может быть частью рядов данных.

```
<define name="chart-regression-curve">
    <element name="chart:regression-curve">
        <ref name="chart-regression-curve-attlist"/>
        <empty/>
    </element>
</define>
```

Имя стиля

Атрибут chart:style-name указывает на стиль диаграммы, содержащий свойства форматирования индикатора ошибок. Стиль диаграммы может, главным образом, содержать свойство «вид регрессии», описанное в разделе 15.35.1.

```
<define name="chart-regression-curve-attlist" combine="interleave">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

10.14.1 Метки биржевой диаграммы

Свойства биржевой диаграммы, как например, различные цвета заливки свечей или стили линий наибольшего и наименьшего значения (линия диапазона), находятся в отдельных элементах.

Свечи биржевых диаграмм, у которых значение закрытия располагаются выше, чем значение открытия, используют свойства форматирования элемента <chart:stock-gain-marker>, в то время как биржевые диаграммы, у которых значение закрытия располагаются ниже, чем значение открытия, используют свойства элемента chart:stock-loss-marker>.

```
<define name="chart-stock-gain-marker">
    <element name="chart:stock-gain-marker">
        <ref name="common-stock-marker-attlist"/>
    </element>
</define>
<define name="chart-stock-loss-marker">
    <element name="chart:stock-loss-marker">
        <ref name="common-stock-marker-attlist"/>
    </element>
</define>
<define name="chart-stock-range-line">
    <element name="chart:stock-range-line">
        <ref name="common-stock-marker-attlist"/>
    </element>
</define>
```

Имя стиля

Атрибут стиля диаграммы chart:style-name определяет свойства оформления меток биржевых диаграмм.

```
<define name="common-stock-marker-attlist">
    <optional>
        <attribute name="chart:style-name">
            <ref name="styleNameRef"/>
```

```
        </attribute>
    </optional>
</define>
```

11 Содержимое форм

Форма — это контейнер для элементов управления пользовательского интерфейса, с которыми взаимодействует пользователь. Например, кнопки, текстовые поля, переключатели, выпадающие списки — элементы управления, которые могут содержаться в форме. В XML существуют следующие основные правила для элементов управления и форм:

- все элементы управления должны быть расположены в форме;
- все нескрытые элементы управления должны быть связаны с абсолютной или относительной позицией, этот визуальный аспект элементов управления представлен графическими фигурами, связанными с элементом управления (см. раздел 9.2.12 для получения дополнительной информации);
- формы могут быть вложенными;
- формы не связаны с текстовым потоком и разметкой документа, это не распространяется на элементы управления;
- формы могут быть привязанными к данным, элементы управления отражают содержимое базы данных.

Формы определяют следующие правила поведения:

- передача форм подобна использующейся в [HTML4];

  Примечание — Передача поддерживается только для невложенных форм, которые содержат только элементы управления, которые могут быть преобразованы в HTML.
- соединение с источником данных, когда это происходит, элементы управления в форме становятся привязанными к данным;
- передача и привязка согласно модели данных [XForms].

Формы содержатся в секции `<office:forms>` XML-документа. Этот элемент может содержать произвольную последовательность элементов `<form:form>` или `<xforms:model>`. Обратите внимание на то, что элементы управления всегда объявляются внутри элемента `<form:form>`, в то время как элемент `<xforms:model>` содержит только XForms модели данных. Таким образом, элемент `<office:forms>` может содержать только элементы `<form:form>`, но

не элементы `<xforms:model>`, в то время как `<xforms:model>` обычно сопровождается дополнительным элементом `<form:form>`.

```
<define name="office-forms">
    <optional>
        <element name="office:forms">
            <ref name="office-forms-attlist"/>
            <zeroOrMore>
                <choice>
                    <ref name="form-form"/>
                    <ref name="xforms-model"/>
                </choice>
            </zeroOrMore>
        </element>
    </optional>
</define>
```

Для простоты использования (заполнения) форм приложения могут фокусироваться на элементах управления, чтобы пользователь мог немедленно напечатать, внести данные в первый элемент управления. Чтобы достичь такого поведения, флаг `form:automatic-focus` может иметь значение `true`.

```
<define name="office-forms-attlist" combine="interleave">
    <optional>
        <attribute name="form:automatic-focus" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

В приложениях, поддерживающих создание и использование (заполнение) форм, атрибут `form:apply-design-mode` определяет, предоставляет ли приложение формы в данном документе в доступном для редактирования или заполнения виде.

```
<define name="office-forms-attlist" combine="interleave">
    <optional>
        <attribute name="form:apply-design-mode" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.1 Форма

Элемент `<form:form>` представляет форму пользовательского интерфейса и определяет содержание и свойства форм.

Этот элемент содержится или в элементе `<office:forms>` или в элементе `<form:form>`. Он также содержит элементы управления, субформы, элемент

<form:properties>, который определяет свойства формы, и элемент <office:events-listeners>, содержащий события формы.

```
<define name="form-form">
    <element name="form:form">
        <ref name="common-form-control-attlist"/>
        <ref name="form-form-attlist"/>
        <optional>
            <ref name="form-properties"/>
        </optional>
        <optional>
            <ref name="office-event-listeners"/>
        </optional>
        <zeroOrMore>
            <choice>
                <ref name="controls"/>
                <ref name="form-form"/>
            </choice>
        </zeroOrMore>
        <optional>
            <ref name="form-connection-resource"/>
        </optional>
    </element>
</define>
```

С элементом <form:form> могут быть связаны следующие атрибуты:

- имя (см. раздел 11.4);
- имя службы (см. раздел 11.4);
- действие;
- целевой фрейм;
- метод;
- тип кодировки;
- разрешение удаления;
- разрешение вставки;
- разрешение обновления;
- применение фильтра;
- тип команды;
- команда;
- источник данных;
- мастер-поля;
- поля деталей;
- избежать обработку команды;
- фильтр;
- игнорирование результата;
- режим навигации;

- порядок;
- циклы табуляции.

11.1.1 Действие

Атрибут `xlink:href` представляет IRI агента обработки формы.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="xlink:href">
            <ref name="anyURI"/>
        </attribute>
        <optional>
            <attribute name="xlink:type" a:defaultValue="simple">
                <value>simple</value>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:actuate" a:defaultValue="onRequest">
                <value>onRequest</value>
            </attribute>
        </optional>
    </optional>
</define>
```

11.1.2 Целевой фрейм

Атрибут `office:target-frame` определяет целевой фрейм формы.

Этот атрибут может иметь одно из следующих значений:

- `_self` — форма заменяет содержимое текущего фрейма;
- `_blank` — форма открывается в новом фрейме;
- `_parent` — форма открывается в родительском фрейме текущего фрейма;
- `_top` — форма отображается в самом верхнем фрейме, это фрейм, который содержит текущей фрейм в качестве дочернего, но который не содержится внутри никакого другого фрейма;
- название фрейма — форма отображается в заданном фрейме, если фрейм с таким именем отсутствует, будет создан новый фрейм с заданным именем.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="office:target-frame" a:defaultValue="_blank">
            <ref name="targetFrameName"/>
        </attribute>
    </optional>
</define>
```

11.1.3 Метод

Атрибут form:method определяет HTTP-метод для передачи данных из формы на сервер. Значения этого атрибута могут быть get или post. Значение по умолчанию равно get. Эти значения нечувствительны к регистру.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:method" a:defaultValue="get">
            <choice>
                <value>get</value>
                <value>post</value>
                <ref name="string"/>
            </choice>
        </attribute>
    </optional>
</define>
```

11.1.4 Тип кодировки

Если значение атрибута form:method равно post, то атрибут form:enctype определяет тип содержимого, используемого для отправки содержимого формы на сервер. По умолчанию значение этого атрибута равно application/x-www-form-urlencoded. Другие подходящие MIME-типы также разрешаются.

Для получения дополнительной информации см. § 17.3 [HTML4] .

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:enctype"
                    a:defaultValue="application/x-www-form-urlencoded">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

11.1.5 Разрешение удаления

Атрибут form:allow-deletes определяет, разрешается ли удаление записей данных. Этот атрибут применяется, только если форма связана с данными.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:allow-deletes" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.1.6 Разрешение вставок

Атрибут `form:allow-inserts` определяет, могут или нет вставляться новые записи данных. Этот атрибут применяется, только если форма связана с данными.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:allow-inserts" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.1.7 Разрешение обновлений

Атрибут `form:allow-updates` определяет, могут ли изменяться записи данных.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:allow-updates" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.1.8 Применение фильтра

Атрибут `form:apply-filter` определяет, нужно ли применять фильтр (см. также атрибут фильтра).

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:apply-filter" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.1.9 Тип команды

Атрибут `form:command-type` определяет тип команды выполняемой над источником данных. Этот атрибут может принимать одно из следующих значений:

- `table` — команда содержит имя таблицы, форма извлекает все данные в таблице;
- `query` — команда содержит имя запроса, форма извлекает и выполняет запрос;

- `command` — команда содержит SQL-выражение, форма выполняет SQL-выражение.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:command-type" a:defaultValue="command">
            <choice>
                <value>table</value>
                <value>query</value>
                <value>command</value>
            </choice>
        </attribute>
    </optional>
</define>
```

11.1.10 Команда

Атрибут `form:command` определяет команду выполняемую над источником данных.

Это значение интерпретируется по разному, в зависимости от значения атрибута типа команды формы. Значение может быть именем таблицы базы данных, именем объекта запроса или SQL-выражения.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:command"/>
    </optional>
</define>
```

11.1.11 Источник данных

Атрибут `form:datasource` определяет имя источника данных, используемых для формы.

Этот атрибут может принимать следующие значения:

- URL, определяющий подключение к базе данных;
- имя источника данных, которое офисное приложение может использовать для установления соединений с базой данных.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:datasource">
            <choice>
                <ref name="anyURI"/>
                <ref name="string"/>
            </choice>
        </attribute>
    </optional>
</define>
```

11.1.12 Мастер-поля

Атрибут form:master-fields используется для вложенных форм, связанных с данными. Он определяет названия столбцов результирующего набора в родительской форме. Обычно они обозначают поля внешних ключей в родительской форме. Значения столбцов используются для параметризации данных вложенной формы. Каждый раз, когда родительская форма изменяет текущую строку, вложенная форма снова опрашивает базу данных, базируясь на значениях мастер-полей.

Атрибут содержит список названий полей через запятую.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:master-fields">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

11.1.13 Поля деталей

Атрибут form:detail-fields используется для вложенных форм базы данных. Он определяет названия столбцов во вложенной форме, которые связаны со столбцами в родительской форме. Столбцы используются как параметры в команде вложенной формы для извлечения деталей для подходящей записи мастер-формы.

Атрибут содержит список названий полей через запятую.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:detail-fields">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

11.1.14 Избежать обработку команды

Если значение атрибута form:command-type равно command, атрибут form:escape-processing определяет, может ли приложение выполнять команду до передачи ее драйверу базы данных.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:escape-processing" a:defaultValue="true">
            <ref name="boolean"/>
```

```
        </attribute>
    </optional>
</define>
```

11.1.15 Фильтр

Атрибут `form:filter` определяет фильтр для команды, на которой основывается форма. Фильтр всегда добавляется к любому, возможно существующему, фильтру вне зависимости от того, базируется ли форма на `query`, `table` или `command`. Фильтр обычно формирует SQL-выражение типа «WHERE» без ключевого слова «WHERE».

Атрибут `form:apply-filter` определяет, применяется ли фактически фильтр для команды.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:filter">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

11.1.16 Игнорирование результата

Атрибут `form:ignore-result` определяет, отбрасываются ли результаты, которые извлечены из источника данных. Если его значение равно `true`, форма, связанная с базой данных, отбросит все результаты запросов к базе данных, таким образом, будут доступны только операции вставки и модификации новых данных. По существу, реализуется режим, в котором может быть осуществлена только вставка новых данных в базу данных.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:ignore-result" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.1.17 Режим навигации

Атрибут `form:navigation-mode` определяет, как осуществляется навигация по записям базы данных.

Значение этого атрибута может быть одним из следующих:

- none — специальная навигационная панель не предоставляется пользовательским интерфейсом, навигация по форме осуществляется с использованием клавиши TAB и SHIFT/TAB на клавиатуре;
- current — специальная навигационная панель предоставляется и осуществляется навигация по текущей формой;
- parent — специальная навигационная панель предоставляется и осуществляется навигация по родительской форме текущей формы.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:navigation-mode">
            <ref name="navigation"/>
        </attribute>
    </optional>
</define>

<define name="navigation">
    <choice>
        <value>none</value>
        <value>current</value>
        <value>parent</value>
    </choice>
</define>
```

11.1.18 Порядок

Атрибут form:order определяет критерий сортировки для команды. Сортировка всегда добавляется к любой, возможно существующей, сортировке вне зависимости от того, базируется ли форма на query, table или command. Значение атрибута обычно формирует SQL-выражение типа «ORDER BY» без ключевого слова «ORDER BY».

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:order">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

11.1.19 Цикл табуляции

Атрибут form:tab-cycle определяет, как реагирует приложение на нажатие пользователем клавиши TAB в элементе управления формы. Поведение приложения зависит от того, будет ли форма связана с источником данных.

Этот атрибут может принимать следующие значения:

- `records` — если пользователь нажимает клавишу TAB в последнем элементе управления формы, фокус ввода перемещается на первый элемент управления, указанный в списке табуляции этой формы и перемещает форму к следующей записи;
- `current` — если пользователь нажимает клавишу TAB в последнем элементе управления формы, фокус ввода перемещается на первый элемент управления, указанный в списке табуляции этой формы, при этом указатель на запись формы не изменяется;
- `page` — если пользователь нажимает клавишу TAB в последнем элементе управления формы, фокус ввода перемещается на первый элемент управления, указанный в списке табуляции следующей формы.

```
<define name="form-form-attlist" combine="interleave">
    <optional>
        <attribute name="form:tab-cycle">
            <ref name="tab-cycles"/>
        </attribute>
    </optional>
</define>
<define name="tab-cycles">
    <choice>
        <value>records</value>
        <value>current</value>
        <value>page</value>
    </choice>
</define>
```

11.1.20 Ресурс соединения

Элемент `<form:connection-resource>` определяет источник базы данных в соответствии с [XLink]. Его атрибут `xlink:href` либо ссылается на файл, содержащий базу данных, либо содержит информацию о том, как произвести соединение с базой данных, например [JDBC] URL.

```
<define name="form-connection-resource">
    <element name="form:connection-resource">
        <attribute name="xlink:href">
            <ref name="anyURI"/>
        </attribute>
        <empty/>
    </element>
</define>
```

11.2 Модель XForms

Модель формы, описанная в разделе 11.1, реализует модель данных, где каждый элемент управления определяет пару имя — значение, где имя

определяется идентификатором элемента, а значение — может редактироваться элементом управления. При этом отсутствует взаимосвязь между элементами управления (для программирования макросов). Для приложений, в которых данный вид форм логически недостаточен, организация W3C ввела XForms (см. [XForms]), стандарт для форм, основанных на XML.

XForms спроектированы для встраивания в другой XML-формат. Он состоит из двух основных частей: модель XForms, которая содержит логику формы и данные формы, и элементы управления XForms, которые могут быть связаны с моделью данных. В OASIS Open Office 1.0 модель W3C XForms была встроена как элемент `<xforms:model>`, определенный в контейнере формы `<office:forms>`. Элементы управления (см. раздел 11.3) будут оставлены как есть, за исключением того, что они получают атрибут `xforms:bind`, который позволяет связать любой элемент управления OpenDocument с ранее описанной моделью XForms.

11.2.1 Модель XForms (импорт)

Импортирована модель XForms, описанная в [XForms]. Для предотвращения появления дубликатов схемы XForms в данной спецификации только определены элементы модели XForms и допущено произвольное содержимое.

```
<define name="xforms-model">
    <element name="xforms:model">
        <ref name="anyAttListOrElements"/>
    </element>
</define>
```

11.3 Элементы управления

Элементы управления используются для взаимодействия с формами. Каждый элемент управления определяется именем, однако имена не обязательно должны быть уникальными.

Элементы управления присоединяются к окружающему документу (и если возможно к потоку текста) путем присоединения их к элементам, которые являются полем для заполнения для элемента управления. Для получения подробной информации см. раздел 9.2.12.

В дополнение к атрибутам, определенным в формате файла, элементы управления могут иметь атрибуты, специфические для приложения. Эти атрибуты

хранятся в элементе <form:properties> для каждого элемента управления. События элементов управления описываются в элементе <office:event-listeners>.

Когда пользователь отсылает данные формы на обработку, имена некоторых элементов управления спарены со значениями элементов управления, эти пары и передаются. Такие элементы управления называются успешными элементами управления. Для получения подробной информации см. раздел 17.13.2 в [HTML4].

Формат файла предусматривает элементы для следующих стандартных элементов управления:

- текст;
- текстовая область;
- пароль;
- файл;
- форматированный текст;
- число;
- дата;
- время;
- фиксированный текст;
- поле со списком;
- список;
- кнопка;
- изображение;
- флажок;
- радиокнопка;
- фрейм;
- графический фрейм;
- скрытый;
- сетка.

Также возможно определение элементов управления, специфичных для приложений. Эти элементы управления описываются элементом <form:generic-control>.

11.3.1 Текст

Элемент `<form:text>` определят элемент управления, предназначенный для отображения и ввода текста.

```
<define name="column-controls" combine="choice">
    <element name="form:text">
        <ref name="form-text-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="controls" combine="choice">
    <ref name="column-controls"/>
</define>
<define name="form-text-attlist">
    <ref name="form-control-attlist"/>
    <ref name="common-current-value-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="common-maxlength-attlist"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-readonly-attlist"/>
    <ref name="common-tab-attlist"/>
    <ref name="common-title-attlist"/>
    <ref name="common-value-attlist"/>
    <ref name="common-convert-empty-attlist"/>
    <ref name="common-data-field-attlist"/>
</define>
<define name="form-control-attlist">
    <ref name="common-form-control-attlist"/>
    <ref name="common-control-id-attlist"/>
    <ref name="xforms-bind-attlist"/>
</define>
<define name="common-form-control-content">
    <optional>
        <ref name="form-properties"/>
    </optional>
    <optional>
        <ref name="office-event-listeners"/>
    </optional>
</define>
```

С элементом `<form:text>` могут быть связаны следующие атрибуты:

- имя и имя службы (для получения подробной информации об этих атрибутах см. раздел 11.4);
- идентификатор элемента управления, текущее значение, запрещен, максимальная длина, возможность печати, только для чтения, индекс табуляции, остановка табуляции, заголовок и значение (для получения подробной информации об этих атрибутах см. 11.5);
- преобразование пустого значения и поле данных (для получения подробной информации об этих атрибутах см. раздел 11.5.22).

11.3.2 Текстовая область

Элемент `<form:textarea>` определяет элемент управления для отображения и ввода многострочного текста.

Элемент `<form:textarea>` может использоваться с простыми текстовыми значениями (определенных атрибутом `form:current-value`) так же, как и с форматированным текстом (определенным как содержимое абзаца). В обоих случаях, когда `form:current-value` и один или больше `<text:p>` элементов присутствуют, на приложение, считывающее документ, возлагается решение, какую информацию использовать.

```
<define name="column-controls" combine="choice">
    <element name="form:textarea">
        <ref name="form-textarea-attlist"/>
        <ref name="common-form-control-content"/>
        <zeroOrMore>
            <ref name="text-p"/>
        </zeroOrMore>
    </element>
</define>
<define name="form-textarea-attlist">
    <ref name="form-control-attlist"/>
    <ref name="common-current-value-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="common-maxlength-attlist"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-readonly-attlist"/>
    <ref name="common-tab-attlist"/>
    <ref name="common-title-attlist"/>
    <ref name="common-value-attlist"/>
    <ref name="common-convert-empty-attlist"/>
    <ref name="common-data-field-attlist"/>
</define>
```

С элементом `<form:textarea>` могут быть связаны следующие атрибуты:

- имя и имя службы (для получения более подробной информации об этих атрибутах см. раздел 11.4);
- идентификатор элемента управления, текущее значение, запрещен, максимальная длина, возможность печати, индекс табуляции, остановка табуляции, заголовок и значение (для получения подробной информации об этих атрибутах см. раздел 11.5).
- преобразование пустого значения и поле данных (для получения подробной информации об этих атрибутах см. раздел 11.5.22).

11.3.3 Пароль

Элемент `<form:password>` определяет элемент управления, который скрывает вводимый пользователем текст, используя специальный символ, например, звездочка. Этот тип элемента управления обычно используется для ввода важной информации, такой как пароль.

```
<define name="controls" combine="choice">
    <element name="form:password">
        <ref name="form-password-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="form-password-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="common-maxlength-attlist"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-tab-attlist"/>
    <ref name="common-title-attlist"/>
    <ref name="common-value-attlist"/>
    <ref name="common-convert-empty-attlist"/>
</define>
```

С элементом `<form:password>` могут быть связаны следующие атрибуты:

- имя и имя службы (для получения подробной информации об этих атрибутах см. раздел 11.4);
- идентификатор элемента управления, запрещен, максимальная длина, возможность печати, индекс табуляции, остановка табуляции, заголовок и значение (для получения подробной информации об этих атрибутах см. раздел 11.5);
- отображаемый символ.

Отображаемый символ

Атрибут `form:echo-char` определяет символ, используемый формой для маскировки текста, который вводит пользователь в элемент управления типа пароль.

```
<define name="form-password-attlist" combine="interleave">
    <optional>
        <attribute name="form:echo-char" a:defaultValue="*">
            <ref name="character"/>
        </attribute>
    </optional>
</define>
```

11.3.4 Файл

Элемент `<form:file>` определяет элемент управления для выбора файла.

```
<define name="controls" combine="choice">
    <element name="form:file">
        <ref name="form-file-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="form-file-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
    <ref name="common-current-value-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="common-maxlength-attlist"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-readonly-attlist"/>
    <ref name="common-tab-attlist"/>
    <ref name="common-title-attlist"/>
    <ref name="common-value-attlist"/>
</define>
```

С элементом `<form:file>` могут быть связаны следующие атрибуты:

- имя и имя службы (для получения более подробной информации об этих атрибутах см. раздел 11.4);
- идентификатор элемента управления, текущее значение, запрещен, возможность печати, только для чтения, индекс табуляции, остановка табуляции, заголовок и значение (для получения подробной информации об этих атрибутах см. раздел 11.5).

11.3.5 Форматированный текст

Элемент `<form:formatted-text>` определяет элемент управления для ввода форматированного текста, который позволяет осуществлять форматирование и для отображения, и для редактирования.

```
<define name="column-controls" combine="choice">
    <element name="form:formatted-text">
        <ref name="form-formatted-text-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="form-formatted-text-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
    <ref name="common-current-value-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="common-maxlength-attlist"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-readonly-attlist"/>
    <ref name="common-tab-attlist"/>
    <ref name="common-title-attlist"/>
    <ref name="common-value-attlist"/>
    <ref name="common-convert-empty-attlist"/>
```

```
    <ref name="common-data-field-attlist"/>
</define>
```

С элементом `<form:formatted-text>` могут быть связаны следующие атрибуты:

- имя и имя службы (для получения подробной информации об этих атрибутах см. раздел 11.4);
- идентификатор элемента управления, текущее значение, запрещен, максимальная длина, возможность печати, только для чтения, индекс табуляции, остановка табуляции, заголовок и значение (для получения подробной информации об этих атрибутах см. раздел 11.5);
- преобразование пустого значения и поле данных (для получения подробной информации об этих атрибутах см. раздел 11.5.22);
- максимальное значение;
- минимальное значение;
- проверка.

Максимальное значение

Атрибут `form:max-value` определяет максимальное значение, которое может ввести пользователь.

```
<define name="form-formatted-text-attlist" combine="interleave">
    <optional>
        <attribute name="form:max-value">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Минимальное значение

Атрибут `form:min-value` определяет минимальное значение, которое может ввести пользователь.

```
<define name="form-formatted-text-attlist" combine="interleave">
    <optional>
        <attribute name="form:min-value">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Проверка

Атрибут form:validation определяет, будет ли текст, вводимый пользователем, проверяться во время ввода.

```
<define name="form-formatted-text-attlist" combine="interleave">
    <optional>
        <attribute name="form:validation" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.3.6 Число

Элемент <form:number> описывает элемент управления, который позволяет пользователю вводить числа с плавающей запятой. Атрибуты, которые могут быть связаны с этим элементом управления, схожи с атрибутами элемента <form:formatted-text> за исключением того, что могут использоваться только числовые типы данных.

```
<define name="column-controls" combine="choice">
    <element name="form:number">
        <ref name="form-number-attlist"/>
        <ref name="common-numeric-control-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="common-numeric-control-attlist">
    <ref name="form-control-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="common-maxlength-attlist"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-readonly-attlist"/>
    <ref name="common-tab-attlist"/>
    <ref name="common-title-attlist"/>
    <ref name="common-convert-empty-attlist"/>
    <ref name="common-data-field-attlist"/>
</define>
```

С элементом <form:number> могут быть связаны следующие атрибуты:

- имя и имя службы (для получения подробной информации об этих атрибутах см. раздел 11.4);
- идентификатор элемента управления, запрещен, максимальная длина, возможность печати, только для чтения, индекс табуляции, остановка табуляции, заголовок и значение (для получения подробной информации об этих атрибутах см. раздел 11.5);
- преобразование пустого значения и поле данных (для получения подробной информации об этих атрибутах см. раздел 11.5.22);

- значение и текущее значение;
- минимальное и максимальное значение.

Значение

Атрибуты для значения и текущего значения общие, схожи для всех полей, за исключением того, что они могут использоваться только с данными с плавающей точкой.

```
<define name="form-number-attlist" combine="interleave">
    <optional>
        <attribute name="form:value">
            <ref name="double"/>
        </attribute>
    </optional>
</define>
<define name="form-number-attlist" combine="interleave">
    <optional>
        <attribute name="form:current-value">
            <ref name="double"/>
        </attribute>
    </optional>
</define>
```

Минимум и максимум

Атрибуты для минимального и максимального значения определяют наименьшее и наибольшее числовое значения, которые применимы для данного элемента управления.

```
<define name="form-number-attlist" combine="interleave">
    <optional>
        <attribute name="form:min-value">
            <ref name="double"/>
        </attribute>
    </optional>
</define>
<define name="form-number-attlist" combine="interleave">
    <optional>
        <attribute name="form:max-value">
            <ref name="double"/>
        </attribute>
    </optional>
</define>
```

11.3.7 Дата и время

Элементы управления для даты и времени схожи с числовыми, за исключением того, что в них используются значения даты и времени соответственно. Они поддерживают те же атрибуты, что и числовые поля, за исключением атрибутов, отвечающих за разные типы данных.

```
<define name="column-controls" combine="choice">
    <element name="form:date">
        <ref name="form-date-attlist"/>
        <ref name="common-numeric-control-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="controls" combine="choice">
    <element name="form:time">
        <ref name="form-time-attlist"/>
        <ref name="common-numeric-control-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
```

С элементами `<form:date>` и `<form:time>` могут быть связаны следующие атрибуты:

- имя и имя службы (для получения подробной информации об этих атрибутах см. раздел 11.4);
- идентификатор элемента управления, запрещен, максимальная длина, возможность печати, только для чтения, индекс табуляции, остановка табуляции, заголовок и значение (для получения подробной информации об этих атрибутах см. раздел 11.5);
- преобразование пустого значения и поле данных (для получения подробной информации об этих атрибутах см. раздел 11.5.22);
- значение и текущее значение;
- минимальное и максимальное значения.

Значение

Атрибуты для значения и текущего значения схожи с атрибутами для `<form:number>`, за исключением того, что они могут содержать только дату или время соответственно.

```
<define name="form-date-attlist" combine="interleave">
    <optional>
        <attribute name="form:value">
            <ref name="date"/>
        </attribute>
    </optional>
</define>
<define name="form-time-attlist" combine="interleave">
    <optional>
        <attribute name="form:value">
            <ref name="time"/>
        </attribute>
    </optional>
</define>
<define name="form-date-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="form:current-value">
            <ref name="date"/>
        </attribute>
    </optional>
</define>
<define name="form-time-attlist" combine="interleave">
    <optional>
        <attribute name="form:current-value">
            <ref name="time"/>
        </attribute>
    </optional>
</define>
```

Минимум и максимум

Атрибуты для минимального и максимального значения определяют наименьшую и наибольшую дату (или время), которые применимы для данного элемента управления.

```
<define name="form-date-attlist" combine="interleave">
    <optional>
        <attribute name="form:min-value">
            <ref name="date"/>
        </attribute>
    </optional>
</define>
<define name="form-time-attlist" combine="interleave">
    <optional>
        <attribute name="form:min-value">
            <ref name="time"/>
        </attribute>
    </optional>
</define>
<define name="form-date-attlist" combine="interleave">
    <optional>
        <attribute name="form:max-value">
            <ref name="date"/>
        </attribute>
    </optional>
</define>
<define name="form-time-attlist" combine="interleave">
    <optional>
        <attribute name="form:max-value">
            <ref name="time"/>
        </attribute>
    </optional>
</define>
```

11.3.8 Фиксированный текст

Элемент `<form:fixed-text>` описывает элемент управления, который устанавливает дополнительную информацию для элементов управления или только передают информацию в приложение. Связь между метками и отмеченными элементами управления может быть установлена определением

атрибута form:for для метки. Только одна метка может быть связана с данным элементом управления.

```
<define name="controls" combine="choice">
    <element name="form:fixed-text">
        <ref name="form-fixed-text-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="form-fixed-text-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
    <ref name="for"/>
    <ref name="common-disabled-attlist"/>
    <ref name="label"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-title-attlist"/>
</define>
```

С элементом <form:fixed-text> могут быть связаны следующие атрибуты:

- имя и имя службы (для получения подробной информации об этих атрибутах см. раздел 11.4);
- идентификатор элемента управления, запрещен, для, метка, возможность печати, заголовок (для получения подробной информации об этих атрибутах см. раздел 11.5);
- многострочность.

Многострочность

Атрибут form:multi-line определяет, отображается ли текстовая метка в несколько строк.

```
<define name="form-fixed-text-attlist" combine="interleave">
    <optional>
        <attribute name="form:multi-line" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.3.9 Поле со списком

Элемент <form:combobox> описывает элемент управления, который позволяет отображать и редактировать текст, а также содержит список возможных значений текста.

```
<define name="column-controls" combine="choice">
    <element name="form:combobox">
        <ref name="form-combobox-attlist"/>
```

```
9408        <ref name="common-form-control-content"/>
9409        <zeroOrMore>
9410            <ref name="form-item"/>
9411        </zeroOrMore>
9412    </element>
9413 </define>
9414 <define name="form-combobox-attlist" combine="interleave">
9415    <ref name="form-control-attlist"/>
9416    <ref name="common-current-value-attlist"/>
9417    <ref name="common-disabled-attlist"/>
9418    <ref name="dropdown"/>
9419    <ref name="common-maxlength-attlist"/>
9420    <ref name="common-printable-attlist"/>
9421    <ref name="common-readonly-attlist"/>
9422    <ref name="size"/>
9423    <ref name="common-tab-attlist"/>
9424    <ref name="common-title-attlist"/>
9425    <ref name="common-value-attlist"/>
9426    <ref name="common-convert-empty-attlist"/>
9427    <ref name="common-data-field-attlist"/>
9428    <ref name="list-source"/>
9429    <ref name="list-source-type"/>
9430 </define>
```

С элементом `<form:combobox>` могут быть связаны следующие атрибуты:

- имя и имя службы (для получения подробной информации об этих атрибутах см. раздел 11.4);
- идентификатор элемента управления, текущее значение, запрещен, выпадающий список, максимальная длина, возможность печати, только для чтения, размер, индекс табуляции, остановка табуляции, заголовок и значение (для получения подробной информации об этих атрибутах см. раздел 11.5);
- преобразование пустого значения, поле данных, список-источник, тип списка-источника (для получения подробной информации об этих атрибутах см. раздел 11.5.22);
- автоматическое дополнение.

Автоматическое дополнение

Атрибут `form:auto-complete` определяет, будет ли автоматически дополняться текст из списка во время набора пользователем при совпадении начальных символов.

```
9431 <define name="form-combobox-attlist" combine="interleave">
9432    <optional>
9433        <attribute name="form:auto-complete">
9434            <ref name="boolean"/>
9435        </attribute>
9436    </optional>
```

```
</define>
```

Пункт

Элемент `<form:item>` описывает пункт списка для элемента управления поле со списком.

```
<define name="form-item">
    <element name="form:item">
        <ref name="form-item-attlist"/>
        <text/>
    </element>
</define>
<define name="form-item-attlist" combine="interleave">
    <ref name="label"/>
</define>
```

С элементом `<form:item>` может быть связан следующий атрибут:

- метка (для получения подробной информации об этом атрибуте см. раздел 11.5).

11.3.10 Список

Элемент `<form:listbox>` описывает элемент управления ввода, который позволяет пользователю выбирать один или несколько элементов из списка. Это альтернативное представление группы радиокнопок.

```
<define name="column-controls" combine="choice">
    <element name="form:listbox">
        <ref name="form-listbox-attlist"/>
        <ref name="common-form-control-content"/>
        <zeroOrMore>
            <ref name="form-option"/>
        </zeroOrMore>
    </element>
</define>
<define name="form-listbox-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="dropdown"/>
    <ref name="common-printable-attlist"/>
    <ref name="size"/>
    <ref name="common-tab-attlist"/>
    <ref name="common-title-attlist"/>
    <ref name="bound-column"/>
    <ref name="common-data-field-attlist"/>
    <ref name="list-source"/>
    <ref name="list-source-type"/>
</define>
```

С элементом `<form:listbox>` могут быть связаны следующие атрибуты:

- имя и имя службы (для получения подробной информации об этих атрибутах см. раздел 11.4);

- идентификатор элемента управления, запрещен, выпадающий список, возможность печати, только для чтения, размер, индекс табуляции, остановка табуляции, заголовок (для получения подробной информации об этих атрибутах см. раздел 11.5);
- колонка обрамления, поле данных, список-источник, тип списка-источника (для получения подробной информации об этих атрибутах см. раздел 11.5.22);
- несколько пунктов;
- источник XForms.

Несколько пунктов

Атрибут `form:multiple` определяет, имеет ли пользователь возможность выбрать несколько пунктов в блоке списка.

```
<define name="form-listbox-attlist" combine="interleave">
    <optional>
        <attribute name="form:multiple" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Источник XForms

Атрибут `form:xforms-list-source` позволяет динамически создавать список для выбора путем связывания содержимого блока списка с XForms (см. раздел 11.2, [XForms]). Атрибут ссылается на элемент `<xforms:bind>` и создает объект списка для каждой записи в наборе данных, определенном этим атрибутом.

```
<define name="form-listbox-attlist" combine="interleave">
    <optional>
        <attribute name="form:xforms-list-source">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Выбор

Элемент `<form:option>` представляет пункты для списка. Один из пунктов может быть выбран заранее и содержать связанное значение.

```
<define name="form-option">
```

```
9484     <element name="form:option">
9485         <ref name="form-option-attlist"/>
9486         <text/>
9487     </element>
9488 </define>
9489 <define name="form-option-attlist" combine="interleave">
9490     <ref name="current-selected"/>
9491     <ref name="selected"/>
9492     <ref name="label"/>
9493     <ref name="common-value-attlist"/>
9494 </define>
```

С элементом `<form:option>` могут быть связаны следующие атрибуты:

- текущий выделенный пункт, выделение, метка, значение (для получения подробной информации об этих атрибутах см. раздел 11.5).

11.3.11 Кнопка

Элемент `<form:button>` определяет кнопку. Обычно нажатие кнопки вызывает некоторое действие.

```
9495 <define name="controls" combine="choice">
9496     <element name="form:button">
9497         <ref name="form-button-attlist"/>
9498         <ref name="common-form-control-content"/>
9499     </element>
9500 </define>
9501 <define name="form-button-attlist" combine="interleave">
9502     <ref name="form-control-attlist"/>
9503     <ref name="button-type"/>
9504     <ref name="common-disabled-attlist"/>
9505     <ref name="label"/>
9506     <ref name="image-data"/>
9507     <ref name="common-printable-attlist"/>
9508     <ref name="common-tab-attlist"/>
9509     <ref name="target-frame"/>
9510     <ref name="target-location"/>
9511     <ref name="common-title-attlist"/>
9512     <ref name="common-value-attlist"/>
9513     <ref name="common-form-relative-image-position-attlist"/>
9514 </define>
```

С элементом `<form:button>` могут быть связаны следующие атрибуты:

- имя и имя службы (для получения подробной информации об этих атрибутах см. раздел 11.4);
- тип кнопки, идентификатор элемента управления, запрещен, изображение, возможность печати, индекс табуляции, остановка табуляции, целевой фрейм, целевой адрес, заголовок, значение, относительное расположение изображения (для получения подробной информации об этих атрибутах см. раздел 11.5);
- кнопка по умолчанию;

- переключение;
- фокусировать по щелчку;
- отправка XForms.

Кнопка по умолчанию

Атрибут `form:default-button` определяет, является ли кнопка кнопкой формы по умолчанию. Если пользователь щелкает на кнопке по умолчанию или нажимает «Ввод» в то время, как элемент управления находится в фокусе, приложение выполняет одинаковые действия.

Если форма содержит более одной кнопки по умолчанию, то поведение приложения не определяется.

```
9515 <define name="form-button-attlist" combine="interleave">
9516     <optional>
9517         <attribute name="form:default-button" a:defaultValue="false">
9518             <ref name="boolean"/>
9519         </attribute>
9520     </optional>
9521 </define>
```

Переключение

Атрибут `form:toggle` определяет, переключается ли кнопка между состояниями «нажата» и «не нажата» при воздействии на нее (посредством мыши или клавиатуры). Если этот атрибут установлен в `false`, элемент управления кнопка ведет себя подобно обыкновенной кнопке.

```
9522 <define name="form-button-attlist" combine="interleave">
9523     <optional>
9524         <attribute name="form:toggle" a:default-value="false">
9525             <ref name="boolean"/>
9526         </attribute>
9527     </optional>
9528 </define>
```

Фокусировать по щелчку

Атрибут `form:focus-on-click` определяет, должен ли элемент управления кнопка при щелчке мышью на нем захватывать фокус ввода.

```
9529 <define name="form-button-attlist" combine="interleave">
9530     <optional>
9531         <attribute name="form:focus-on-click">
9532             <ref name="boolean"/>
9533         </attribute>
9534     </optional>
9535 </define>
```

Отправка XForms

Кнопки могут использоваться для отправки XForms посредством добавления атрибута `form:xforms-submission`. Если такая кнопка инициируется, выполняется определенная ранее отправка XForms с заданным именем.

```
9536 <define name="form-button-attlist" combine="interleave">
9537     <optional>
9538         <attribute name="form:xforms-submission">
9539             <ref name="string"/>
9540         </attribute>
9541     </optional>
9542 </define>
```

11.3.12 Изображение

Элемент `<form:image>` определяет графический элемент управления «кнопка». Этот элемент соответствует элементу ввода типа «изображение» в HTML 4.01. Обратите внимание: HTML 4.01 позволяет типу кнопки быть только типом отправки для кнопки-изображения. В формате файлов офисных приложений кнопка-изображение может быть любого типа.

```
9543 <define name="controls" combine="choice">
9544     <element name="form:image">
9545         <ref name="form-image-attlist"/>
9546         <ref name="common-form-control-content"/>
9547     </element>
9548 </define>
9549 <define name="form-image-attlist" combine="interleave">
9550     <ref name="form-control-attlist"/>
9551     <ref name="button-type"/>
9552     <ref name="common-disabled-attlist"/>
9553     <ref name="image-data"/>
9554     <ref name="common-printable-attlist"/>
9555     <ref name="common-tab-attlist"/>
9556     <ref name="target-frame"/>
9557     <ref name="target-location"/>
9558     <ref name="common-title-attlist"/>
9559     <ref name="common-value-attlist"/>
9560 </define>
```

С элементом `<form:image>` могут быть связаны следующие атрибуты:

- имя и имя службы (см. раздел 11.4 для получения информации об этих атрибутах);
- тип кнопки, идентификатор управляющего элемента, изображение, индекс табуляции, остановка табуляции, возможность печати, целевой фрейм, целевой адрес, заголовок, значение (для получения подробной информации об этих атрибутах см. раздел 11.5).

11.3.13 Флажок

Элемент `<form:checkbox>` определяет элемент управления, который может переключаться пользователем. Элемент управления находится во включенном состоянии, когда атрибут `form:current-state`, связанный с элементом управления, имеет значение `checked`. Когда пользователь отправляет форму, в качестве выбранных рассматриваются только элементы управления, имеющее состояние `checked`.

```
<define name="column-controls" combine="choice">
    <element name="form:checkbox">
        <ref name="form-checkbox-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="form-checkbox-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="label"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-tab-attlist"/>
    <ref name="common-title-attlist"/>
    <ref name="common-value-attlist"/>
    <ref name="common-data-field-attlist"/>
    <ref name="common-form-visual-effect-attlist"/>
    <ref name="common-form-relative-image-position-attlist"/>
</define>
```

С элементом `<form:checkbox>` могут быть связаны следующие атрибуты:

- имя и имя службы (см. раздел 11.4 для получения информации об этих атрибутах);
- идентификатор управляющего элемента, запрещен, метка, вывод на печать, индекс табуляции, остановка табуляции, заголовок, значение, визуальный эффект и относительное размещение изображения (для получения подробной информации об этих атрибутах см. раздел 11.5).
- поле данных (для получения информации об этом атрибуте см. раздел 11.5.22);.
- текущее состояние;
- с тремя состояниями;
- состояние.

Текущее состояние

Атрибут `form:current-state` определяет текущее состояние элемента управления флажок.

Атрибут может принимать одно из следующих значений:

- `unchecked` — флажок не установлен;
- `checked` — флажок установлен, значение элемента управления передается с формой;
- `unknown` — это значение доступно только в том случае, когда элемент управления находится в режиме с тремя состояниями (см. атрибут «С тремя состояниями»), это значение может использоваться, например, при соединении с полем базы данных для обозначения значения NULL.

```
<define name="states">
    <choice>
        <value>unchecked</value>
        <value>checked</value>
        <value>unknown</value>
    </choice>
</define>
<define name="form-checkbox-attlist" combine="interleave">
    <optional>
        <attribute name="form:current-state">
            <ref name="states"/>
        </attribute>
    </optional>
</define>
```

С тремя состояниями

Атрибут `form:is-tristate` определяет, что флажок может иметь три состояния вместо обычных двух.

```
<define name="form-checkbox-attlist" combine="interleave">
    <optional>
        <attribute name="form:is-tristate" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Состояние

Атрибут `form:state` определяет состояние элемента управления флажок по умолчанию. Это состоянии используется при инициализации объекта управления.

```
<define name="form-checkbox-attlist" combine="interleave">
```

```
9601     <optional>
9602         <attribute name="form:state" a:defaultValue="unchecked">
9603             <ref name="states"/>
9604         </attribute>
9605     </optional>
9606 </define>
```

11.3.14 Радиокнопка

Элемент `<form:radio>` определяет элемент управления, который функционирует подобно флажку за исключением того, что когда несколько радиокнопок разделяют одно имя элемента управления, они являются взаимно исключающими. Если одна радиокнопка включена, то все остальные радиокнопки с таким же именем выключены. Если в исходном состоянии нет включенных радиокнопок, то способ выбора приложением радиокнопки, включенной в исходном состоянии, не определяется.

Если группа радикнопок связана с одним полем базы данных, соответствующие значения радиокнопок записываются в поле базы данных.

```
9607 <define name="controls" combine="choice">
9608     <element name="form:radio">
9609         <ref name="form-radio-attlist"/>
9610         <ref name="common-form-control-content"/>
9611     </element>
9612 </define>
9613 <define name="form-radio-attlist" combine="interleave">
9614     <ref name="form-control-attlist"/>
9615     <ref name="current-selected"/>
9616     <ref name="common-disabled-attlist"/>
9617     <ref name="label"/>
9618     <ref name="common-printable-attlist"/>
9619     <ref name="selected"/>
9620     <ref name="common-tab-attlist"/>
9621     <ref name="common-title-attlist"/>
9622     <ref name="common-value-attlist"/>
9623     <ref name="common-data-field-attlist"/>
9624     <ref name="common-form-visual-effect-attlist"/>
9625     <ref name="common-form-relative-image-position-attlist"/>
9626 </define>
```

С элементом `<form:radio>` могут быть связаны следующие атрибуты:

- имя и имя службы (см. раздел 11.4 для получения информации об этих атрибутах);
- идентификатор управляющего элемента, текущий выбор, запрещен, метка, вывод на печать, выбор, индекс табуляции, остановка табуляции, заголовок, значение, визуальный эффект, относительное расположение изображения (для получения подробной информации об этих атрибутах см. раздел 11.5);

- поле данных (для получения информации об этом атрибуте см. раздел 11.5.22).

11.3.15 Фрейм

Элемент `<form:frame>` определяет фрейм, который может использоваться для визуального расположения элементов управления. Этот элемент не имеет значения и не обеспечивает ввода информации пользователем.

```
<define name="controls" combine="choice">
    <element name="form:frame">
        <ref name="form-frame-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="form-frame-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="for"/>
    <ref name="label"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-title-attlist"/>
</define>
```

С элементом `<form:frame>` могут быть связаны следующие атрибуты:

- имя и имя службы (см. раздел 11.4 для получения информации об этих атрибутах);
- идентификатор управляющего элемента, запрещен, для, метка, вывод на печать, заголовок (для получения подробной информации об этих атрибутах см. раздел 11.5).

11.3.16 Фрейм изображения

Элемент `<form:image-frame>` определяет графический элемент управления. Элемент управления отображает изображение, расположение которого указано в элементе управления.

```
<define name="controls" combine="choice">
    <element name="form:image-frame">
        <ref name="form-image-frame-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="form-image-frame-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="image-data"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-readonly-attlist"/>
    <ref name="common-title-attlist"/>
```

```
9654     <ref name="common-data-field-attlist"/>
9655 </define>
```

С элементом <form:image-frame> могут быть связаны следующие атрибуты:

- имя и имя службы (см. раздел 11.4 для получения информации об этих атрибутах);
- идентификатор управляющего элемента, запрещен, изображение, вывод на печать, только для чтения, заголовок (для получения подробной информации об этих атрибутах см. раздел 11.5);
- поле данных (для получения информации об этом атрибуте см. раздел 11.5.22).

11.3.17 Скрытый элемент

Элемент <form:hidden> определяет элемент управления, не имеющий визуального представления. Этот элемент обычно используется как контейнер для информации.

```
9656 <define name="controls" combine="choice">
9657     <element name="form:hidden">
9658         <ref name="form-hidden-attlist"/>
9659         <ref name="common-form-control-content"/>
9660     </element>
9661 </define>
9662 <define name="form-hidden-attlist" combine="interleave">
9663     <ref name="form-control-attlist"/>
9664     <ref name="common-value-attlist"/>
9665 </define>
```

С элементом <form:hidden> могут быть связаны следующие атрибуты:

- имя и имя службы (см. раздел 11.4 для получения информации об этих атрибутах);
- значение (для получения подробной информации об этом атрибуте см. раздел 11.5).

11.3.18 Сетка

Элемент <form:grid> определяет элемент управления, отображающий табличные данные. Этот элемент управления хранит информацию о данных и является обязательным для форм, получающих данные из источников данных. Текущие данные для отображения в элементе управления «сетка» определяются

родительской формой, которая хранит информацию о данных в виде набора некоторого количества строк.

Каждый столбец в сетке определяется элементом `<form:column>`. Каждый столбец связан с полем набора строк формы.

```
<define name="controls" combine="choice">
    <element name="form:grid">
        <ref name="form-grid-attlist"/>
        <ref name="common-form-control-content"/>
        <zeroOrMore>
            <ref name="form-column"/>
        </zeroOrMore>
    </element>
</define>
<define name="form-grid-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-tab-attlist"/>
    <ref name="common-title-attlist"/>
</define>
```

С элементом `<form:grid>` могут быть связаны следующие атрибуты:

- имя и имя службы (см. раздел 11.4 для получения информации об этих атрибутах);
- идентификатор управляющего элемента, запрещен, вывод на печать, индекс табуляции, остановка табуляции, заголовок (для получения подробной информации об этих атрибутах см. раздел 11.5).

Столбец

Элемент `<form:column>` определяет столбец элемента управления сетка. Столбец содержит элемент управления, отображающий данные сетки в столбце.

```
<define name="form-column">
    <element name="form:column">
        <ref name="form-column-attlist"/>
        <oneOrMore>
            <ref name="column-controls"/>
        </oneOrMore>
    </element>
</define>
<define name="form-column-attlist" combine="interleave">
    <ref name="common-form-control-attlist"/>
    <ref name="label"/>
    <ref name="text-style-name"/>
</define>
```

С элементом `<form:column>` могут быть связаны следующие атрибуты:

- имя и имя службы (см. раздел 11.4 для получения информации об этих атрибутах);

- легенда (для получения подробной информации об этом атрибуте см. раздел 11.5);
- стиль столбца.

Стиль столбца

Атрибут `form:text-style-name` определяет стиль абзаца, применяемый ко всем элементам управления в столбце (см. также раздел 9.2.12). В отличие от других стилей абзацев этот стиль может ссылаться на стиль данных.

```
<define name="text-style-name">
    <optional>
        <attribute name="form:text-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

11.3.19 Диапазон значений

Новый элемент `<form:value-range>` определяет элемент управления, позволяющий пользователю выбирать значение из последовательного диапазона чисел. Возможные представления включают полосы прокрутки и счетчики.

```
<define name="controls" combine="choice">
    <element name="form:value-range">
        <ref name="form-value-range-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="form-value-range-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
    <ref name="common-disabled-attlist"/>
    <ref name="common-printable-attlist"/>
    <ref name="common-tab-attlist"/>
    <ref name="common-title-attlist"/>
    <ref name="common-value-attlist"/>
</define>
```

С элементом `<form:value-range>` могут быть связаны следующие атрибуты:

- имя и имя службы (см. раздел 11.4 для получения информации об этих атрибутах);
- идентификатор управляющего элемента, текущее значение, запрещен, вывод на печать, только для чтения, индекс табуляции, остановка табуляции, заголовок, значение (для получения подробной информации об этих атрибутах см. раздел 11.5);

- максимальное значение;
- минимальное значение;
- размер шага;
- размер страничного шага;
- задержка повтора;
- ориентация.

Максимальное значение

Атрибут `form:max-value` определяет максимальное значение, которое может ввести пользователь.

```
<define name="form-value-range-attlist" combine="interleave">
    <optional>
        <attribute name="form:max-value">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Минимальное значение

Атрибут `form:min-value` определяет минимальное значение, которое может ввести пользователь.

```
<define name="form-value-range-attlist" combine="interleave">
    <optional>
        <attribute name="form:min-value">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Размер шага

Атрибут `form:step-size` определяет величину шага, которая будет использоваться элементом управления, представляющим значение.

```
<define name="form-value-range-attlist" combine="interleave">
    <optional>
        <attribute name="form:step-size" a:defaultName="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Размер шага страницы

Атрибут form:page-step-size определяет величину шага второго уровня, который будет использоваться элементом управления, представляющим значение. В пользовательском интерфейсе он обычно связывается с нажатием пользователем клавиш "Page Up" или "Page Down".

```
<define name="form-value-range-attlist" combine="interleave">
    <optional>
        <attribute name="form:page-step-size">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Задержка повтора

Атрибут form:delay-for-repeat определяет интервал времени, по истечению которого удерживание нажатой кнопки мыши приведет к повторению действия.

```
<define name="form-value-range-attlist" combine="interleave">
    <optional>
        <attribute name="form:delay-for-repeat">
            <ref name="duration"/>
        </attribute>
    </optional>
</define>
```

Ориентация

Атрибут form:orientation определяет ориентацию элемента управления, которая может быть горизонтальной или вертикальной.

```
<define name="form-value-range-attlist" combine="interleave">
    <optional>
        <attribute name="form:orientation">
            <choice>
                <value>horizontal</value>
                <value>vertical</value>
            </choice>
        </attribute>
    </optional>
</define>
```

11.3.20 Базовый элемент управления

Элемент <form:generic-control> определяет наполнение базового элемента управления. Базовый элемент управления может содержать любые

свойства и любые события. Приложение определяет тип элемента управления и устанавливает соответствующий элемент управления.

```
<define name="controls" combine="choice">
    <element name="form:generic-control">
        <ref name="form-generic-control-attlist"/>
        <ref name="common-form-control-content"/>
    </element>
</define>
<define name="form-generic-control-attlist" combine="interleave">
    <ref name="form-control-attlist"/>
</define>
```

С элементом `<form:generic-control>` могут быть связаны следующие атрибуты:

- имя и имя службы (см. раздел 11.4 для получения информации об этих атрибутах).

11.4 Общие атрибуты элементов управления и форм

11.4.1 Имя

Атрибут `form:name` определяет имя формы или элемента управления. Он может использоваться для получения идентификатора формы или элемента управления, что важно для обработки скриптами и для отправки содержимого элементам управления.

```
<define name="common-form-control-attlist" combine="interleave">
    <optional>
        <attribute name="form:name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

11.4.2 Реализация элемента управления

Элемент управления может быть задан атрибутом типа элемент управления, определяющим, какую конкретную обработку или реализацию должен поддерживать агент пользователя. Для простоты расширяемости значение этого атрибута является токен с пространством имен, т. е. это токен, использующий префикс пространства имен, более похожий на атрибуты в XML.

```
<define name="common-form-control-attlist" combine="interleave">
    <optional>
        <attribute name="form:control-implementation">
            <ref name="namespacedToken"/>
        </attribute>
```

```
    </optional>
</define>
```

11.4.3 Привязка к XForms

Любой элемент управления может быть связан с XForms формой (см. раздел 11.2, а также [XForms]) атрибутом `xforms:bind`. Для кнопок атрибут привязки ссылается на элемент `<xforms:submission>` с заданным идентификатором. Нажатие кнопки приводит к выполнению соответствующего действия отправки XForms. Для всех других типов элементов управления атрибут `xforms:bind` ссылается на элемент `<xforms:bind>` с заданным идентификатором. Такой связанный элемент управления читает и записывает данные, как определено соответствующим элементом привязки.

```
<define name="xforms-bind-attlist">
    <optional>
        <attribute name="xforms:bind">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

11.5 Общие атрибуты элементов управления

11.5.1 Тип кнопки

Атрибут `form:button-type` определяет тип кнопки. Этот атрибут поддерживается следующими элементами:

- `<form:button>`
- `<form:image>`

Атрибут может принимать следующие значения:

- `submit` — нажатие кнопки отправляет форму;
- `reset` — нажатие кнопки сбрасывает все элементы управления формы в их значение по умолчанию;
- `push` — нажатие кнопки не приводит к выполнению каких-либо действий по умолчанию, используется для возможности назначения кнопке скрипта, при нажатии кнопки запускается выполнение скрипта;
- `url` — нажатие кнопки приводит к загрузке URL, определенного в атрибуте `form:target-url`.

```
<define name="types">
    <choice>
        <value>submit</value>
        <value>reset</value>
        <value>push</value>
        <value>url</value>
    </choice>
</define>
<define name="button-type">
    <optional>
        <attribute name="form:button-type" a:defaultValue="push">
            <ref name="types"/>
        </attribute>
    </optional>
</define>
```

11.5.2 Идентификатор элемента управления

Все элементы управления, за исключением скрытых, имеют визуальное представление в родительском документе. Поэтому они нуждаются в определении абсолютного или относительного положения, описывающего их размещение в документе. Положение представляется фигурой, которая содержит ссылку на элемент управления в элементе формы.

Атрибут `form:id` используется для однозначной идентификации элемента управления. Каждый нескрытый элемент управления должен иметь такой связанный с ним атрибут, который может быть использован для ссылки на элемент управления.

Этот атрибут поддерживается следующими элементами:

- `<form:text>`
- `<form:textarea>`
- `<form:password>`
- `<form:file>`
- `<form:formatted-text>`
- `<form:fixed-text>`
- `<form:combobox>`
- `<form:listbox>`
- `<form:button>`
- `<form:image>`
- `<form:checkbox>`
- `<form:radio>`
- `<form:frame>`

- `<form:image-frame>`
- `<form:grid>`

```
<define name="common-control-id-attlist">
    <attribute name="form:id">
        <ref name="ID"/>
    </attribute>
</define>
```

11.5.3 Текущий выбор

Атрибут `form:current-selected` определяет текущее состояние переключателя или блок списка.

Этот атрибут поддерживается следующими элементами:

- `<form:option>`
- `<form:radio>`

```
<define name="current-selected">
    <optional>
        <attribute name="form:current-selected" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.5.4 Значение и текущее значение

Каждый элемент управления имеет значение по умолчанию и текущее значение. Текущее значение меняется действиями пользователя, значение по умолчанию — нет. Значение по умолчанию определяется атрибутом `form:value`.

Значение по умолчанию используется при возникновении специальных событий, таких как сброс формы, которые меняют текущее значение элемента управления на его значение по умолчанию. Если элемент управления не имеет значения по умолчанию, результат сброса формы не определен.

Кроме хранения текущего значения вместе с элементом управления, также возможно связывать элементы управления с другими поставщиками значений, которые работают как приемники и источники данных, как например, поля базы данных (в формах, хранящих информацию о данных) или, например, ячейки в документе электронных таблиц. В этом случае текущее значение хранится не в самом элементе управления, а во внешней связи, которая может хранить или не хранить его вместе с документом. Для получения большей информации о свойствах баз данных см. раздел 11.5.22.

Значение по умолчанию

Атрибут form:value определяет значение элемента управления ввода по умолчанию.

Этот атрибут поддерживается следующими элементами:

- <form:text>
- <form:textarea>
- <form:password>
- <form:file>
- <form:formatted-text>
- <form:combobox>
- <form:option>
- <form:button>
- <form:image>
- <form:checkbox>
- <form:radio>
- <form:hidden>

```
<define name="common-value-attlist">
    <optional>
        <attribute name="form:value">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Текущее значение

Атрибут form:current-value определяет текущее состояние элемента управления ввода. Он перекрывает значение атрибута form:value, если он представлен.

Этот атрибут поддерживается следующими элементами:

- <form:text>
- <form:textarea>
- <form:file>
- <form:formatted-text>
- <form:combobox>

```
<define name="common-current-value-attlist">
```

```
    <optional>
        <attribute name="form:current-value">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

11.5.5 Запрещен

Атрибут form:disabled определяет, доступен ли элемент управления пользователю для ввода. Этот атрибут поддерживается следующими элементами:

- <form:text>
- <form:textarea>
- <form:password>
- <form:file>
- <form:formatted-text>
- <form:fixed-text>
- <form:combobox>
- <form:listbox>
- <form:button>
- <form:image>
- <form:checkbox>
- <form:radio>
- <form:frame>
- <form:image-frame>
- <form:grid>

Запрещенные элементы управления не включаются в последовательность навигации табуляцией и не могут получать фокус ввода.

```
<define name="common-disabled-attlist">
    <optional>
        <attribute name="form:disabled" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.5.6 Выпадающий список

Атрибут `form:dropdown` определяет, будет ли список в поле со списком или в списке видимым всегда или видимым, только когда пользователь нажимает на кнопку раскрытия списка.

Этот атрибут поддерживается следующими элементами:

- `<form:combobox>`
- `<form:listbox>`

Если его значение `true`, список всегда видим. Если значение `false`, список виден только когда пользователь нажимает кнопку раскрытия списка.

```
<define name="dropdown">
    <optional>
        <attribute name="form:dropdown" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.5.7 Для

Атрибут `form:for` определяет идентификаторы элементов управления, с которыми связан данный элемент управления.

Этот атрибут поддерживается следующими элементами:

- `<form:fixed-text>`
- `<form:frame>`

Атрибут использует запятую в качестве разделителя в списке идентификаторов элементов управления.

```
<define name="for">
    <optional>
        <attribute name="form:for">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

11.5.8 Изображение

Атрибут `form:image-data` связывает элемент управления с внешним файлом, содержащим изображение.

Этот атрибут поддерживается следующими элементами:

- `<form:button>`

- <form:image>
- <form:image-frame>

```
<define name="image-data">
    <optional>
        <attribute name="form:image-data">
            <ref name="anyURI"/>
        </attribute>
    </optional>
</define>
```

11.5.9 Метка

Атрибут form:label содержит метку для таких элементов управления как кнопка переключателя или флажка.

Этот атрибут поддерживается следующими элементами:

- <form:fixed-text>
- <form:item>
- <form:option>
- <form:checkbox>
- <form:radio>
- <form:frame>
- <form:column>

```
<define name="label">
    <optional>
        <attribute name="form:label">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

11.5.10 Максимальная длина

Атрибут form:max-length определяет максимальное число символов, которые пользователь может ввести в элементы управления, предназначенные для ввода.

Этот атрибут поддерживается следующими элементами:

- <form:text>
- <form:textarea>
- <form:password>
- <form:formatted-text>
- <form:combobox>

Значение по умолчанию данного атрибута неограниченно, что позволяет пользователю вводить неограниченное число символов.

```
<define name="common-maxlength-attlist">
    <optional>
        <attribute name="form:max-length">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

11.5.11 Возможность печати

Атрибут `form:printable` определяет, возможно ли распечатать элемент управления, когда пользователь печатает документ, содержащий элемент управления.

Этот атрибут поддерживается следующими элементами:

- `<form:text>`
- `<form:textarea>`
- `<form:password>`
- `<form:file>`
- `<form:formatted-text>`
- `<form:fixed-text>`
- `<form:combobox>`
- `<form:listbox>`
- `<form:button>`
- `<form:image>`
- `<form:checkbox>`
- `<form:radio>`
- `<form:frame>`
- `<form:image-frame>`
- `<form:grid>`

```
<define name="common-printable-attlist">
    <optional>
        <attribute name="form:printable" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.5.12 Только для чтения

Атрибут form:readonly определяет, может ли пользователь изменять значение элемента управления.

Этот атрибут поддерживается следующими элементами:

- <form:text>
- <form:textarea>
- <form:file>
- <form:formatted-text>
- <form:combobox>
- <form:listbox>
- <form:image-frame>

Элементы управления только для чтения включены в последовательность навигации по клавише TAB.

```
<define name="common-readonly-attlist">
    <optional>
        <attribute name="form:readonly" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.5.13 Выбранный

Атрибут form:selected определяет состояние по умолчанию для кнопок переключения или опций. Когда элемент управления инициализирован, он находится в состоянии по умолчанию, определенном его атрибутом.

Этот атрибут поддерживается следующими элементами:

- <form:option>
- <form:radio>

В группе переключателей с одинаковыми именами только один переключатель может иметь этот атрибут, установленный в значение true.

```
<define name="selected">
    <optional>
        <attribute name="form:selected" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.5.14 Размер

Атрибут form:size определяет число строк, которые одновременно видны в поле со списком или в списке.

Этот атрибут поддерживается следующими элементами:

- <form:combobox>
- <form:listbox>

```
<define name="size">
    <optional>
        <attribute name="form:size">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

11.5.15 Индекс табуляции

Атрибут form:tab-index определяет порядок навигации по клавише TAB между элементами управления формы. Порядок табуляции является порядком, в котором элементы управления получают фокус ввода, когда пользователь нажимает клавишу TAB на клавиатуре. Порядок табуляции может включать элементы, которые содержатся внутри других элементов.

Этот атрибут поддерживается следующими элементами:

- <form:text>
- <form:textarea>
- <form:password>
- <form:file>
- <form:formatted-text>
- <form:combobox>
- <form:listbox>
- <form:button>
- <form:image>
- <form:checkbox>
- <form:radio>
- <form:grid>

Правила табуляции похожи на правила, используемые в HTML 4.0.

Навигация по элементам управления, которые получают фокус ввода, осуществляется в порядке описанном следующими правилами:

- элементы управления, которые имеют положительное значение атрибута form:tab-index, получают фокус ввода в первую очередь;
- навигация начинается с элементов управления, имеющих наименьшее значение атрибута form:tab-index, и заканчивается на элементе управления с наибольшим значением, значения не обязательно должны быть последовательными или начинаться с особого значения;
- элементы управления, которые имеют одинаковые значения атрибута form:tab-index, получают фокус ввода в соответствии с их позицией на форме;
- затем фокус ввода получают элементы управления, которые не содержат атрибут form:tab-index или содержат атрибут со значением 0, эти элементы управления получают фокус ввода в соответствии с их позицией в форме;
- элементы управления, которые имеют атрибут form:disabled, установленный в значение true, не включаются в навигацию независимо от значения их атрибута form:tab-index.

```
<define name="common-tab-attlist" combine="interleave">
    <optional>
        <attribute name="form:tab-index" a:defaultValue="0">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

11.5.16 Остановка табуляции

Атрибут form:tab-stop определяет, будет ли элемент управления включен в порядок навигации.

Этот атрибут поддерживается следующими элементами:

- <form:text>
- <form:textarea>
- <form:password>
- <form:file>
- <form:formatted-text>
- <form:combobox>

- <form:listbox>
- <form:button>
- <form:image>
- <form:checkbox>
- <form:radio>
- <form:grid>

Если атрибут имеет значение false, элемент управления не включается в навигацию.

```
<define name="common-tab-attlist" combine="interleave">
    <optional>
        <attribute name="form:tab-stop" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

11.5.17 Целевой фрейм

Атрибут office:target-frame определяет связанный целевой фрейм области.

Этот атрибут поддерживается следующими элементами:

- <form:button>
- <form:image>

```
<define name="target-frame">
    <optional>
        <attribute name="office:target-frame" a:defaultValue="_blank">
            <ref name="targetFrameName"/>
        </attribute>
    </optional>
</define>
```

11.5.18 Целевой адрес

Атрибут xlink:href определяет URL, который загружается при нажатии кнопки.

Этот атрибут поддерживается следующими элементами:

- <form:button>
- <form:image>

Значение этого атрибута вычисляется, только если значение атрибута form:button-type является location.

```
<define name="target-location">
    <optional>
        <attribute name="xlink:href">
            <ref name="anyURI"/>
        </attribute>
    </optional>
</define>
```

11.5.19 Заголовок

Атрибут `form:title` содержит дополнительную информацию об элементе управления. Значение атрибута может быть использовано в качестве всплывающей подсказки.

Этот атрибут поддерживается следующими элементами:

- `<form:text>`
- `<form:textarea>`
- `<form:password>`
- `<form:file>`
- `<form:formatted-text>`
- `<form:fixed-text>`
- `<form:combobox>`
- `<form:listbox>`
- `<form:button>`
- `<form:image>`
- `<form:checkbox>`
- `<form:radio>`
- `<form:image>`
- `<form:image-frame>`
- `<form:grid>`

```
<define name="common-title-attlist">
    <optional>
        <attribute name="form:title"/>
    </optional>
</define>
```

11.5.20 Визуальный эффект

Атрибут form:visual-effect определяет визуальный эффект, примененный для элемента управления. Значения атрибута могут быть flat для плоского визуального эффекта и 3d для трехмерного эффекта.

Этот атрибут поддерживается следующими элементами:

- <form:checkbox>
- <form:radio>

```
<define name="common-form-visual-effect-attlist" combine="interleave">
    <optional>
        <attribute name="form:visual-effect">
            <choice>
                <value>flat</value>
                <value>3d</value>
            </choice>
        </attribute>
    </optional>
</define>
```

11.5.21 Относительное расположение изображения

Атрибуты form:image-position и form:image-align вместе определяют позицию изображения, отображаемого на элементе управления формы, относительно текстовой метки.

Когда атрибут form:image-position имеет значение center, изображение, отображаемое на элементе управления, должно быть отцентрировано относительно текста элемента управления.

Когда атрибут form:image-position имеет одно из следующих значений: start, end, top, bottom, изображение располагается до, после, сверху или снизу от текста. В этом случае атрибут form:image-align определяет, какая граница (start, end) или ось (center) изображения и текста совмещаются. Если атрибут form:image-position не представлен, то подразумевается значение center.

Атрибуты form:image-position и form:image-align поддерживаются следующими элементами:

- <form:button>
- <form:checkbox>
- <form:radio>

```
<define name="common-form-relative-image-position-attlist"
        combine="interleave">
```

```
9947        <choice>
9948            <optional>
9949                <attribute name="form:image-position" a:defaultValue="center">
9950                    <value>center</value>
9951                </attribute>
9952            </optional>
9953            <group>
9954                <attribute name="form:image-position">
9955                    <choice>
9956                        <value>start</value>
9957                        <value>end</value>
9958                        <value>top</value>
9959                        <value>bottom</value>
9960                    </choice>
9961                </attribute>
9962                <optional>
9963                    <attribute name="form:image-align" a:defaultValue="center">
9964                        <choice>
9965                            <value>start</value>
9966                            <value>center</value>
9967                            <value>end</value>
9968                        </choice>
9969                    </attribute>
9970                </optional>
9971            </group>
9972        </choice>
9973    </define>
```

11.5.22 Атрибуты связи с базой данных

Элемент управления может быть привязан к полям базы данных. В этом случае элементы управления становятся связанными с данными. Элемент управления получает значения из поля базы данных путем просмотра результирующего набора, который предоставляется формой. Каждый раз, когда изменяется строка в форме, значение элемента управления может измениться. Изменения значения сохраняются в связанном поле базы данных.

Связанный столбец

Атрибут `form:bound-column` определяет значение столбца списка источника, который используется для заполнения значений полей базы данных.

Этот атрибут поддерживается элементом `<form:listbox>`.

```
9974    <define name="bound-column">
9975        <optional>
9976            <attribute name="form:bound-column">
9977                <ref name="string"/>
9978            </attribute>
9979        </optional>
9980    </define>
```

Преобразование пустого значения в Null

Атрибут form:convert-empty-to-null определяет, воспринимать ли пустое текущее значение как NULL. Этот атрибут важен для элементов управления связанных с данными для определения того, какие значения сохранять для связанного поля базы данных.

Атрибут поддерживается следующими элементами:

- <form:text>
- <form:textarea>
- <form:formatted-text>
- <form:combobox>

Если значение атрибута равно true, пустая строка в элементе управления воспринимается как NULL-значение. Если значение атрибута false, пустая строка в элементе управления воспринимается как пустая строка.

```
<define name="common-convert-empty-attlist">
    <optional>
        <attribute name="form:convert-empty-to-null" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Поле данных

Атрибут form:data-field определяет название результирующего набора. Результирующий набор определяется формой, к которой принадлежит элемент управления.

Атрибут поддерживается следующими элементами:

- <form:text>
- <form:textarea>
- <form:formatted-text>
- <form:combobox>
- <form:listbox>
- <form:checkbox>
- <form:radio>
- <form:image-frame>

```
<define name="common-data-field-attlist">
    <optional>
```

```
9990        <attribute name="form:data-field">
9991            <ref name="string"/>
9992        </attribute>
9993    </optional>
9994 </define>
```

Источник списка

Атрибут `form:list-source` определяет источник, используемый для наполнения списка в окне списка или в поле со списком. Первый столбец результирующего набора источника наполняет список.

Атрибут поддерживается следующими элементами:

- `<form:combobox>`
- `<form:listbox>`

```
9995 <define name="list-source">
9996     <optional>
9997         <attribute name="form:list-source">
9998             <ref name="string"/>
9999         </attribute>
10000    </optional>
10001 </define>
```

Типы источника данных для списков

Атрибут `form:list-source-type` определяет тип источника данных, которые используются для заполнения списка данных в списках и полях со списком.

Атрибут поддерживается следующими элементами:

- `<form:combobox>`
- `<form:listbox>`

Данный атрибут может принимать одно из следующих значений:

- `table` — список заполнен с использованием содержимого таблицы базы данных;
- `query` — список заполнен результатом выполненного запроса;
- `sql` — Список заполнен выполнением выражения SQL;
- `sql-pass-through` — список заполнен результатом выполнения оператора любого типа, который прошел непосредственно к драйверу базы данных, без интерпретации приложением;

- value-list — список заполнен значениями определенными пользователем с использованием атрибута form:value в элементе <form:option>, эти установки применимы только к спискам;
- table-fields — список заполнен с использованием имен полей в таблице базы данных.

```
10002 <define name="list-source-type">
10003     <optional>
10004         <attribute name="form:list-source-type">
10005             <choice>
10006                 <value>table</value>
10007                 <value>query</value>
10008                 <value>sql</value>
10009                 <value>sql-pass-through</value>
10010                 <value>value-list</value>
10011                 <value>table-fields</value>
10012             </choice>
10013         </attribute>
10014     </optional>
10015 </define>
```

11.6 События

HTML определяет список стандартных событий для элементов управления. Эти события представлены атрибутами, связанными с элементами управления. В офисных приложениях формата XML эти события и любые дополнительные события, определенные компонентами приложения, хранятся как элементы в элементе <office:event-listeners>.

Для события отдельного элемента атрибут script:event-name устанавливает тип события и другие атрибуты, определяющие язык и обработчик события.

11.6.1 События, имеющие эквивалентные типы событий в HTML

Следующая таблица описывает события XML, имеющие эквивалентные события в HTML. Их имена содержатся в пространстве имен "http://www.w3.org/2001/xml-events". В этой спецификации используется префикс пространства имен DOM (см. также раздел 12.4.1).

Значение атрибута script:event-name	*Эквивалентное событие HTML*	*Описание события*
dom:change	onchange	Происходит, когда элемент управления больше не активен, и его значение было изменено после

Значение атрибута script:event-name	*Эквивалентное событие HTML*	*Описание события*
		активизации
dom:DOMFocusIn	onfocus	Происходит, когда элемент управления активизируется с использованием мыши или клавиши TAB
dom:DOMFocusOut	onblur	Происходит, когда элемент управления больше не активизируется в результате движения мыши или табуляции. Может использоваться с такими элементами как form:on-focus
dom:keydown	onkeydown	Происходит при нажатии клавиши на элементе управления
dom:keyup	onkeyup	Происходит, когда отпускается клавиша на элементе управления
dom:mouseover	onmouseover	Происходит, когда на элемент управления направляется курсор мыши
dom:mousemove	onmousemove	Происходит, когда на элементе управления двигается курсор мыши
dom:mousedown	onmousedown	Происходит, когда на элементе управления нажимается кнопка мыши
dom:mouseup	onmouseup	Происходит, когда на элементе управления отпускается кнопка мыши
on-mouseout	onmouseout	Происходит, когда курсор мыши сдвигается с элемента управления
dom:reset	onreset	Происходит при восстановлении формы
dom:submit	onsubmit	Происходит при отправке формы

11.6.2 Типы событий

В дополнение к типам событий HTML, формат XML для офисных приложений допускает дополнительные события, обрабатываемые во время выполнения.

Значение атрибута script:event-name	*Применяется к объекту*	*Описание события*
form:approveaction	Кнопка или изображение	Происходит перед событием on-performaction. Позволяет пользователю запретить действие
form:performaction	Кнопка или изображение	Происходит, когда должно быть выполнено управляющее действие. Частая интерпретация этого события — «нажатие на кнопку»
form:textchange	Все элементы управления,	Происходит, когда пользователь изменяет текст в элементе управления

Значение атрибута script:event-name	***Применяется к объекту***	***Описание события***
	допускающие ввод текста	
form:itemstatechange	Флажок или переключатель	Происходит при изменении состояния флажка или переключателя
form:mousedrag	Все элементы управления	Происходит, когда пользователь нажимает и держит одну из кнопок мыши, и двигает курсор мыши на элемент управления
form:approvereset	Те же, что и для события form:on-reset	Происходит перед событием on-reset. Позволяет пользователю наложить на него запрет
form:approveupdate	Все элементы управления, которые могут быть связаны с полем базы данных, то есть содержащие атрибуты поля данных	Происходит перед событием on-update. Позволяет пользователю наложить запрет на обновление
form:update	Все элементы управления, которые могут быть связаны с полем базы данных, то есть содержащие атрибуты поля данных	Происходит, когда фиксируется содержимое элемента управления, связанного с полем базы данных
form:load	Формы	Происходит, когда форма устанавливает соединение с источником данных
form:startrealod	Формы	Происходит, когда форма начинает обновлять соединение с источником данных
form:reload	Формы	Происходит, когда форма обновляет соединение с источником данных
form:startunload	Формы	Происходит, когда форма собирается завершить соединение с источником данных
form:unload	Формы	Происходит, когда форма завершает соединение с источником данных
form:confirmdelete	Формы	Происходит, когда пользователь собирается

Значение атрибута script:event-name	***Применяется к объекту***	***Описание события***
		удалить запись
form:approverowchange	Формы	Происходит перед событием on-rowchange. Позволяет пользователю наложить запрет на изменение
form:rowchange	Формы	Происходит после завершения обновлений строк, таких как удаления, изменения и вставки
form:approvecursormove	Формы	Происходит перед сдвигом на следующую строку. Позволяет пользователю наложить запрет на сдвиг
form:cursormove	Формы	Происходит после сдвига на другую строку
form:supplyparameter	Формы	Происходит, когда форме необходимо заполнить параметры для установления связи с источником данных
form:error	Формы, объекты поле со списком и список	Происходит при ошибке, связанной с базой данных
form:adjust	Диапазон значений	Происходит, когда значение элемента диапазона значений было откорректировано

11.7 Свойства

Элемент <form:properties> может быть использован для хранения следующих установок элементов управления и форм:

- параметры, не определенные в формате документа;
- параметры, обеспечиваемые внешними соединениями;
- параметры, специфические для приложения.

Свойства состоят из пары имя/значение. Имя идентифицирует свойство. Значение может быть базовым типом данных или списком базовых типов данных.

11.7.1 Набор свойств

Элемент <form:properties> содержит элементы свойств. Свойства могут быть закодированы с помощью элемента form:property, кроме свойств списка, использующих элемент form:list-property.

```
<define name="form-properties">
    <element name="form:properties">
        <oneOrMore>
            <ref name="form-property"/>
```

```
        </oneOrMore>
    </element>
</define>
```

11.7.2 Свойство

Элемент `<form:property>` описывает самостоятельное свойство и содержит его имя, тип и значение.

```
<define name="form-property" combine="choice">
    <element name="form:property">
        <ref name="form-property-name"/>
        <ref name="form-property-value-and-type-attlist"/>
    </element>
</define>
```

Имя свойства

Атрибут `form:property-name` устанавливает имя элемента свойства.

```
<define name="form-property-name" combine="interleave">
    <attribute name="form:property-name">
        <ref name="string"/>
    </attribute>
</define>
```

Значение и тип свойства

Значение и тип свойств формы представляются через общий атрибут `office:value-type` и подходящие атрибуты значений. Более подробную информацию об этих атрибутах см. в разделе 6.7.1.

Свойства формы могут принимать значение одного из этих типов или быть пустыми. В последнем случае используется специальное значение типа `void`. Такие свойства не имеют значения атрибута.

```
<define name="form-property-value-and-type-attlist" combine="interleave">
    <choice>
        <ref name="common-value-and-type-attlist"/>
        <attribute name="office:value-type">
            <value>void</value>
        </attribute>
    </choice>
</define>
```

11.7.3 Список свойств

Элемент `<form:list-property>` устанавливает свойство, содержащее список значений. Атрибут типа значения указывает, какие типы допустимы в элементах списка. Элемент содержит последовательность элементов значений

списка, каждый из которых содержит атрибут значения, совместимого с типом из элемента `<form:list-property>`. Атрибуты значений сходны с используемыми в спецификации, кроме тех, у которых атрибут типа связан с составным элементом, атрибуты значений которого связаны со значениями из списка. Более подробную информацию об атрибутах значений см. в разделе 6.7.1.

```
<define name="form-property" combine="choice">
    <element name="form:list-property">
        <ref name="form-property-name"/>
        <ref name="form-property-type-and-value-list"/>
    </element>
</define>
```

Значение списка

Элемент значение списка содержит атрибуты значений для типа значений, заданного в элементе `<form:list-property>`.

```
<define name="form-property-type-and-value-list">
    <choice>
        <group>
            <attribute name="office:value-type">
                <value>float</value>
            </attribute>
            <zeroOrMore>
                <element name="form:list-value">
                    <attribute name="office:value">
                        <ref name="double"/>
                    </attribute>
                </element>
            </zeroOrMore>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>percentage</value>
            </attribute>
            <zeroOrMore>
                <element name="form:list-value">
                    <attribute name="office:value">
                        <ref name="double"/>
                    </attribute>
                </element>
            </zeroOrMore>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>currency</value>
            </attribute>
            <zeroOrMore>
                <element name="form:list-value">
                    <attribute name="office:value">
                        <ref name="double"/>
                    </attribute>
                    <optional>
                        <attribute name="office:currency">
                            <ref name="string"/>
                        </attribute>
```

```
                    </optional>
                </element>
            </zeroOrMore>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>date</value>
            </attribute>
            <zeroOrMore>
                <element name="form:list-value">
                    <attribute name="office:date-value">
                        <ref name="dateOrDateTime"/>
                    </attribute>
                </element>
            </zeroOrMore>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>time</value>
            </attribute>
            <zeroOrMore>
                <element name="form:list-value">
                    <attribute name="office:time-value">
                        <ref name="duration"/>
                    </attribute>
                </element>
            </zeroOrMore>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>boolean</value>
            </attribute>
            <zeroOrMore>
                <element name="form:list-value">
                    <attribute name="office:boolean-value">
                        <ref name="boolean"/>
                    </attribute>
                </element>
            </zeroOrMore>
        </group>
        <group>
            <attribute name="office:value-type">
                <value>string</value>
            </attribute>
            <zeroOrMore>
                <element name="form:list-value">
                    <attribute name="office:string-value">
                        <ref name="string"/>
                    </attribute>
                </element>
            </zeroOrMore>
        </group>
        <attribute name="office:value-type">
            <value>void</value>
        </attribute>
    </choice>
</define>
```

Пример – Свойства формы:

В следующем примере содержится строковое свойство «Name» со значением «Name 1» и список строковых значений «Items», содержащий строки «Item 1», «Item 2», «Item 3».

```
<form:properties>
    <form:property form:property-name="Name"
                   office:value-type="string"
                   office:string-value="Name 1">
    <form:list-property form:property-name="Items"
                   office:value-type="string" >
        <form:list-value office:string-value="Item 1"/>
        <form:list-value office:string-value="Item 2"/>
        <form:list-value office:string-value="Item 3"/>
    </form:list-property>
</form:properties>
```

12 Общее содержимое

12.1 Аннотация

Элемент `<office:annotation>` задает аннотацию в формате OpenDocument. Текст аннотации содержится в элементах `<text:p>` и `<text:list>`.

```
<define name="office-annotation">
    <element name="office:annotation">
        <ref name="office-annotation-attlist"/>
        <ref name="draw-caption-attlist"/>
        <ref name="common-draw-position-attlist"/>
        <ref name="common-draw-size-attlist"/>
        <ref name="common-draw-shape-with-text-and-styles-attlist"/>
        <optional>
            <ref name="dc-creator"/>
        </optional>
        <optional>
            <ref name="dc-date"/>
        </optional>
        <optional>
            <ref name="meta-date-string"/>
        </optional>
        <zeroOrMore>
            <choice>
                <ref name="text-p"/>
                <ref name="text-list"/>
            </choice>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<office:annotation>` могут быть связаны следующие атрибуты

- отображение;
- положение, размер, стиль, слой, z-индекс, идентификатор, преобразование (см. раздел 9.2.15);
- текстовая привязка, фон таблицы, конечное положение рисунка (см. раздел 9.2.16);
- точка подписи, скругление углов (см. раздел 9.2.10).

Отображение

Атрибут `office:display` определяет, отображается аннотация или нет.

```
10168 <define name="office-annotation-attlist" combine="interleave">
10169     <optional>
10170         <attribute name="office:display">
10171             <ref name="boolean"/>
10172         </attribute>
10173     </optional>
10174 </define>
```

Атрибуты заголовка

Следующие атрибуты могут быть назначены элементу `<office:annotation>` для того, чтобы повлиять на отображение: `svg:x`, `svg:y`, `svg:width`, `svg:height`, `draw:caption-point-x`, `draw:caption-point-y`, `draw:corner-radius`, `table:end-cell-address`, `table:end-x`, `table:end-y`, `text:anchor-type`, `text:anchor-page-number`, `draw:layer`, `draw:style-name`, `draw:text-style-name`, `draw:transform`, `draw:name`, `draw:z-index` и `draw:id`. Их смысл тот же, как если бы они применялись к элементу `<draw:caption>` (см. раздел 9.2.10). Использование данных атрибутов не обязательно.

12.1.1 Автор

Необязательный элемент `<dc:creator>`, описанный в разделе 3.1.7, определяет автора примечания.

12.1.2 Дата и время создания

Необязательный элемент `<dc:date>`, описанный в разделе 3.1.9, задает дату и время создания примечания.

12.1.3 Строка даты и времени создания

Если приложение имеет только строку с датой и не может разобрать эту строку, то оно может записать ее в элемент `<meta:date-string>`.

```
10175 <define name="meta-date-string">
10176     <element name="meta:date-string">
10177         <ref name="string"/>
10178     </element>
10179 </define>
```

12.2 Формат чисел

Формат чисел для OpenDocument состоит из трех частей:

- префикс – текст отображаемый до числа;
- спецификация отображаемого формата, например, A, B, C, или 1, 2, 3;
- суффикс – текст, отображаемый после числа.

12.2.1 Префикс и суффикс

Атрибуты `style:num-prefix` и `style:num-suffix` определяют, что отображать до и после номера.

Если префикс и суффикс не содержат алфавитно-цифровых символов, то атрибут `format` [XSLT] может быть создан из атрибутов OpenDocument через объединение значений атрибутов `style:num-prefix`, `style:num-format` и `style:num-suffix`.

```
10180 <define name="common-num-format-prefix-suffix-attlist" combine="interleave">
10181     <optional>
10182         <attribute name="style:num-prefix">
10183             <ref name="string"/>
10184         </attribute>
10185     </optional>
10186     <optional>
10187         <attribute name="style:num-suffix">
10188             <ref name="string"/>
10189         </attribute>
10190     </optional>
10191 </define>
```

12.2.2 Спецификация формата

Атрибут `style:num-format` определяет формат чисел так же, как атрибут `format` [XSLT]. Поддерживаются следующие форматы номера:

- числовой: 1, 2, 3, ... ;
- алфавитный: a, b, c, ... или A, B, C, ... ;
- римский: i, ii, iii, iv, ... или I, II, III, IV, ...

Значение данного атрибута может быть «1», «a», «A», «i», или «I». Для некоторых элементов значение атрибута может быть пустым. В этом случае номер не отображается.

```
10192 <define name="common-num-format-attlist" combine="interleave">
10193     <choice>
10194         <attribute name="style:num-format">
10195             <choice>
```

```
                <value>1</value>
                <value>i</value>
                <value>I</value>
                <ref name="string"/>
                <empty/>
            </choice>
        </attribute>
        <group>
            <attribute name="style:num-format">
                <choice>
                    <value>a</value>
                    <value>A</value>
                </choice>
            </attribute>
            <ref name="style-num-letter-sync-attlist"/>
        </group>
        <empty/>
    </choice>
</define>
```

12.2.3 Синхронизация букв в форматах номера

Если буквы используются в алфавитном порядке для нумерации, то существует два пути для обработки переполнения в рамках одного разряда символов, как описано ниже.

- Вставляется новый символ. Его начальное значение A, и добавление осуществляется каждый раз, когда невозможно определить следующее значение символа, не выходя при этом за рамки разряда. Последовательность нумерации в этом случае будет приблизительно такой a, b, c, ..., z, aa, ab, ac, ..., az, ba, ... и т. д.
- Вставляемый символ всегда имеет то же значение, что и следующий символ. Последовательность нумерации в этом случае будет приблизительно такой a, b, c, ..., z, aa, bb, cc, ..., zz, aaa, ... и т. д. Такой процесс называется **синхронизацией букв**.

Атрибут `style:num-letter-sync` определяет, будет ли производиться синхронизация букв.

```
<define name="style-num-letter-sync-attlist" combine="interleave">
    <optional>
        <attribute name="style:num-letter-sync">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

12.3 Метаданные отслеживания изменений

Метаданные для отслеживания изменений содержатся внутри элемента `<office:change-info>`. Он содержит имя автора и дату создания отслеженного изменения, а в качестве дополнения — комментарий.

```
<define name="office-change-info">
    <element name="office:change-info">
        <ref name="dc-creator"/>
        <ref name="dc-date"/>
        <zeroOrMore>
            <ref name="text-p"/>
        </zeroOrMore>
    </element>
</define>
```

Автор

Элемент `<dc:creator>`, как описано в разделе 3.1.7, задает имя автора, который вносил поправки в документ.

Дата и время

Элемент `<dc:date>`, как описано в разделе 3.1.9, задает дату и время, когда имели место изменения.

Комментарий

Дополнительный комментарий может быть включен как элементы `<text:p>`.

12.4 Таблицы датчиков событий

Многие объекты, такие как элементы управления, изображения, текстовые блоки или целый документ, поддерживают события. Событие связывает наступление особого условия с действием, которое осуществляется в случае его возникновения. Например, если пользователь помещает курсор над изображением, в этом случае запускается некоторое действие, поддерживаемое офисным приложением. Это событие, называемое «on-mouse-over», может быть связано с макрокомандой, которая выполняется в момент достижения условия, то есть, когда пользователь помещает курсор над изображением.

XML-представление событий и таблиц событий имеет структуруприведенную ниже.

- Все элементы событий, связанные с объектом, расположены в контейнерном элементе, называемом `<office:event-listeners>`.
- Каждая связь «событие-в-действие» регистрируется в одном `<script:event-listener>` элементе.
- В зависимости от вида действия, которое вызывает событие, используются следующие элементы:
  — элемент `<script:event-listener>` представляет события, связанные с макрокомандой или скриптом;
  — элемент `<presentation:event-listener>` представляет события, связанные с действием, характерным для презентации, например, перейти на следующую страницу (события презентации описаны в разделе 9.9).

Элемент `<office:event-listeners>` определяет таблицу событий,связанную с конкретным объектом.

```
<define name="office-event-listeners">
    <element name="office:event-listeners">
        <zeroOrMore>
            <choice>
                <ref name="script-event-listener"/>
                <ref name="presentation-event-listener"/>
            </choice>
        </zeroOrMore>
    </element>
</define>
```

12.4.1 Датчик событий

Элемент `<script:event-listener>` связывает событие с макрокомандой.

```
<define name="script-event-listener" combine="interleave">
    <element name="script:event-listener">
        <ref name="script-event-listener-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<script:event-listener>` могут быть связаны следующие атрибуты;

- имя события;
- язык скрипта;
- имя и расположение макрокоманды.

Имя события

Атрибут `script:event-name` задает имя события. После того как события станут доступными, их имена и значения зависят от приложения и языка скрипта. Имени должен предшествовать префикс пространства имен, для того чтобы соответствующее пространство имен вместе с именем события могли быть использованы для идентификации семантики события. Для событий, определенных в событийной модели DOM, рекомендуется использовать имена событий, описанные в § 1.4.2 [DOMEvents]. Соответствующее пространство имен — "http://www.w3.org/2001/xml-events" .

```
<define name="script-event-listener-attlist" combine="interleave">
    <attribute name="script:event-name">
        <ref name="string"/>
    </attribute>
</define>
```

Язык скрипта

Атрибут `script:language` определяет язык скрипта, в котором написан макрос или скрипт, связанный с событием (см. раздел 2.5.1).

```
<define name="script-event-listener-attlist" combine="interleave">
    <attribute name="script:language">
        <ref name="string"/>
    </attribute>
</define>
```

Имя и расположение макроса

Код макроса, который должен вызываться для события, может быть определен либо с помощью IRI нотации [XLink], или простым названием, описанным в атрибуте `script:macro-name`. Если используется XLink, то IRI может иметь произвольный протокол, например один, который кодирует название библиотеки макросов вместе с названием макроса, определенным в этой библиотеке. Как и XLink IRI, так и простое название зависят от языка скрипта.

```
<define name="script-event-listener-attlist" combine="interleave">
    <choice>
        <attribute name="script:macro-name">
            <ref name="string"/>
        </attribute>
        <group>
            <attribute name="xlink:href">
                <ref name="anyURI"/>
            </attribute>
            <optional>
```

```
                <attribute name="xlink:type" a:defaultValue="simple">
                    <value>simple</value>
                </attribute>
            </optional>
            <optional>
                <attribute name="xlink:actuate" a:defaultValue="onRequest">
                    <value>onRequest</value>
                </attribute>
            </optional>
        </group>
    </choice>
</define>
```

12.5 Математическое содержимое

Математическое содержание представлено MathML 2.0 (см. [MathML]).

```
<define name="math-math">
    <element name="math:math">
        <ref name="mathMarkup"/>
    </element>
</define>

<!-- To avoid inclusion of the complete MathML schema, anything -->
<!-- is allowed within a math:math top-level element            -->
<define name="mathMarkup">
    <zeroOrMore>
        <choice>
            <attribute>
                <anyName/>
            </attribute>
            <text/>
            <element>
                <anyName/>
                <ref name="mathMarkup"/>
            </element>
        </choice>
    </zeroOrMore>
</define>
```

12.6 DDE-соединения

Динамическая связь обмена данных (DDE) состоит из параметров для DDE целевого приложения, имени файла и командной строки. DDE-соединение принимает параметр, который определяет, будет ли оно обновлено автоматически или по запросу пользователя. Каждое DDE-соединение должно быть именовано.

Все элементы, использующие DDE-соединения, должны включать в себя их содержание (или их представление) так, чтобы документы, использующие DDE, могли должным образом отображаться на компьютерах, которые не поддерживают механизм DDE, или там, где цель DDE не доступна. Приложения должны сохранять информацию о DDE-соединении, даже если они не могут использовать его так, чтобы другие приложения могли использовать возможности DDE.

12.6.1 Контейнер описания DDE-соединения

В пределах текстовых документов и электронных таблиц описание DDE-соединения содержится в одном элементе объявления. Для текстовых документов элемент `<text:dde-connection-decls>` описан в разделе 4.7. Для электронных таблиц элемент `<table:dde-links>` описан в разделе 8.10.

12.6.2 Описание DDE-соединения для текстовых полей

Каждое DDE-соединение, используемое текстовым полем, описывается при помощи элемента объявления. Многократные поля DDE могут обращаться к одному DDE-соединению с использованием одного имени. Элемент описания не имеет никакого содержания.

```
<define name="text-dde-connection-decl">
    <element name="text:dde-connection-decl">
        <ref name="text-dde-connection-decl-attlist"/>
        <ref name="common-dde-connection-decl-attlist"/>
    </element>
</define>
```

С элементом `<text:dde-connection-decl>` могут быть связаны следующие атрибуты:

- имя соединения;
- целевое DDE-приложение;
- целевая DDE-тема;
- целевой DDE-объект;
- флаг автоматического обновления.

Имя соединения

Атрибут `office:name` задает имя соединения, к которому оно будет обращаться.

```
<define name="text-dde-connection-decl-attlist" combine="interleave">
    <attribute name="office:name">
        <ref name="string"/>
    </attribute>
</define>
```

Целевое приложение

Атрибут `office:dde-application` задает имя целевого приложения, которое будет использовано для DDE-соединения.

```
<define name="common-dde-connection-decl-attlist" combine="interleave">
    <attribute name="office:dde-application">
        <ref name="string"/>
    </attribute>
</define>
```

Пример —

Именем для приложений OpenOffice.org будет `soffice`. Поэтому внутренние DDE-ссылки имеют атрибут `text:dde-application="soffice"`.

Целевая тема

Атрибут `office:dde-topic` задает имя темы, которое будет использовано для DDE-соединения.

```
<define name="common-dde-connection-decl-attlist" combine="interleave">
    <attribute name="office:dde-topic">
        <ref name="string"/>
    </attribute>
</define>
```

Пример —

Программное обеспечение OpenOffice.org интерпретирует тему DDE в качестве имени файла.

Целевой объект

Атрибут `office:dde-item` определяет информацию, которую должно предоставить целевое приложение.

```
<define name="common-dde-connection-decl-attlist" combine="interleave">
    <attribute name="office:dde-item">
        <ref name="string"/>
    </attribute>
</define>
```

Пример —

Если целевым DDE-приложением соединения является OpenOffice.org Writer, то элемент задает имя закладки. OpenOffice.org предоставляет запрашиваемое текущее содержание текста для приложения.

Автоматическое обновление

Офисные приложения по умолчанию автоматически обновляют DDE-ссылки. Если ручное обновление связи предпочтительнее, то атрибут `text:automatic-`

update может использоваться для обновления ссылок DDE-соединения только по запросу пользователя.

Если значением атрибута является true, тогда от приложения ожидается автоматическое обновление DDE-ссылок. Если значение атрибута false, то DDE-ссылки обновляются только по запросу пользователя.

```
<define name="common-dde-connection-decl-attlist" combine="interleave">
    <optional>
        <attribute name="office:automatic-update" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

12.6.3 Объявление DDE-соединений для таблиц

Данные DDE-соединений таблиц содержатся в элементе <office:dde-source>. Использование этого элемента различно для электронных таблиц и таблиц текстовых документов. Для таблиц текстовых документов элемент непосредственно содержится в элементе таблицы <table:table>. Для документов электронных таблиц он содержится в элементе <table:dde-link>, который описывает одиночное DDE-соединение.

Элемент <table:dde-link> состоит из элемента источника данных DDE <office:dde-source> и простого элемента таблицы, который может использоваться для кэширования данных источника DDE. Таблица не нуждается в названии и не содержит информацию о стиле. Используются только данные, содержащиеся в ячейке атрибутов. Сами ячейки остаются пустыми.

```
<define name="table-dde-link">
    <element name="table:dde-link">
        <ref name="office-dde-source"/>
        <ref name="table-table"/>
    </element>
</define>
```

Элемент <office:dde-source> поддерживает атрибуты office:dde-application, office:dde-topic, office:dde-item и office:automatic-update, как описано в разделе 12.6.2. В дополнение к этому, он поддерживает следующие атрибуты:

- имя соединения;
- режим преобразования.

```
<define name="office-dde-source">
```

```
    <element name="office:dde-source">
        <ref name="office-dde-source-attlist"/>
        <ref name="common-dde-connection-decl-attlist"/>
    </element>
</define>
```

Имя соединения

Атрибут `office:name` задает имя соединения, по которому к нему можно будет обращаться.

```
<define name="office-dde-source-attlist" combine="interleave">
    <optional>
        <attribute name="office:name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Режим преобразования

Атрибут `office:conversion-mode` задает метод, по которому сервер DDE преобразует данные в числа. Существует три возможных значения:

- `into-default-style-data-style` — числа преобразуются в стиль данных, который выбран как стиль по умолчанию;
- `into-english-number` — числа преобразуются по умолчанию в английском формате;
- `keep-text` — числа не преобразуются и воспринимаются как текст.

```
<define name="office-dde-source-attlist" combine="interleave">
    <optional>
        <attribute name="office:conversion-mode"
                    a:defaultValue="into-default-style-data-style">
            <choice>
                <value>into-default-style-data-style</value>
                <value>into-english-number</value>
                <value>keep-text</value>
            </choice>
        </attribute>
    </optional>
</define>
```

13 SMIL-анимации

В этом разделе описаны элементы, основывающиеся на [SMIL20], и атрибуты, которые могут быть использованы в формате OpenDocument для создания анимационных эффектов.

13.1 Основные элементы анимации

Основные элементы анимации являются непосредственными производными от основных элементов анимации, определенных в § 3.5 и § 12.5 из [SMIL20], и в разделе § 19.2 в [SVG].

13.1.1 Анимировать

Элемент `<anim:animate>` ведет себя подобно элементу [SMIL20] `<smil:animate>` (см. § 3.5.1 из [SMIL20] для получения подробной информации).

```
<define name="animation-element" combine="choice">
    <element name="anim:animate">
        <ref name="common-anim-target-attlist"/>
        <ref name="common-anim-named-target-attlist"/>
        <ref name="common-anim-values-attlist"/>
        <ref name="common-anim-spline-mode-attlist"/>
        <ref name="common-spline-anim-value-attlist"/>
        <ref name="common-repeat-timing-attlist"/>
        <ref name="common-fill-timing-attlist"/>
        <ref name="common-anim-add-accum-attlist"/>
    </element>
</define>
```

13.1.2 Множество

Элемент `<anim:set>` ведет себя подобно элементу [SMIL20] `<smil:set>` (см. § 3.5.2 из [SMIL20] для получения подробной информации).

```
<define name="animation-element" combine="choice">
    <element name="anim:set">
        <ref name="common-anim-target-attlist"/>
        <ref name="common-anim-named-target-attlist"/>
        <ref name="common-anim-set-values-attlist"/>
        <ref name="common-fill-timing-attlist"/>
        <ref name="common-anim-add-accum-attlist"/>
    </element>
</define>
```

13.1.3 Анимированное движение

Элемент `<anim:animateMotion>` ведет себя подобно элементу [SVG] `<svg:animateMotion>` (см. § 19.2.12 из [SVG] и § 3.5.3 из [SMIL20] для получения подробной информации).

```
<define name="animation-element" combine="choice">
    <element name="anim:animateMotion">
        <ref name="anim-animate-motion-attlist"/>
        <ref name="common-anim-target-attlist"/>
        <ref name="common-anim-named-target-attlist"/>
        <ref name="common-anim-add-accum-attlist"/>
        <ref name="common-anim-values-attlist"/>
        <ref name="common-fill-timing-attlist"/>
        <ref name="common-spline-anim-value-attlist"/>
    </element>
</define>
```

Траектория движения

Атрибут [SVG] `svg:path` может использоваться для определения траектории, вдоль которой движется элемент (см. § 19.2.12 из [SVG] для получения подробной информации).

```
<define name="anim-animate-motion-attlist" combine="interleave">
    <optional>
        <attribute name="svg:path">
            <ref name="pathData"/>
        </attribute>
    </optional>
</define>
```

Начальное положение

Атрибут [SVG] `svg:origin` может использоваться для определения начального положения (см. § 19.2.12 из [SVG] для получения подробной информации).

```
<define name="anim-animate-motion-attlist" combine="interleave">
    <optional>
        <attribute name="svg:origin">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Способ вычисления

Атрибут [SMIL20] `smil:calcMode` используется для определения режима интерполяции анимации (см. § 19.2.12 из [SVG] для получения подробной информации).

```
<define name="anim-animate-motion-attlist" combine="interleave">
    <optional>
        <attribute name="smil:calcMode" a:defaultValue="paced">
            <choice>
                <value>discrete</value>
                <value>linear</value>
                <value>paced</value>
                <value>spline</value>
            </choice>
        </attribute>
    </optional>
</define>
```

13.1.4 Анимация цвета

Элемент `<anim:animateColor>` ведет себя подобно элементу [SMIL20] `<smil:animateColor>` (см. § 3.5.4 из [SMIL20] для получения подробной информации).

```
<define name="animation-element" combine="choice">
    <element name="anim:animateColor">
        <ref name="common-anim-target-attlist"/>
        <ref name="common-anim-named-target-attlist"/>
        <ref name="common-anim-add-accum-attlist"/>
        <ref name="common-anim-values-attlist"/>
        <ref name="common-anim-spline-mode-attlist"/>
        <ref name="common-spline-anim-value-attlist"/>
        <ref name="anim-animate-color-attlist"/>
        <ref name="common-fill-timing-attlist"/>
    </element>
</define>
```

Интерполяция цвета

Атрибут `anim:color-interpolation` определяет пространство цветов, используемое для интерполяции цвета.

```
<define name="anim-animate-color-attlist" combine="interleave">
    <optional>
        <attribute name="anim:color-interpolation" a:defaultValue="rgb">
            <choice>
                <value>rgb</value>
                <value>hsl</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Направление интерполяции цвета

Атрибут `anim:color-interpolation-direction` определяет направление, используемое для интерполяции цвета. Этот атрибут работает только для цветового пространства HSL.

```
<define name="anim-animate-color-attlist" combine="interleave">
    <optional>
        <attribute name="anim:color-interpolation-direction"
                                a:defaultValue="clockwise">
            <choice>
                <value>clockwise</value>
                <value>counter-clockwise</value>
            </choice>
        </attribute>
    </optional>
</define>
```

13.1.5 Анимация преобразования

Элемент `<anim:animateTransform>` основывается на элементе [SVG] `<svg:animateTransform>` (см. § 19.2.14 из [SVG] для получения подробной информации).

```
<define name="animation-element" combine="choice">
    <element name="anim:animateTransform">
        <ref name="common-anim-target-attlist"/>
        <ref name="common-anim-named-target-attlist"/>
        <ref name="common-anim-add-accum-attlist"/>
        <ref name="common-anim-values-attlist"/>
        <ref name="anim-animate-transform-attlist"/>
        <ref name="common-fill-timing-attlist"/>
    </element>
</define>
```

Тип преобразования

Атрибут [SVG] `svg:type` используется для определения типа преобразования (см. § 19.2.14 из [SVG] для получения подробной информации).

```
<define name="anim-animate-transform-attlist" combine="interleave">
    <attribute name="svg:type">
        <choice>
            <value>translate</value>
            <value>scale</value>
            <value>rotate</value>
            <value>skewX</value>
            <value>skewY</value>
        </choice>
    </attribute>
</define>
```

13.1.6 Фильтр перехода

Элемент `<anim:transitionFilter>` основывается на элементе [SMIL20] `<smil:transitionFilter>` (см. § 12.5.1 из [SMIL20] для получения подробной информации).

```
<define name="animation-element" combine="choice">
    <element name="anim:transitionFilter">
        <ref name="common-anim-target-attlist"/>
        <ref name="common-anim-add-accum-attlist"/>
        <ref name="common-anim-values-attlist"/>
        <ref name="common-anim-spline-mode-attlist "/>
        <ref name="anim-transition-filter-attlist"/>
        <ref name="common-fill-timing-attlist"/>
    </element>
</define>
```

Тип перехода

Атрибут [SMIL20] `smil:type` используется для определения типа или семейства перехода (см. § 12.8 из [SMIL20] для получения списка поддерживаемых типов).

```
<define name="anim-transition-filter-attlist" combine="interleave">
    <attribute name="smil:type">
        <ref name="string"/>
    </attribute>
</define>
```

Подтип перехода

Атрибут [SMIL20] `smil:subtype` может использоваться для определения подтипа перехода (см. § 12.8 из [SMIL20] для получения списка поддерживаемых подтипов).

```
<define name="anim-transition-filter-attlist" combine="interleave">
    <optional>
        <attribute name="smil:subtype">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Направление перехода

Атрибут [SMIL20] `smil:direction` может использоваться для определения направления перехода (см. § 12.4.1 из [SMIL20] для получения подробной информации).

```
<define name="anim-transition-filter-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="smil:direction" a:defaultValue="forward">
            <choice>
                <value>forward</value>
                <value>reverse</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Цвет затухания

Атрибут [SMIL20] `smil:fadeColor` может использоваться для определения цвета переходов затухания для переходов, которые используют начальный или конечный цвета (см. § 12.5.1 из [SMIL20] для получения подробной информации).

```
<define name="anim-transition-filter-attlist" combine="interleave">
    <optional>
        <attribute name="smil:fadeColor">
            <choice>
                <value>forward</value>
                <value>reverse</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Способ перехода

Атрибут [SMIL20] `smil:mode` используется для определения, будет ли анимированный элемент осуществлять переход «внутрь» или «наружу» (см. § 12.5.1 из [SMIL20] для получения подробной информации).

```
<define name="anim-transition-filter-attlist" combine="interleave">
    <optional>
        <attribute name="smil:mode" a:defaultValue="in">
            <choice>
                <value>in</value>
                <value>out</value>
            </choice>
        </attribute>
    </optional>
</define>
```

13.2 Атрибуты анимационной модели

Анимационная модель использует такой же принцип и синтаксис, какой описан в § 3 из [SMIL20].

13.3 Общие атрибуты анимации

Идентификатор элемента

Атрибут `anim:id` определяет идентификатор, который используется для опознавания элемента внутри документа.

```
<define name="common-anim-attlist" combine="interleave">
    <optional>
        <attribute name="anim:id">
            <ref name="ID"/>
        </attribute>
    </optional>
</define>
```

13.3.1 Целевые атрибуты анимации

Целевой элемент

Атрибут [SMIL20] `smil:targetElement` используется для определения целевого элемента, который будет анимирован (см. § 3.4.1 из [SMIL20] для получения подробной информации). Для получения подробной информации об использовании этого атрибута в документах презентации см. раздел 9.8.2.

```
<define name="common-anim-target-attlist" combine="interleave">
    <optional>
        <attribute name="smil:targetElement">
            <ref name="IDREF"/>
        </attribute>
    </optional>
</define>
```

Целевой атрибут

Атрибут [SMIL20] `smil:attributeName` используется для определения целевого атрибута по имени (см. § 3.4.1 из [SMIL20] для получения подробной информации). Для получения подробной информации об использовании этого атрибута в документах презентации см. раздел 9.8.2).

```
<define name="common-anim-named-target-attlist" combine="interleave">
    <attribute name="smil:attributeName">
        <ref name="string"/>
    </attribute>
</define>
```

Подпункт целевого элемента

Атрибут `anim:sub-item` определяет необязательный подпункт целевого элемента. Возможные значения для этого элемента зависят от типа документа и типа целевого элемента (см. раздел 9.8.2 для получения подробной информации об использовании этого атрибута в документах презентации).

```
<define name="common-anim-target-attlist" combine="interleave">
    <optional>
        <attribute name="anim:sub-item">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

13.3.2 Функциональные атрибуты анимации

Список значений

Атрибут [SMIL20] `smil:values` определяет значения, используемые для анимации целевого элемента (см. § 3.4.2 из [SMIL20] для получения более подробной информации).

```
<define name="common-anim-values-attlist" combine="interleave">
    <optional>
        <attribute name="smil:values">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Способ вычисления

Атрибут [SMIL20] `smil:calcMode` используется для определения режима интерполяции функции анимации (см. § 3.4.2 из [SMIL20] для получения более подробной информации).

```
<define name="common-anim-spline-mode-attlist" combine="interleave">
    <optional>
        <attribute name="smil:calcMode" a:defaultValue="discrete">
            <choice>
                <value>discrete</value>
                <value>linear</value>
                <value>paced</value>
                <value>spline</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Ключевые моменты

Атрибут [SMIL20] `smil:keyTimes` определяет раскадровку анимации (см. § 3.7.1 из [SMIL20] для получения более подробной информации).

```
<define name="common-spline-anim-value-attlist" combine="interleave">
    <optional>
        <attribute name="smil:keyTimes">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Ключевые сплайны

Атрибут [SMIL20] `smil:keySplines` определяет кубическую функцию Безье, которая контролирует межкадровый интервал (см. § 3.7.1 из [SMIL20] для получения более подробной информации).

```
<define name="common-spline-anim-value-attlist" combine="interleave">
    <optional>
        <attribute name="smil:keySplines">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Накопление

Атрибут [SMIL20] `smil:accumulate` определяет накопление значений функции анимации (см. § 3.4.3 из [SMIL20] для получения более подробной информации).

```
<define name="common-anim-add-accum-attlist" combine="interleave">
    <optional>
        <attribute name="smil:accumulate">
            <choice>
                <value>none</value>
                <value>sum</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Добавление

Атрибут [SMIL20] `smil:additive` определяет, добавляется ли какое-либо значение в функции анимации (см. § 3.4.3 из [SMIL20] для получения более подробной информации).

```
<define name="common-anim-add-accum-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="smil:additive">
            <choice>
                <value>replace</value>
                <value>sum</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Формула

Атрибут `anim:formula` определяет формулу, которая используется в функции анимации. Идентификатор '$' будет заменен значением между 0 и 1 (включительно), которое отражает пропорциональное смещение в длительности анимированных элементов. В некоторых специфичных типах документов могут содержаться дополнительные идентификаторы. Следующее является минимумом поддерживаемой грамматики:

```
identifier = '$' | 'pi'

function = 'abs'|'sqrt'|'sin'|'cos'|'tan'|'atan'|'acos'|'asin'|'exp'|'log'
binary_function = 'min'|'max'

basic_expression =
    number |
    identifier |
    function '(' additive_expression ')' |
    binary_function
        '(' additive_expression ',' additive_expression ')' |
    '(' additive_expression ')'

unary_expression =
    '-' basic_expression |
    basic_expression

multiplicative_expression =
    unary_expression
    (   ( '*' unary_expression )* |
        ( '/' unary_expression )* )

additive_expression =
    multiplicative_expression
    (   ( '+' multiplicative_expression )* |
        ( '-' multiplicative_expression )* )
```

Для получения подробной информации о дополнительных идентификаторах для документов презентаций см. раздел 9.8.2.

Если задан атрибут `anim:formula`, атрибуты: `smil:values`, `smil:to`, `smil:from` и `smil:by` перезаписываются так, как это определено в следующем разделе.

```
<define name="common-anim-values-attlist" combine="interleave">
```

```
    <optional>
        <attribute name="anim:formula">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Функции простой анимации

В дополнение к описанию анимации со списком значений, может использоваться упрощенная версия, использующая атрибуты [SMIL20] `smil:from`, `smil:to` и `smil:by` (см. § 3.4.4 из [SMIL20] для получения более подробной информации).

```
<define name="common-anim-set-values-attlist" combine="interleave">
    <optional>
        <attribute name="smil:to">
            <ref name="string"/>
        </attribute>
    </optional>
</define>

<define name="common-anim-values-attlist" combine="interleave">
    <ref name="common-anim-set-values-attlist"/>
    <optional>
        <attribute name="smil:from">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="smil:by">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

13.4 Распределение времени анимации

Распределение времени анимации использует те же самые понятия и синтаксис, которые определяются в § 10 и § 11 из глав [SMIL20].

13.4.1 Атрибуты распределения времени анимации

Запуск элемента

Атрибут [SMIL20] `smil:begin` может использоваться для определения времени запуска элемента (см. § 10.3.1 из [SMIL20] для получения более подробной информации).

```
<define name="common-begin-end-timing-attlist" combine="interleave">
    <optional>
```

```
10638        <attribute name="smil:begin">
10639            <ref name="string"/>
10640        </attribute>
10641    </optional>
10642 </define>
```

Завершение элемента

Атрибут [SMIL20] `smil:end` может использоваться для определения времени завершения элемента (см. § 10.3.1 из [SMIL20] для получения более подробной информации).

```
10643 <define name="common-begin-end-timing-attlist" combine="interleave">
10644     <optional>
10645         <attribute name="smil:end">
10646             <ref name="string"/>
10647         </attribute>
10648     </optional>
10649 </define>
```

Длительность элемента

Атрибут [SMIL20] `smil:dur` может использоваться для определения длительности элемента (см. § 10.3.1 из [SMIL20] для получения более подробной информации).

```
10650 <define name="common-dur-timing-attlist" combine="interleave">
10651     <optional>
10652         <attribute name="smil:dur">
10653             <ref name="string"/>
10654         </attribute>
10655     </optional>
10656 </define>
```

Синхронизация завершения элемента

Атрибут [SMIL20] `smil:endsync` может использоваться для управления подразумеваемой длительностью контейнеров времени, как функция потомков (см. § 10.3.1 из [SMIL20] для получения более подробной информации).

```
10657 <define name="common-endsync-timing-attlist" combine="interleave">
10658     <optional>
10659         <attribute name="smil:endsync">
10660             <choice>
10661                 <value>first</value>
10662                 <value>last</value>
10663                 <value>all</value>
10664                 <value>media</value>
10665             </choice>
10666         </attribute>
10667     </optional>
10668 </define>
```

Элементы повторения

Атрибуты [SMIL20] `smil:repeatCount` и `smil:repeatDur` определяют поведение повторяющейся анимации (см. § 10.3.1 из [SMIL20] для получения более подробной информации).

```
<define name="common-repeat-timing-attlist" combine="interleave">
    <optional>
        <attribute name="smil:repeatDur">
            <ref name="string"/>
        </attribute>
        <attribute name="smil:repeatCount">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Заполнение

Атрибут [SMIL20] `smil:fill` определяет поведение элемента после окончания анимации (см. § 10.3.1 из [SMIL20] для получения более подробной информации).

```
<define name="common-fill-timing-attlist" combine="interleave">
    <optional>
        <attribute name="smil:fill">
            <choice>
                <value>remove</value>
                <value>freeze</value>
                <value>hold</value>
                <value>auto</value>
                <value>default</value>
                <value>transition</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Заполнение по умолчанию

Атрибут [SMIL20] `smil:fillDefault` определяет поведение по умолчанию для атрибута `smil:fill` (см. § 10.3.1 из [SMIL20] для получения более подробной информации).

```
<define name="common-fill-default-attlist" combine="interleave">
    <optional>
        <attribute name="smil:fillDefault">
            <choice>
                <value>remove</value>
                <value>freeze</value>
                <value>hold</value>
                <value>transition</value>
                <value>auto</value>
                <value>inherit</value>
```

```
            </choice>
        </attribute>
    </optional>
</define>
```

Перезапуск

Атрибут [SMIL20] `smil:restart` может использоваться для определения поведения элемента при перезапуске (см. § 10.3.1 из [SMIL20] для получения более подробной информации).

```
<define name="common-restart-timing-attlist" combine="interleave">
    <optional>
        <attribute name="smil:restart" a:defaultValue="default">
            <choice>
                <value>never</value>
                <value>always</value>
                <value>whenNotActive</value>
                <value>default</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Перезапуск по умолчанию

Атрибут [SMIL20] `smil:restartDefault` может использоваться для определения поведения элемента при перезапуске по умолчанию (см. § 10.3.1 из [SMIL20] для получения более подробной информации).

```
<define name="common-restart-default-attlist" combine="interleave">
    <optional>
        <attribute name="smil:restartDefault" a:defaultValue="inherit">
            <choice>
                <value>never</value>
                <value>always</value>
                <value>whenNotActive</value>
                <value>inherit</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Ускорение

Атрибут [SMIL20] `smil:accelerate` может использоваться для определения простого ускорения времени элемента (см. § 11.1.2 из [SMIL20] для получения более подробной информации).

```
<define name="common-time-manip-attlist" combine="interleave">
    <optional>
        <attribute name="smil:accelerate" a:defaultValue="0.0">
            <ref name="double"/>
```

```
        </attribute>
    </optional>
</define>
```

Замедление

Атрибут [SMIL20] `smil:decelerate` может использоваться для определения простого замедления времени элемента (см. § 11.1.2 из [SMIL20] для получения более подробной информации).

```
<define name="common-time-manip-attlist" combine="interleave">
    <optional>
        <attribute name="smil:decelerate" a:defaultValue="0.0">
            <ref name="double"/>
        </attribute>
    </optional>
</define>
```

Автоматический реверс

Атрибут [SMIL20] `smil:autoreverse` может использоваться для определения автоматического проигрывания в обратном направлении (см. § 11.1.2 из [SMIL20] для получения более подробной информации).

```
<define name="common-time-manip-attlist" combine="interleave">
    <optional>
        <attribute name="smil:autoReverse" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

13.4.2 Параллельные анимации

Элемент `<anim:par>` основан на элементе [SMIL20] `<smil:par>` и определяет параллельный контейнер времени (см. § 10.3.2 из [SMIL20] для получения более подробной информации).

```
<define name="animation-element" combine="choice">
    <element name="anim:par">
        <ref name="common-anim-attlist"/>
        <ref name="common-timing-attlist"/>
        <ref name="common-endsync-timing-attlist"/>
        <zeroOrMore>
            <ref name="animation-element"/>
        </zeroOrMore>
    </element>
</define>

<define name="common-basic-timing-attlist" combine="interleave">
   <ref name="common-begin-end-timing-attlist"/>
   <ref name="common-dur-timing-attlist"/>
   <ref name="common-repeat-timing-attlist"/>
```

```
</define>

<define name="common-timing-attlist" combine="interleave">
   <ref name="common-basic-timing-attlist"/>
   <ref name="common-restart-timing-attlist"/>
   <ref name="common-restart-default-attlist"/>
   <ref name="common-fill-timing-attlist"/>
   <ref name="common-fill-default-attlist"/>
   <ref name="common-time-manip-attlist"/>
</define>
```

13.4.3 Последовательные анимации

Элемент `<anim:seq>` основан на элементе [SMIL20] `<smil:seq>` и определяет последовательный контейнер времени (см. § 10.3.2 из [SMIL20] для получения более подробной информации).

```
<define name="animation-element" combine="choice">
    <element name="anim:seq">
        <ref name="common-anim-attlist"/>
        <ref name="common-endsync-timing-attlist"/>
        <ref name="common-timing-attlist"/>
    </element>
</define>
```

13.4.4 Итеративные анимации

Элемент `<anim:iterate>` определяет параллельный контейнер времени. Отличие от элемента `<anim:par>` заключается в том, что элемент `<anim:iterate>` не определяет эффекты для своего целевого элемента. Вместо этого он повторяется для всех возможных элементов-потомков целевого элемента и выполняет все эффекты элементов-потомков с потомками целевого элемента как целями.

```
<define name="animation-element" combine="choice">
    <element name="anim:iterate">
        <ref name="common-anim-attlist"/>
        <ref name="anin-iterate-attlist"/>
        <ref name="common-timing-attlist"/>
        <ref name="common-endsync-timing-attlist"/>
        <zeroOrMore>
            <ref name="animation-element"/>
        </zeroOrMore>
    </element>
</define>
```

Целевой элемент

Атрибут [SMIL20] `smil:targetElement` определяет целевой элемент, к потомкам которого должны быть применены эффекты (см. раздел 9.8.2 для

получения подробной информации об использовании атрибутов в документах презентации).

```
<define name="anin-iterate-attlist" combine="interleave">
    <optional>
        <attribute name="smil:targetElement">
            <ref name="IDREF"/>
        </attribute>
    </optional>
</define>
```

Тип итерации

Атрибут `anim:iterate-type` определяет порядок повторения элементов-потомков целевого элемента. Возможные значения зависят от типа документа и типа целевого элемента (см. раздел 9.8.2 для получения подробной информации об использовании атрибутов в документах презентации).

```
<define name="anin-iterate-attlist" combine="interleave">
    <optional>
        <attribute name="anim:iterate-type">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Интервал между итерациями

Атрибут `anim:iterate-interval` определяет задержку между выполнением эффектов потомков этого элемента. Эффекты следующего потомка целевого элемента начинаются, когда истекает заданное время с момента начала эффекта предыдущего потомка. Интервал между итерациями в ноль секунд задает поведение, аналогичное поведению при использовании элемента `<anim:par>`.

```
<define name="anin-iterate-attlist" combine="interleave">
    <optional>
        <attribute name="anim:iterate-interval">
            <ref name="duration"/>
        </attribute>
    </optional>
</define>
```

13.5 Элементы мультимедиа

13.5.1 Звук

Элемент `<anim:audio>` основан на элементе [SMIL20] `<smil:audio>`. Это позволяет во время анимации проигрывать аудиопотоки (см. § 7.3.1 из [SMIL20] для получения более подробной информации).

```
<define name="animation-element" combine="choice">
    <element name="anim:audio">
        <ref name="common-anim-attlist"/>
        <ref name="anim-audio-attlist"/>
        <ref name="common-basic-timing-attlist"/>
    </element>
</define>
```

Источник

Атрибут `xlink:href` определят IRI аудиопотока.

```
 <define name="anim-audio-attlist" combine="interleave">
    <optional>
        <attribute name="xlink:href">
            <ref name="anyURI"/>
        </attribute>
    </optional>
</define>
```

Громкость звука

Атрибут `anim:audio-level` определяет громкость проигрывания записи. Это значение является числом из диапазона от 0 (неслышный) до 1 (громкость звука в системе).

```
<define name="anim-audio-attlist" combine="interleave">
    <optional>
        <attribute name="anim:audio-level">
            <ref name="double"/>
        </attribute>
    </optional>
</define>
```

13.6 Специальные элементы

13.6.1 Команда

Элемент `<anim:command>` используется для отправки общих команд приложению во время анимации. Доступные типы команд и их параметры зависят

от типа документа и типа целевого элемента (см. раздел 9.8.2 для получения подробной информации об использовании элемента в документах презентации).

```
<define name="animation-element" combine="choice">
    <element name="anim:command">
        <ref name="common-anim-attlist"/>
        <ref name="anim-command-attlist"/>
        <ref name="common-begin-end-timing-attlist"/>
        <ref name="common-anim-target-attlist"/>
        <zeroOrMore>
            <element name="anim:param">
                <attribute name="anim:name"/>
                <attribute name="anim:value"/>
            </element>
        </zeroOrMore>
    </element>
</define>
```

Команда

Атрибут `anim:command` определяет команду, которая будет выполняться приложением после начала анимации.

```
<define name="anim-command-attlist" combine="interleave">
    <attribute name="anim:command">
        <ref name="string"/>
    </attribute>
</define>
```

14 Стили

Многие объекты в офисных документах имеют свойства форматирования. Свойство форматирования влияет на визуальное представление объекта, но оно не воздействует ни на содержимое, ни на структуру документа. Примерами свойств форматирования являются:

- семейство шрифтов;
- размер шрифта;
- цвет шрифта;
- поля страницы.

В формате OpenDocument свойства форматирования хранятся в стиле. Это влияет на пользовательский интерфейс в типичных офисных приложениях, где свойства форматирования могут быть присвоены объекту непосредственно или косвенно, применением стиля к объекту. Применение свойств форматирования непосредственно к объекту дает тот же результат, что и применение безымянного стиля с теми же свойствами к этому объекту. Следовательно, стили пользовательского интерфейса остаются принципиально неизменными в формате OpenDocument, в то время как свойства форматирования, применяемые непосредственно к объекту, принимаются как безымянные стили. В процессе использования безымянного стиля ему может быть присвоено имя, после чего он становится автоматическим стилем.

Использование стилей для хранения свойств форматирования обусловлено двумя основными причинами.

- Формат и разметка документа отделяются от его содержимого.
- Если два или более объектов имеют одинаковые свойства форматирования и стиля, то свойства форматирования, которые непосредственно присваиваются объекту, могут быть представлены одним автоматическим стилем для всех объектов. Это экономит пространство на диске и позволяет стилям интегрироваться в общий стиль документа.

Этот раздел посвящен описанию различных стилей.

14.1 Элемент стиля

Некоторые семейства стилей очень похожи по структуре и могут быть представлены одним элементом. Например, элемент `<style:style>` может представлять стили абзаца, текста и графики.

Самостоятельные семейства стилей, которые используют эти элементы, описываются отдельно. В пределах данного раздела описываются общие атрибуты элемента стиля.

```
<define name="style-style">
    <element name="style:style">
        <ref name="style-style-attlist"/>
        <ref name="style-style-content"/>
        <zeroOrMore>
            <ref name="style-map"/>
        </zeroOrMore>
    </element>
</define>
```

С элементом `<style:style>` могут быть связаны следующие атрибуты:

- имя стиля;
- отображаемое имя;
- семейство стилей;
- родительский стиль;
- следующий стиль;
- стиль списка;
- имя мастер-страницы;
- автоматическое обновление;
- имя стиля данных;
- класс;
- уровень нумерации структуры.

Имя стиля

Атрибут `style:name` определяет имя стиля. Этот атрибут, объединенный с атрибутом `style:family`, уникально идентифицирует стиль. Каждый из элементов `<office:styles>`, `<office:automatic-styles>` и `<office:master-styles>` не должен содержать двух стилей из одного семейства с одинаковым именем.

Для автоматических стилей имя генерируется во время экспорта документа. Если документ экспортируется несколько раз, то каждый раз будут генерироваться новые имена.

В XML-документе имя каждого стиля уникально, независимо от того, какой язык пользовательского интерфейса был выбран для офисного приложения. Обычно это имена, используемые в англоязычной версии пользовательского интерфейса.

```
<define name="style-style-attlist" combine="interleave">
    <attribute name="style:name">
        <ref name="styleName"/>
    </attribute>
</define>
```

Отображаемое имя

Атрибут `style:display-name` определяет имя стиля, каким оно должно быть отображено в пользовательском интерфейсе. В отличие от имени стиля, это имя может содержать некоторые произвольные символы. Если данный атрибут не представлен, отображаемое имя совпадает с именем стиля.

```
<define name="style-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:display-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Семейство стилей

Атрибут `style:family` определяет принадлежность к семейству стилей, например, абзац, текст или врезка. Он может принимать одно из следующих значений: `paragraph`, `text`, `section`, `table`, `table-column`, `table-row`, `table-cell`, `table-page`, `chart`, `default`, `drawing-page`, `graphic`, `presentation`, `control`, `ruby`.

Родительский стиль

Атрибут `style:parent-style-name` определяет имя родительского стиля. Если родительский стиль не указан, то используется значение по умолчанию, определенное в приложении. Родительский стиль не может быть автоматическим и должен существовать.

```
<define name="style-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:parent-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Следующий стиль

Атрибут `style:next-style-name` определяет стиль, используемый в следующем абзаце, при вставке конца абзаца в пользовательском интерфейсе. По умолчанию текущий стиль используется как следующий.

```
<define name="style-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:next-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Стиль списка

Стиль абзаца и стили других семейств, содержащие свойства абзаца (например графические стили), могут иметь связанный стиль списка. Это относится к автоматическим и общим стилям.

Стиль списка, определяемый атрибутом `style:list-style-name`, применяется лишь к заголовкам и абзацам, которые содержатся в списке, не определяющем свой стиль самостоятельно, и не унаследовавшим его ни от одного из родителей.

Значение атрибута `style:list-style-name` может быть пустым. В этом случае связь со стилем списка, унаследованная от родителя, будет удалена.

```
<define name="style-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:list-style-name">
            <choice>
                <ref name="styleName"/>
                <empty/>
            </choice>
        </attribute>
    </optional>
</define>
```

Имя мастер-страницы

Стиль абзаца или таблицы может иметь связанный атрибут `style:master-page-name`. Это относится к автоматическим и общим стилям. Если этот атрибут связан со стилем, разрыв страницы вставляется, когда стиль применен и установленная мастер-страница применяется на предыдущей странице.

Этот атрибут игнорируется, если он связан со стилем абзаца, применяемым к абзацу внутри таблицы.

```
<define name="style-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:master-page-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Автоматическое обновление

Атрибут `style:auto-update` определяет, будет ли автоматически обновляться стиль при изменении свойств форматирования объекта, которому этот стиль присвоен. Например, существует стиль абзаца, который содержит свойства форматирования, определяющие, что текст абзаца выравнивается по центру и этот стиль применен к абзацу. Если пользователь вручную изменяет форматирование этого абзаца текста на выравнивание по правой стороне и значение атрибута `style:auto-update` установлено `true`, то стиль абзаца будет автоматически обновляться с учетом изменений форматирования в новом абзаце, и каждый абзац, использующий этот стиль абзаца, будет выравнен по правой стороне. Атрибут может иметь значение `true` или `false`.

```
<define name="style-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:auto-update" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Имя стиля данных

Стиль ячейки таблицы может иметь связанный стиль данных. Это относится к автоматическим и общим стилям. Обращение к стилю данных происходит через атрибут `style:data-style-name`. Для получения дополнительной информации о стилях данных обратитесь к разделу 14.7.

```
<define name="style-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:data-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Класс

Стиль может принадлежать некоторому классу стилей. Класс является некоторой строкой. Класс не несет смысловой нагрузки в пределах формата файла, но в пользовательском интерфейсе его вклад позволяет оценить пример вывода списка стилей, где стили сгруппированы по названию класса.

```
<define name="style-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:class">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Уровень нумерации структуры

Для стиля из семейства абзацев атрибут `style:default-outline-level` определяет значение уровня структуры по умолчанию. Он берет числовой атрибут `text:outline-level` из элемента заголовка `<text:h>`. Если этот атрибут существует для стиля абзаца, и этот стиль присвоен абзацу пользователем, то офисное приложение должно преобразовать абзац в заголовок заданного уровня. Однако атрибут не влияет на разграничение заголовков и абзацев в формате файла. Разграничение между заголовками и параметрами производится элементами `<text:h>` или `<text:p>`. Если элемент `<text:p>` ссылается на стиль абзаца с атрибутом `style:default-outline-level`, абзац останется абзацем и не станет заголовком.

```
<define name="style-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:default-outline-level">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Свойства форматирования

Если стилю присвоены атрибуты форматирования, то элемент стиля содержит один или несколько элементов, содержащих свойства форматирования. Для получения более подробной информации об этих элементах обратитесь к разделу 15.

Образец стиля

Пример —

Представление стиля абзаца «Text body» в формате OpenDocument.

```
<style:style style:name="Text body" style:family="paragraph"
             style:parent-style-name="Standard">
    <style:paragraph-properties fo:margin-top="0cm"
                                fo:margin-bottom=".21cm"/>
</style:style>
```

14.1.1 Соотнесение стилей

Элемент `<style:map>` устанавливает соотнесение с другим стилем при выполнении определенных условий. Если стиль содержит такие соотнесения, то он называется условным. Для каждого условия существует один элемент, использующийся стилем.

Условные стили обычно поддерживаются стилями абзаца, содержащимися в текстовых документах, и стилями ячеек таблиц в электронных таблицах. Условные стили также поддерживаются стилями данных.

```
<define name="style-map">
    <element name="style:map">
        <ref name="style-map-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<style:map>` могут быть связаны следующие атрибуты:

- условие;
- применяемый стиль;
- адрес базовой ячейки.

Условие

Атрибут `style:condition` определяет условие, при выполнения которого будет применяться соотнесение стилей.

Значением этого атрибута является логическое выражение. Синтаксис этого выражения сходен с синтаксисом Xpath. Если приложение обнаруживает не разделяющее однозначно условие, то игнорируется весь элемент `<style:map>`.

Для стилей абзацев применимы следующие условия:

- `list-level()=`*n*, где *n* — число от 1 до 10;
- `outline-level()=`*n*, где *n* — число от 1 до 10;
- `table()` и `table-header()`
- `section()`
- `header()` и `footer()`
- `footnote()` и `endnote()`

Для стилей ячеек применимы следующие условия:

- `is-true-formula(`*formula*`)`
- `cell-content-is-between(`*value*`, `*value*`)`
- `cell-content-is-not-between(`*value*`, `*value*`)`
- `cell-content() `*operator value*, где *operator* — один из: `'<'`, `'>'`, `'<='`, `'>='`, `'='`, `'!='`, и значение является *numberValue*, *string* или *formula*
- *numberValue* — целое или десятичное число, число больше тысячи не может содержать десятичных знаков;
- *string* — включает один или больше параметров, взятых в кавычки;
- *formula* — это формула (см. раздел 8.1.3) без знака равенства (=) в начале.

Для стилей данных применимы следующие условия:

- `value() `*op n*, где *op* — оператор отношения, *n* — число;
- для логических стилей значениями условий могут быть `true` и `false`.

Условия, применяемые к различным типам стилей, могут различаться.

```
<define name="style-map-attlist" combine="interleave">
    <attribute name="style:condition">
        <ref name="string"/>
    </attribute>
</define>
```

Применяемый стиль

Атрибут `style:apply-style-name` определяет, какой стиль будет применяться, если условие из атрибута `style:condition` имеет значение `true`. Если стиль, к которому происходит обращение, не определен или является автоматическим, то возникает ошибка.

```
10947 <define name="style-map-attlist" combine="interleave">
10948     <attribute name="style:apply-style-name">
10949         <ref name="styleNameRef"/>
10950     </attribute>
10951 </define>
```

Адрес базовой ячейки

Для стилей ячеек таблиц атрибут `style:base-cell-address` определяет базовую ячейку для относительных адресов в формулах. Этот атрибут применяется только к тем стилям ячеек, где условие содержит формулу. Значение этого атрибута должно быть абсолютным адресом ячейки с именем таблицы.

```
10952 <define name="style-map-attlist" combine="interleave">
10953     <optional>
10954         <attribute name="style:base-cell-address">
10955             <ref name="cellAddress"/>
10956         </attribute>
10957     </optional>
10958 </define>
```

Пример — Соотнесение стилей:

```
<style:style style:name="Text body" style:family="paragraph"
             style:parent-style-name="Standard"
             style:next-style-name="Text body">
    <style:paragraph-properties fo:margin-top="0cm"
                                fo:margin-bottom=".21cm"/>
    <style:map style:condition="footnote"
               style:apply-style-name="footnote"/>
    <style:map style:condition="heading(1)"
               style:apply-style-name="Heading 1"/>
    <style:map style:condition="heading(2)"
               style:apply-style-name="Heading 2"/>
</style:style>
```

14.2 Стили по умолчанию

Стиль по умолчанию устанавливает свойства форматирования по умолчанию для обязательного семейства стилей. Эти значения по умолчанию применяются, если свойство форматирования не определено ни автоматическим,

ни общим стилем. Стили по умолчанию существуют для всех семейств стилей, представленных в элементе `<style:style>`, определенном в разделе 14.1.

Стили по умолчанию представляются элементом `<style:default-style>`. Единственный атрибут, поддерживаемый этим элементом — `style:family`. Его значение соответствует одноименному атрибуту элемента `<style:style>`, и те же свойства дочерних элементов поддерживаются им в зависимости от стиля семейства.

```
<define name="style-default-style">
    <element name="style:default-style">
        <ref name="style-style-content"/>
    </element>
</define>
```

14.3 Разметка страницы

Элемент `<style:page-layout>` определяет физические свойства страницы. Этот элемент содержит элемент `<style:page-layout-properties>`, который определяет свойства форматирования страницы, и два дополнительных элемента, определяющих свойства нижних и верхних колонтитулов.

```
<define name="style-page-layout">
    <element name="style:page-layout">
        <ref name="style-page-layout-attlist"/>
        <optional>
            <ref name="style-page-layout-properties"/>
        </optional>
        <optional>
            <ref name="style-header-style"/>
        </optional>
        <optional>
            <ref name="style-footer-style"/>
        </optional>
    </element>
</define>
```

С элементом `<style:page-layout>` могут быть связаны следующие атрибуты:

- имя;
- использование страницы.

Имя

Атрибут `style:name` определяет имя разметки страницы.

```
<define name="style-page-layout-attlist" combine="interleave">
    <attribute name="style:name">
        <ref name="styleName"/>
```

```
    </attribute>
</define>
```

Использование страницы

Атрибут `style:page-usage` определяет тип страницы, которую должен создать мастер страницы.

```
<define name="style-page-layout-attlist" combine="interleave">
    <optional>
        <attribute name="style:page-usage" a:defaultValue="all">
            <choice>
                <value>all</value>
                <value>left</value>
                <value>right</value>
                <value>mirrored</value>
            </choice>
        </attribute>
    </optional>
</define>
```

14.3.1 Стили колонтитулов

Элементы стиля верхнего и нижнего колонтитула `<style:header-style>` и `<style:footer-style>` определяют свойства форматирования колонтитулов на странице. Эти элементы должны содержаться внутри элемента разметки страницы. Содержимое элемента `style:header-footer-properties>` определяет свойства форматирования верхнего или нижнего колонтитула.

```
<define name="style-header-style">
    <element name="style:header-style">
        <optional>
            <ref name="style-header-footer-properties"/>
        </optional>
    </element>
</define>
<define name="style-footer-style">
    <element name="style:footer-style">
        <optional>
            <ref name="style-header-footer-properties"/>
        </optional>
    </element>
</define>
```

14.4 Мастер-страницы

В текстовых документах и электронных таблицах элемент `<style:master-page>` включает содержимое верхних и нижних колонтитулов. В таких приложениях последовательность страниц генерируется путем использования отдельной мастер-страницы или набора мастер-страниц.

Для графических документов и презентаций элемент `<style:master-page>` используется для определения мастер-страниц как общего предка для **графических страниц**. Каждая графическая страница здесь непосредственно связана с одной мастер-страницей, указанной в атрибуте `draw:master-page-name` стиля графических страниц.

Мастер-страницы содержатся в элементе `<office:master-styles>` (см. также раздел 2.8).

Все документы должны содержать по крайней мере один элемент мастер-страницы.

```
11009 <define name="style-master-page">
11010     <element name="style:master-page">
11011         <ref name="style-master-page-attlist"/>
11012         <optional>
11013             <ref name="style-header"/>
11014             <optional>
11015                 <ref name="style-header-left"/>
11016             </optional>
11017         </optional>
11018         <optional>
11019             <ref name="style-footer"/>
11020             <optional>
11021                 <ref name="style-footer-left"/>
11022             </optional>
11023         </optional>
11024         <optional>
11025             <ref name="office-forms"/>
11026         </optional>
11027         <zeroOrMore>
11028             <ref name="style-style"/>
11029         </zeroOrMore>
11030         <zeroOrMore>
11031             <ref name="shape"/>
11032         </zeroOrMore>
11033         <optional>
11034             <ref name="presentation-notes"/>
11035         </optional>
11036     </element>
11037 </define>
```

С элементом `<style:master-page>` могут быть связаны следующие атрибуты:

- имя страницы;
- отображаемое имя;
- разметка страницы;
- стиль страницы;
- имя следующего стиля.

В элемент `<style:master-page>` могут быть включены следующие элементы:

- нижние и верхние колонтитулы;
- формы;
- стили;
- фигуры;
- заметки к презентации.

Имя страницы

Атрибут `style:name` определяет имя мастер-страницы. Ссылаться на мастер-страницу можно через имя страницы. Требуется присутствие данного атрибута и уникальность имени.

```
<define name="style-master-page-attlist" combine="interleave">
    <attribute name="style:name">
        <ref name="styleName"/>
    </attribute>
</define>
```

Отображаемое имя

Атрибут `style:display-name` определяет имя мастер-страницы, которое должно появиться в пользовательском интерфейсе. В отличие от имени стиля как такового, это имя может содержать произвольные символы. Если этот атрибут не указан, отображаемое имя совпадает с именем стиля.

```
<define name="style-master-page-attlist" combine="interleave">
    <optional>
        <attribute name="style:display-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Разметка страницы

Атрибут `style:page-layout-name` определяет разметку страницы, которая включает размеры, обрамление и ориентацию мастер-страницы (см. раздел 14.3 для получения дополнительной информации о разметке страницы).

```
<define name="style-master-page-attlist" combine="interleave">
    <attribute name="style:page-layout-name">
        <ref name="styleNameRef"/>
    </attribute>
</define>
```

Стиль страницы

В графических приложениях дополнительные атрибуты графических страниц могут быть присвоены графической странице с помощью атрибута draw:style-name. Данный атрибут является необязательным. Фиксированным семейством стилей страниц является drawing-page. Это используется для определения дополнительного исходного наполнения.

```
11055 <define name="style-master-page-attlist" combine="interleave">
11056     <optional>
11057         <attribute name="draw:style-name">
11058             <ref name="styleNameRef"/>
11059         </attribute>
11060     </optional>
11061 </define>
```

Имя следующего стиля

Для текстовых документов и электронных таблиц атрибут style:next-style-name устанавливает мастер-страницу, использующуюся для следующей страницы, если текущая страница полностью заполнена. Данный атрибут является необязательным. Если имя следующего стиля не указано, текущая мастер-страница используется для следующей страницы. Значением этого атрибута должно быть имя другого элемента style:master-page.

```
11062 <define name="style-master-page-attlist" combine="interleave">
11063     <optional>
11064         <attribute name="style:next-style-name">
11065             <ref name="styleNameRef"/>
11066         </attribute>
11067     </optional>
11068 </define>
```

14.4.1 Колонтитулы

Элементы верхнего и нижнего колонтитула определяют содержимое колонтитулов. Они содержатся внутри элемента мастер-страницы. Элементы <style:header> и <style:footer> включают содержимое верхнего и нижнего колонтитула. Два дополнительных элемента <style:header-left> и <style:footer-left> могут быть использованы для указания различий в оформлении левых страниц, если необходимо. Если последние два элемента отсутствуют, содержимое колонтитулов на левой и правой странице будет одинаковым.

Если атрибут style:page-usage, связанный с разметкой страницы, имеет значение all или mirrored и отсутствуют элементы <style:header-left> и <style:footer-left>, то содержимое колонтитулов одинаково на левой и правой странице.

Если атрибут style:page-usage имеет значение left или right, то элементы <style:header-left> и <style:footer-left> игнорируются.

Содержимое колонтитулов может быть трех видов:

- Стандартное текстовое содержимое, например абзацы, таблицы или списки. Такие колонтитулы обычно поддерживаются текстовыми документами.
- Последовательность любых элементов из перечисленных: <style:region-left>, <style:region-center>, <style:region-right>. Такие колонтитулы обычно поддерживаются в электронных таблицах.
- Пустым, что отключает отображение всех колонтитулов. Невозможно отключить отображение колонтитулов только на левых страницах.

```
11069 <define name="style-header">
11070     <element name="style:header">
11071         <ref name="common-style-header-footer-attlist"/>
11072         <ref name="header-footer-content"/>
11073     </element>
11074 </define>
11075 <define name="style-footer">
11076     <element name="style:footer">
11077         <ref name="common-style-header-footer-attlist"/>
11078         <ref name="header-footer-content"/>
11079     </element>
11080 </define>
11081 <define name="style-header-left">
11082     <element name="style:header-left">
11083         <ref name="common-style-header-footer-attlist"/>
11084         <ref name="header-footer-content"/>
11085     </element>
11086 </define>
11087 <define name="style-footer-left">
11088     <element name="style:footer-left">
11089         <ref name="common-style-header-footer-attlist"/>
11090         <ref name="header-footer-content"/>
11091     </element>
11092 </define>
11093 <define name="header-footer-content">
11094     <choice>
11095         <group>
11096             <ref name="text-decls"/>
11097             <zeroOrMore>
11098                 <choice>
11099                     <ref name="text-h"/>
11100                     <ref name="text-p"/>
```

```
                    <ref name="text-list"/>
                    <ref name="table-table"/>
                    <ref name="text-section"/>
                    <ref name="text-table-of-content"/>
                    <ref name="text-illustration-index"/>
                    <ref name="text-table-index"/>
                    <ref name="text-object-index"/>
                    <ref name="text-user-index"/>
                    <ref name="text-alphabetical-index"/>
                    <ref name="text-bibliography"/>
                    <ref name="text-index-title"/>
                    <ref name="change-marks"/>
                </choice>
            </zeroOrMore>
        </group>
        <group>
            <optional>
                <ref name="style-region-left"/>
            </optional>
            <optional>
                <ref name="style-region-center"/>
            </optional>
            <optional>
                <ref name="style-region-right"/>
            </optional>
        </group>
    </choice>
</define>
```

Отображение

Атрибут `style:display` определяет, отображаются ли колонтитулы.

```
<define name="common-style-header-footer-attlist" combine="interleave">
    <optional>
        <attribute name="style:display" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Области

Элементы областей `<style:region-left>` , `<style:region-center>` и `<style:region-right>` определяют три области колонтитула, которые выравниваются по левому краю, по центру или по правому краю. Каждый из этих участков может содержать последовательность абзацев.

```
<define name="style-region-left">
    <element name="style:region-left">
        <ref name="region-content"/>
    </element>
</define>
<define name="style-region-center">
    <element name="style:region-center">
        <ref name="region-content"/>
    </element>
```

```
11145 </define>
11146 <define name="style-region-right">
11147     <element name="style:region-right">
11148         <ref name="region-content"/>
11149     </element>
11150 </define>
11151
11152 <define name="region-content">
11153     <zeroOrMore>
11154         <ref name="text-p"/>
11155     </zeroOrMore>
11156 </define>
```

14.4.2 Заметки к презентации

Элемент `<presentation:notes>` обычно поддерживается только приложениями презентаций, в которых каждая мастер-страница и графическая страница могут иметь дополнительную страницу заметок.

Страница заметок к презентации содержит следующее.

- Уменьшенное изображение графической страницы.
- Дополнительные графические формы из содержащихся в элементе `<presentation:notes>`. Несмотря на то, что элемент `<presentation:notes>` может содержать любой тип формы, приложения презентаций поддерживают только текстовые блоки (т. е. `<draw:text-box>`, содержащийся в `<draw:frame>`).

```
11157 <define name="presentation-notes">
11158     <element name="presentation:notes">
11159         <ref name="common-presentation-header-footer-attlist"/>
11160         <ref name="presentation-notes-attlist"/>
11161         <zeroOrMore>
11162             <ref name="shape"/>
11163         </zeroOrMore>
11164     </element>
11165 </define>
```

Разметка страницы

Атрибут `style:page-layout-name` определяет разметку страницы, которая включает размеры, границы и расположение страниц заметок (см. раздел 14.3 для получения дополнительной информации о разметке страниц).

```
11166 <define name="presentation-notes-attlist" combine="interleave">
11167     <optional>
11168         <attribute name="style:page-layout-name">
11169             <ref name="styleNameRef"/>
11170         </attribute>
11171     </optional>
11172 </define>
```

Стиль страницы

Атрибут draw:style-name присваивает странице с заметками дополнительные форматирующие атрибуты путем назначения стиля графической страницы. Этот атрибут не является обязательным. Фиксированное семейство для стиля страниц — drawing-page.

```
11173 <define name="presentation-notes-attlist" combine="interleave">
11174     <optional>
11175         <attribute name="draw:style-name">
11176             <ref name="styleNameRef"/>
11177         </attribute>
11178     </optional>
11179 </define>
```

Объявление верхнего колонтитула

Атрибут presentation:use-header-name определяет имя объявления поля верхнего колонтитула (см. раздел 9.11.2), который используется для всех полей верхнего колонтитула (см. раздел 9.10.1), представленных на странице с заметками (см. также раздел 9.1.4).

Объявление нижнего колонтитула

Атрибут presentation:use-footer-name определяет имя объявления поля нижнего колонтитула (см. раздел 9.11.2), который используется для всех полей нижнего колонтитула (см. раздел 9.10.1), представленных на странице примечаний (см. также раздел 9.1.4).

Объявление даты и времени

Атрибут presentation:use-date-time-name определяет имя объявления поля дата-время (см. раздел 9.11.4), которое используется во всех полях дата-время (см. раздел 9.10.3), представленных на странице заметок (см. также раздел 9.1.4).

Пример — Мастер-страница, содержащая заметки к презентации:

```
<office:master-styles>
    ...
    <style:master-page style:name="home" style:page-layout="default">
        <style:style style:name="title" style:family="presentation">
            <style:text-properties fo:font-style="italic"/>
        </style:style>
        <style:style style:name="subtitle" style:family="presentation"
```

```
                    style:parent-style-name="title">
            <style:text-properties style:text-outline="true"/>
        </style:style>
        <draw:rectangle .../>
            <presentation:notes>
                <draw:text ...>this is a note</draw:text>
            </presentation:notes>
    </style:master-page>
    ...
</office:master-styles>
```

14.5 Шаблоны таблиц

Шаблон таблицы — это набор форматирующих свойств, таких как обрамление, цвет фона и свойств текста, которые могут применяться во время создания таблицы. В отличие от других стилей, не таблица ссылается на стиль, а при создании таблицы набор стилей ее ячеек берется из шаблона таблицы. Чтобы изменить форматирующие свойства таблицы, нужно изменить непосредственно стили ячеек и другие стили. Шаблоны таблиц содержатся в элементе `<style:master-styles>`.

```
11180 <define name="table-table-template">
11181     <element name="table:table-template">
11182         <ref name="table-table-template-attlist"/>
11183         <optional>
11184             <ref name="table-first-row"/>
11185         </optional>
11186         <optional>
11187             <ref name="table-last-row"/>
11188         </optional>
11189         <optional>
11190             <ref name="table-first-column"/>
11191         </optional>
11192         <optional>
11193             <ref name="table-last-column"/>
11194         </optional>
11195         <choice>
11196             <ref name="table-body"/>
11197             <group>
11198                 <ref name="table-even-rows"/>
11199                 <ref name="table-odd-rows"/>
11200             </group>
11201             <group>
11202                 <ref name="table-even-columns"/>
11203                 <ref name="table-odd-columns"/>
11204             </group>
11205         </choice>
11206     </element>
11207 </define>
```

Имя стиля

Атрибут `table:name` определяет имя шаблона таблицы.

```
11208 <define name="table-table-template-attlist" combine="interleave">
```

```
    <attribute name="text:name">
        <ref name="string"/>
    </attribute>
</define>
```

Стили углов

Атрибуты table:first-row-start-column, table:first-row-end-column, table:last-row-start-column и table:last-row-end-column определяют, какой стиль должны получить ячейки в четырех углах таблицы — от столбца или от строки, в которых они находятся. Возможными значениями этих атрибутов являются row и column.

```
<define name="table-table-template-attlist" combine="interleave">
    <attribute name="text:first-row-start-column">
        <ref name="rowOrCol"/>
    </attribute>
</define>

<define name="table-table-template-attlist" combine="interleave">
    <attribute name="text:first-row-end-column">
        <ref name="rowOrCol"/>
    </attribute>
</define>

<define name="table-table-template-attlist" combine="interleave">
    <attribute name="text:last-row-start-column">
        <ref name="rowOrCol"/>
    </attribute>
</define>

<define name="table-table-template-attlist" combine="interleave">
    <attribute name="text:last-row-end-column">
        <ref name="rowOrCol"/>
    </attribute>
</define>

<define name="rowOrCol">
    <choice>
        <value>row</value>
        <value>column</value>
    </choice>
</define>
```

14.5.1 Стили строк и столбцов

Элементы <table:first-row> и <table:last-row> определяют стили ячеек, которые должны применяться к первой и последней строкам таблицы. Они содержат атрибут table:style-name, который ссылается на эти стили. Элементы <table:first-col> и <table:last-col> делают то же самое для первого и последнего столбцов. Стили остальных ячеек могут определяться

элементом <table:body> или одной из пар элементов — <table:even-rows>/<table:odd-rows> или <table:even-columns>/<table:odd-columns>, если различные стили должны применяться к четным и нечетным строкам или столбцам.

```
<define name="table-first-row">
    <element name="table:first-row">
        <ref name="common-table-template-attlist"/>
        <empty/>
    </element>
</define>

<define name="table-last-row">
    <element name="table:last-row">
        <ref name="common-table-template-attlist"/>
        <empty/>
    </element>
</define>

<define name="table-first-column">
    <element name="table:first-column">
        <ref name="common-table-template-attlist"/>
        <empty/>
    </element>
</define>

<define name="table-last-column">
    <element name="table:last-column">
        <ref name="common-table-template-attlist"/>
        <empty/>
    </element>
</define>

<define name="table-body">
    <element name="table:body">
        <ref name="common-table-template-attlist"/>
        <empty/>
    </element>
</define>

<define name="table-even-rows">
    <element name="table:even-rows">
        <ref name="common-table-template-attlist"/>
        <empty/>
    </element>
</define>

<define name="table-odd-rows">
    <element name="table:odd-rows">
        <ref name="common-table-template-attlist"/>
        <empty/>
    </element>
</define>

<define name="table-even-columns">
    <element name="table:even-columns">
        <ref name="common-table-template-attlist"/>
        <empty/>
    </element>
</define>
```

```
<define name="table-odd-columns">
    <element name="table:odd-columns">
        <ref name="common-table-template-attlist"/>
        <empty/>
    </element>
</define>

<define name="common-table-template-attlist" combine="interleave">
    <attribute name="text:style-name">
        <ref name="styleNameRef"/>
    </attribute>
</define>
```

14.6 Объявление начертания шрифта

Объявление начертания шрифта в OpenDocument в точности соответствует `@font-face` описанию шрифта в [CSS2] (см. § 15.3.1) и `<font-face>` элементу в [SVG] (см. § 20.8.3), но имеет два следующих расширения.

- Объявления начертания шрифта в OpenDocument необязательно может иметь уникальное название. Это название может быть использовано в стилях (т. е. как атрибут элемента `<style:text-properties>`) как значение атрибута `style:font-name` для непосредственного выбора объявления начертания шрифта. Если на объявление начертания шрифта ссылаются таким образом, то шаги описанного в § 15.5 [CSS2] алгоритма поиска совпадения для выбора объявления шрифта, основанного на дескрипторах font-family, font-style, font-variant, font-weight и font-size, не будут выполнены, а будет использоваться объявление начертания шрифта, заданное непосредственно.
- Существует несколько дополнительных атрибутов дескриптора шрифта. Их назначение раскрыто ниже.

С исключениями, приведенными ранее, соответствующие приложения должны реализовывать алгоритм CSS2 поиска совпадения шрифта, как описано в § 15.5 [CSS2], но они могут также реализовывать его варианты. Главным образом дана возможность реализовывать алгоритм поиска совпадения шрифта, основанного только на объявлении начертания шрифта, то есть поиск совпадения шрифта не применяется к каждому символу независимо, а только однажды для каждого объявления вида шрифта. Это полезно для приложений, связанных с

редактированием, где поиск совпадения, основанный на символах, может быть слишком ресурсоемким.

```
<define name="style-font-face">
    <element name="style:font-face">
        <ref name="style-font-face-attlist"/>
        <optional>
            <ref name="svg-font-face-src"/>
        </optional>
        <optional>
            <ref name="svg-definition-src"/>
        </optional>
    </element>
</define>
```

14.6.1 Дескрипторы шрифта CSS2/SVG

Объявления начертания шрифта поддерживают атрибуты и элементы дескриптора шрифта, описанные в § 20.8.3 [SVG].

```
<define name="style-font-face-attlist" combine="interleave">
    <optional>
        <attribute name="svg:font-family">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="svg:font-style">
            <ref name="fontStyle"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="svg:font-variant">
            <ref name="fontVariant"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="svg:font-weight">
            <ref name="fontWeight"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="svg:font-stretch">
            <choice>
                <value>normal</value>
                <value>ultra-condensed</value>
                <value>extra-condensed</value>
                <value>condensed</value>
                <value>semi-condensed</value>
                <value>semi-expanded</value>
                <value>expanded</value>
                <value>extra-expanded</value>
                <value>ultra-expanded</value>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="svg:font-size">
            <ref name="positiveLength"/>
        </attribute>
```

```
11362     </optional>
11363     <optional>
11364         <attribute name="svg:unicode-range"/>
11365     </optional>
11366     <optional>
11367         <attribute name="svg:units-per-em">
11368             <ref name="integer"/>
11369         </attribute>
11370     </optional>
11371     <optional>
11372         <attribute name="svg:panose-1"/>
11373     </optional>
11374     <optional>
11375         <attribute name="svg:stemv">
11376             <ref name="integer"/>
11377         </attribute>
11378     </optional>
11379     <optional>
11380         <attribute name="svg:stemh">
11381             <ref name="integer"/>
11382         </attribute>
11383     </optional>
11384     <optional>
11385         <attribute name="svg:slope">
11386             <ref name="integer"/>
11387         </attribute>
11388         </optional>
11389     <optional>
11390         <attribute name="svg:cap-height">
11391             <ref name="integer"/>
11392         </attribute>
11393     </optional>
11394     <optional>
11395         <attribute name="svg:x-height">
11396             <ref name="integer"/>
11397         </attribute>
11398     </optional>
11399     <optional>
11400         <attribute name="svg:accent-height">
11401             <ref name="integer"/>
11402         </attribute>
11403     </optional>
11404     <optional>
11405         <attribute name="svg:ascent">
11406             <ref name="integer"/>
11407         </attribute>
11408     </optional>
11409     <optional>
11410         <attribute name="svg:descent">
11411             <ref name="integer"/>
11412         </attribute>
11413     </optional>
11414     <optional>
11415         <attribute name="svg:widths"/>
11416     </optional>
11417     <optional>
11418         <attribute name="svg:bbox"/>
11419     </optional>
11420     <optional>
11421         <attribute name="svg:ideographic">
11422             <ref name="integer"/>
11423         </attribute>
11424     </optional>
```

```
11425     <optional>
11426         <attribute name="svg:alphabetic">
11427             <ref name="integer"/>
11428         </attribute>
11429     </optional>
11430     <optional>
11431         <attribute name="svg:mathematical">
11432             <ref name="integer"/>
11433         </attribute>
11434     </optional>
11435     <optional>
11436         <attribute name="svg:hanging">
11437             <ref name="integer"/>
11438         </attribute>
11439     </optional>
11440     <optional>
11441         <attribute name="svg:v-ideographic">
11442             <ref name="integer"/>
11443         </attribute>
11444     </optional>
11445     <optional>
11446         <attribute name="svg:v-alphabetic">
11447             <ref name="integer"/>
11448         </attribute>
11449     </optional>
11450     <optional>
11451         <attribute name="svg:v-mathematical">
11452             <ref name="integer"/>
11453         </attribute>
11454     </optional>
11455     <optional>
11456         <attribute name="svg:v-hanging">
11457             <ref name="integer"/>
11458         </attribute>
11459     </optional>
11460     <optional>
11461         <attribute name="svg:underline-position">
11462             <ref name="integer"/>
11463         </attribute>
11464     </optional>
11465     <optional>
11466         <attribute name="svg:underline-thickness">
11467             <ref name="integer"/>
11468         </attribute>
11469     </optional>
11470     <optional>
11471         <attribute name="svg:strikethrough-position">
11472             <ref name="integer"/>
11473         </attribute>
11474     </optional>
11475     <optional>
11476         <attribute name="svg:strikethrough-thickness">
11477             <ref name="integer"/>
11478         </attribute>
11479     </optional>
11480     <optional>
11481         <attribute name="svg:overline-position">
11482             <ref name="integer"/>
11483         </attribute>
11484         </optional>
11485     <optional>
11486         <attribute name="svg:overline-thickness">
11487             <ref name="integer"/>
```

```
11488         </attribute>
11489     </optional>
11490 </define>
11491
11492 <define name="svg-font-face-src">
11493     <element name="svg:font-face-src">
11494         <oneOrMore>
11495             <choice>
11496                 <ref name="svg-font-face-uri"/>
11497                 <ref name="svg-font-face-name"/>
11498             </choice>
11499         </oneOrMore>
11500     </element>
11501 </define>
11502
11503 <define name="svg-font-face-uri">
11504     <element name="svg:font-face-uri">
11505         <ref name="common-svg-font-face-xlink-attlist"/>
11506         <zeroOrMore>
11507             <ref name="svg-font-face-format"/>
11508         </zeroOrMore>
11509     </element>
11510 </define>
11511
11512 <define name="svg-font-face-format">
11513     <element name="svg:font-face-format">
11514         <optional>
11515             <attribute name="svg:string"/>
11516         </optional>
11517         <empty/>
11518     </element>
11519 </define>
11520 <define name="svg-font-face-name">
11521     <element name="svg:font-face-name">
11522         <optional>
11523             <attribute name="name"/>
11524         </optional>
11525         <empty/>
11526     </element>
11527 </define>
11528
11529 <define name="svg-definition-src">
11530     <element name="svg:definition-src">
11531         <ref name="common-svg-font-face-xlink-attlist"/>
11532     <empty/>
11533     </element>
11534 </define>
11535
11536 <define name="common-svg-font-face-xlink-attlist" combine="interleave">
11537     <attribute name="xlink:href">
11538         <ref name="anyURI"/>
11539     </attribute>
11540     <optional>
11541         <attribute name="xlink:type" a:defaultValue="simple">
11542             <value>simple</value>
11543         </attribute>
11544     </optional>
11545     <optional>
11546         <attribute name="xlink:actuate" a:defaultValue="onRequest">
11547             <value>onRequest</value>
11548         </attribute>
11549     </optional>
11550 </define>
```

14.6.2 Имя

Атрибут `style:name` определяет уникальное имя объявления шрифта. Это имя может быть использовано в стилях (т. е. как атрибут элемента `<style:text-properties>`) как значение атрибута `style:font-name` для непосредственного выбора объявления начертания шрифта

```
<define name="style-font-face-attlist" combine="interleave">
    <attribute name="style:name">
        <ref name="string"/>
    </attribute>
</define>
```

14.6.3 Начертание

Атрибуты `style:font-adornments` определяют начертание, такое как полужирный или курсив, которое может использоваться для классификации шрифта в дополнение к названию семейства.

```
<define name="style-font-face-attlist" combine="interleave">
    <optional>
        <attribute name="style:font-adornments">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

14.6.4 Общее семейство шрифта

Атрибут `style:font-family-generic` определяет имя общего семейства шрифта (см. раздел 15.4.15 для получения подробной информации).

```
<define name="style-font-face-attlist" combine="interleave">
    <optional>
        <attribute name="style:font-family-generic">
            <ref name="fontFamilyGeneric"/>
        </attribute>
    </optional>
</define>
```

14.6.5 Шаг шрифта

Атрибут `style:font-pitch` определяет какой будет ширина шрифта: фиксированной или варьируемой (см. раздел 15.4.17 для получения подробной информации).

```
<define name="style-font-face-attlist" combine="interleave">
    <optional>
        <attribute name="style:font-pitch">
            <ref name="fontPitch"/>
```

```
11574         </attribute>
11575     </optional>
11576 </define>
11577
```

14.6.6 Набор символов шрифта

Атрибут `style:font-charset` определяет набор символов шрифта (см. раздел 15.4.18 для получения подробной информации).

```
11578 <define name="style-font-face-attlist" combine="interleave">
11579     <optional>
11580         <attribute name="style:font-charset">
11581             <ref name="textEncoding"/>
11582         </attribute>
11583     </optional>
11584 </define>
```

14.7 Стили данных

Стили данных описывают, как отображать различные типы данных, например числа или даты. Элементы и атрибуты, которые используются для представления стилей данных, содержатся в пространстве имен urn:oasis:names:tc:opendocument:xmlns:datastyle:1.0. Префикс `number` обозначает пространство имен стилей данных.

Эти секции описывают представление в OpenDocument следующих стилей данных:

- числовой стиль;
- валютный стиль;
- процентный стиль;
- стиль даты;
- логический стиль;
- текстовый стиль.

14.7.1 Числовой стиль

Элемент `<number:number-style>` описывает стиль десятичных чисел.

Этот элемент может содержать *один* из следующих элементов:

- `<number:number>`
- `<number:scientific-number>`
- `<number:fraction>`

Эти элементы описывают формат отображения числа. Элементы могут стоять до или после элементов `<number:text>`, содержащих любой дополнительный текст, который должен быть отображен до или после числа.

Помимо этого, данный элемент может содержать элемент `<style:text-properties>` и элемент `<style:map>`.

```
<define name="number-number-style">
    <element name="number:number-style">
        <ref name="common-data-style-attlist"/>
        <optional>
            <ref name="style-text-properties"/>
        </optional>
        <optional>
            <ref name="number-text"/>
        </optional>
        <optional>
            <ref name="any-number"/>
            <optional>
                <ref name="number-text"/>
            </optional>
        </optional>
        <zeroOrMore>
            <ref name="style-map"/>
        </zeroOrMore>
    </element>
</define>

<define name="any-number">
    <choice>
        <ref name="number-number"/>
        <ref name="number-scientific-number"/>
        <ref name="number-fraction"/>
    </choice>
</define>
```

Для получения информации об атрибутах, которые могут быть связаны с элементами числового стиля, см. раздел 14.7.9.

В элементе `<number:number-style>` могут содержаться следующие элементы:

- число;
- число в научном формате;
- дробь.

Число

Элемент `<number:number>` определяет свойства отображения для десятичных чисел.

Этот элемент содержится в элементе <number:number-style>. Элемент <number:number> может содержать множество повторяющихся элементов <number:embedded-text>.

Атрибуты number:decimal-replacement и number:display-factor могут быть использованы с этим элементом. Для получения подробной информации о дополнительных атрибутах, которые могут быть связаны с элементом <number:number>, см. раздел 14.7.11.

```
<define name="number-number">
    <element name="number:number">
        <ref name="number-number-attlist"/>
        <ref name="common-decimal-places-attlist"/>
        <ref name="common-number-attlist"/>
        <zeroOrMore>
            <ref name="number-embedded-text"/>
        </zeroOrMore>
    </element>
</define>
```

Замещение в десятичных дробях

Если числовой стиль определяет, что десятичная точка используется, но отображаемое число является целым, тогда замещающий текст может быть отображен вместо десятичной точки. Атрибут number:decimal-replacement определяет замещающий текст.

Некоторые приложения могут поддерживать только такой замещающий текст, который состоит из того же числа символов «-», как и число разрядов.

```
<define name="number-number-attlist" combine="interleave">
    <optional>
        <attribute name="number:decimal-replacement"/>
    </optional>
</define>
```

Множитель отображения

Атрибут number:display-factor определяет множитель, показывающий, во сколько раз будет увеличиваться (уменьшаться) каждое число перед отображением. Например, с помощью множителя 1000 числа будут отображаться в тысячах.

Некоторые приложения могут поддерживать множители отображения только равные 1000 и ее целочисленным степеням: 1, 1000, 1000000, 1000000000 и т. д.

```
<define name="number-number-attlist" combine="interleave">
    <optional>
        <attribute name="number:display-factor" a:defaultValue="1">
            <ref name="double"/>
```

```
        </attribute>
    </optional>
</define>
```

Внедренный текст

Элемент `<number:embedded-text>` определяет текст, который отображается на одной определенной позиции внутри числа. Этот элемент отличается от группирующего разделителя, который появляется несколько раз внутри числа.

Этот элемент содержится в элементе `<number:number>`. Элемент `<number:number>` может содержать множество повторяющихся элементов `<number:embedded-text>` для изображения текста в различных позициях числа.

```
<define name="number-embedded-text">
    <element name="number:embedded-text">
        <ref name="number-embedded-text-attlist"/>
        <text/>
    </element>
</define>
```

Атрибут `number:position` устанавливает позицию, в которой будет показан текст.

Атрибут позиции

Позиция отсчитывается справа налево, от предшествующих десятичной точке, если она есть, или в противном случае от конца числа. Например, позиция 1 в числе показывает, что текст будет вставлен до последней цифры, позиция 2 показывает, что текст будет вставлен до предпоследней цифры, и т. д.

```
<define name="number-embedded-text-attlist" combine="interleave">
    <attribute name="number:position">
        <ref name="integer"/>
    </attribute>
</define>
```

Число в научном формате

Элемент `<number:scientific-number>` определяет свойства отображения для числа, которое в соответствии со стилем должно отображаться в научном формате.

Этот элемент содержится в элементе `<number:number-style>`.

Атрибут `number:min-exponent-digits` может быть использован с этим элементом. Для получения информации о дополнительных атрибутах, которые

могут быть связаны с элементом `<number:scientific-number>`, см. раздел 14.7.11.

```
<define name="number-scientific-number">
    <element name="number:scientific-number">
        <ref name="number-scientific-number-attlist"/>
        <ref name="common-decimal-places-attlist"/>
        <ref name="common-number-attlist"/>
        <empty/>
    </element>
</define>
```

Минимальное число цифр в экспоненте

Атрибут `number:min-exponent-digits` определяет минимальное число цифр, используемых для отображения экспоненты. Этот атрибут поддерживается для элемента `<number:scientific-number>`.

```
<define name="number-scientific-number-attlist" combine="interleave">
    <optional>
        <attribute name="number:min-exponent-digits">
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

Дробь

Элемент `<number:fraction>` определяет свойства отображения для числового стиля, который должен отображаться как дробь.

Этот элемент содержится в элементе `<number:number-style>`.

Атрибуты `number:min-numerator-digits` и `number:min-denominator-digits` могут быть использованы с этим элементом. Для получения информации о атрибутах, которые могут быть связаны с элементами `<number:fraction>`, см. раздел 14.7.11.

```
<define name="number-fraction">
    <element name="number:fraction">
        <ref name="number-fraction-attlist"/>
        <ref name="common-number-attlist"/>
        <empty/>
    </element>
</define>
```

Минимальное количество цифр в числителе

Атрибут `number:min-numerator-digits` определяет минимальное число цифр, которые используются для отображения числителя дроби.

```
<define name="number-fraction-attlist" combine="interleave">
    <optional>
```

```
11670         <attribute name="number:min-numerator-digits">
11671             <ref name="integer"/>
11672         </attribute>
11673     </optional>
11674 </define>
```

Минимальное количество цифр в знаменателе

Атрибут `number:min-denominator-digits` определяет минимальное число цифр, которые используются для отображения знаменателя дроби.

```
11675 <define name="number-fraction-attlist" combine="interleave">
11676     <optional>
11677         <attribute name="number:min-denominator-digits">
11678             <ref name="integer"/>
11679         </attribute>
11680     </optional>
11681 </define>
```

Значение знаменателя

Атрибут `number:denominator-value` определяет целое число, которое используется как знаменатель дроби. Если этот атрибут не установлен, то приложение может выбрать некоторое случайное значение знаменателя.

```
11682 <define name="number-fraction-attlist" combine="interleave">
11683     <optional>
11684         <attribute name="number:denominator-value">
11685             <ref name="integer"/>
11686         </attribute>
11687     </optional>
11688 </define>
```

14.7.2 Валютный стиль

Элемент `<number:currency-style>` описывает стиль для валютных значений.

Этот элемент может содержать один элемент `<number:number>` и один элемент `<number:currency-symbol>`. Он также может содержать элемент `<number:text>`, который выводит дополнительный текст, но он не может содержать два этих элемента подряд.

Кроме того, этот элемент может содержать элемент `<style:text-properties>` и элемент `<style:map>`.

```
11689 <define name="number-currency-style">
11690     <element name="number:currency-style">
11691         <ref name="common-data-style-attlist"/>
11692         <ref name="common-auto-reorder-attlist"/>
11693         <optional>
11694             <ref name="style-text-properties"/>
11695         </optional>
```

```
11696        <optional>
11697            <ref name="number-text"/>
11698        </optional>
11699        <optional>
11700            <choice>
11701                <group>
11702                    <ref name="number-and-text"/>
11703                    <optional>
11704                        <ref name="currency-symbol-and-text"/>
11705                    </optional>
11706                </group>
11707                <group>
11708                    <ref name="currency-symbol-and-text"/>
11709                    <optional>
11710                        <ref name="number-and-text"/>
11711                    </optional>
11712                </group>
11713            </choice>
11714        </optional>
11715        <zeroOrMore>
11716            <ref name="style-map"/>
11717        </zeroOrMore>
11718    </element>
11719 </define>
11720
11721 <define name="currency-symbol-and-text">
11722     <ref name="number-currency-symbol"/>
11723     <optional>
11724         <ref name="number-text"/>
11725     </optional>
11726 </define>
11727 <define name="number-and-text">
11728     <ref name="number-number"/>
11729     <optional>
11730         <ref name="number-text"/>
11731     </optional>
11732 </define>
```

Для получения информации об атрибутах, которые могут быть связаны с элементами числовых стилей, см. раздел 14.7.9.

Следующие элементы могут содержаться в элементе `<number:currency-style>`:

- число (см. раздел 14.7.1);
- символ валюты.

Символ валюты

Элемент `<number:currency-symbol>` определяет, отображен ли символ валюты для отображения в валютном стиле.

Содержимое этого элемента — это текст, который отображается как символ валюты. Если элемент пуст или содержит только пробельные символы, тогда

будет отображен символ валюты по умолчанию для валютного стиля или языка и страны.

Этот элемент содержится в элементе <number:currency-style>.

```
<define name="number-currency-symbol">
    <element name="number:currency-symbol">
        <ref name="number-currency-symbol-attlist"/>
        <text/>
    </element>
</define>
```

Атрибуты number:language и number:country могут быть использованы для установки языка и страны символа валюты. Для получения информации об остальных атрибутах, которые могут быть связаны с элементами валютного стиля, см. раздел 14.7.11.

Атрибуты валюты языка и страны

Если символ валюты, содержащийся в валютном стиле, принадлежит отличающимся от самого валютного стиля языку или стране, то атрибуты number:language и number:country могут быть использованы, чтобы установить язык и страну символа валюты.

```
<define name="number-currency-symbol-attlist" combine="interleave">
    <optional>
        <attribute name="number:language">
            <ref name="languageCode"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="number:country">
            <ref name="countryCode"/>
        </attribute>
    </optional>
</define>
```

14.7.3 Процентный стиль

Элемент <number:percentage-style> определяет стиль процентных значений.

Этот элемент может содержать один элемент <number:number>, который описывает формат отображения для процентов. До или поле элемента могут находиться элементы <number:text>, которые содержат любой дополнительный текст, отображаемый до или после процентов. Некоторые приложения требуют, чтобы как минимум один элемент <number:text> существовал, и его текст должен содержать символ «%».

Помимо этого, элемент `<number:percentage-style>` может содержать элемент `<style:text-properties>` и элемент `<style:map>`.

```
<define name="number-percentage-style">
    <element name="number:percentage-style">
        <ref name="common-data-style-attlist"/>
        <optional>
            <ref name="style-text-properties"/>
        </optional>
        <optional>
            <ref name="number-text"/>
        </optional>
        <optional>
            <ref name="number-and-text"/>
        </optional>
        <zeroOrMore>
            <ref name="style-map"/>
        </zeroOrMore>
    </element>
</define>
```

Для получения информации об атрибутах, которые могут быть связаны с элементом процентного стиля, см. раздел 14.7.9.

14.7.4 Стиль даты

Элемент `<number:date-style>` определяет стиль для значений типа дата.

Этот элемент может содержать *один* экземпляр каждого из следующих элементов: `<number:day>`, `<number:month>`, `<number:year>`, `<number:era>`, `<number:day-of-week>`, `<number:week-of-year>`, `<number:quarter>`, `<number:hours>`, `<number:minutes>`, `<number:seconds>` и `<number:am-pm>`.

Элемент `<number:date-style>` может также содержать элементы `<number:text>`, которые отображают дополнительный текст, но он не может содержать два этих элемента подряд. Кроме того, он может содержать элементы `<style:text-properties>` и `<style:map>`.

```
<define name="number-date-style">
    <element name="number:date-style">
        <ref name="common-data-style-attlist"/>
        <ref name="common-auto-reorder-attlist"/>
        <ref name="common-format-source-attlist"/>
        <optional>
            <ref name="style-text-properties"/>
        </optional>
        <!-- This DTD does not reflect the fact that some elements must not -->
        <!-- occur more than once. -->
        <optional>
            <ref name="number-text"/>
        </optional>
        <oneOrMore>
            <ref name="any-date"/>
```

```
11783            <optional>
11784                <ref name="number-text"/>
11785            </optional>
11786        </oneOrMore>
11787        <zeroOrMore>
11788            <ref name="style-map"/>
11789        </zeroOrMore>
11790    </element>
11791 </define>
11792
11793 <define name="any-date">
11794    <choice>
11795        <ref name="number-day"/>
11796        <ref name="number-month"/>
11797        <ref name="number-year"/>
11798        <ref name="number-era"/>
11799        <ref name="number-day-of-week"/>
11800        <ref name="number-week-of-year"/>
11801        <ref name="number-quarter"/>
11802        <ref name="number-hours"/>
11803        <ref name="number-am-pm"/>
11804        <ref name="number-minutes"/>
11805        <ref name="number-seconds"/>
11806    </choice>
11807 </define>
```

Для получения информации об атрибутах, которые могут быть связаны с элементами стиля даты, см. раздел 14.7.9.

Элемент `<number:date-style>` может содержать следующие элементы:

- `<number:day>` – день месяца;
- `<number:month>` – месяц;
- `<number:year>` – год;
- `<number:era>` – эра;
- `<number:day-of-week>` – день недели;
- `<number:week-of-year>` – неделя года;
- `<number:quarter>` – квартал.

День месяца

Элемент `<number:day>` определят день месяца в дате.

Если этот элемент используется, он должен быть включен в элемент `<number:date-style>`.

```
11808 <define name="number-day">
11809    <element name="number:day">
11810        <ref name="number-day-attlist"/>
11811        <ref name="common-calendar-attlist"/>
11812        <empty/>
11813    </element>
11814 </define>
```

Атрибут number:style может быть использован с этим элементом. Для получения информации об атрибутах, которые могут быть связаны с этим элементом, см. раздел 14.7.11.

Атрибут формата

Атрибут number:style устанавливает, в каком из двух форматов, кратком или полном, отображается элемент дня месяца. Значение этого атрибута может быть short или long. Смысл этих значений зависит от значения атрибута number:format-source, который присоединен к стилю даты.

Если значения атрибута number:format-source для дней равно fixed, то:

- short означает, что день месяца отображается с использованием одной или двух цифр;
- long означает, что день месяца отображается с использованием двух цифр.

```
11815 <define name="number-day-attlist" combine="interleave">
11816     <optional>
11817         <attribute name="number:style" a:defaultValue="short">
11818             <choice>
11819                 <value>short</value>
11820                 <value>long</value>
11821             </choice>
11822         </attribute>
11823     </optional>
11824 </define>
```

Месяц

Элемент <number:month> определяет месяц в дате.

Если этот элемент используется, то он должен быть включен в элемент <number:date-style>.

```
11825 <define name="number-month">
11826     <element name="number:month">
11827         <ref name="number-month-attlist"/>
11828         <ref name="common-calendar-attlist"/>
11829         <empty/>
11830     </element>
11831 </define>
```

Атрибуты number:textual и number:style могут быть использованы с этим элементом. Для получения информации об атрибутах, которые могут быть связаны с этим элементом, см. раздел 14.7.11.

Атрибут текстового представления

Атрибут number:textual определяет, как отображается элемент месяц даты: либо как название, либо как номер. Если значение этого атрибута true, то название месяца будет отображено. Если значение атрибута false, то будет отображен номер месяца.

```
<define name="number-month-attlist" combine="interleave">
    <optional>
        <attribute name="number:textual" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Атрибут притяжательной формы

Атрибут number:possessive-form определяет, как месяц будет отображен: либо как есть (например "17 January 2004"), либо в притяжательной форме (например "17th day of January"). Если значение этого атрибута true, то название месяца будет отображено в притяжательной форме. Если значение атрибута false, то месяц будет отображен как есть.

```
<define name="number-month-attlist" combine="interleave">
    <optional>
        <attribute name="number:possessive-form" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Атрибут формата

Атрибут number:style определяет, в каком из двух форматов, кратком или полном, будет отображен элемент месяца. Значение этого атрибута может быть short или long. Смысл этих значений зависит от атрибута number:format-source, который присоединен к стилю даты.

Если значение атрибута number:format-source для месяцев равно fixed, то:

- short означает, что сокращенное название месяца будет отображено или месяц будет отображен с использованием одной или двух цифр;
- long означает, что полное название месяца будет отображено или месяц будет отображен с использованием двух цифр.

```
<define name="number-month-attlist" combine="interleave">
    <optional>
        <attribute name="number:style" a:defaultValue="short">
            <choice>
                <value>short</value>
```

```
11851                <value>long</value>
11852            </choice>
11853        </attribute>
11854    </optional>
11855 </define>
```

Год

Элемент <number:year> определяет год в дате.

Если этот элемент используется, то он должен быть включен в элемент <number:date-style>.

```
11856 <define name="number-year">
11857     <element name="number:year">
11858         <ref name="number-year-attlist"/>
11859         <ref name="common-calendar-attlist"/>
11860         <empty/>
11861     </element>
11862 </define>
```

Атрибут number:style может быть использован с этим элементом. Для получения информации об атрибутах, которые могут быть связаны с этим элементом, см. раздел 14.7.11.

Атрибут формата

Атрибут number:style определяет, в каком из двух форматов, кратком или полном, будет отображен элемент года. Значение этого атрибута может быть short или long. Смысл этих значений зависит от атрибута number:format-source, который присоединен к стилю даты.

Если значение атрибута number:format-source для лет равно fixed, то:

- short означает, что год будет отображен с использованием двух цифр;
- long означает, что год будет отображен с использованием четырех цифр.

```
11863 <define name="number-year-attlist" combine="interleave">
11864     <optional>
11865         <attribute name="number:style" a:defaultValue="short">
11866             <choice>
11867                 <value>short</value>
11868                 <value>long</value>
11869             </choice>
11870         </attribute>
11871     </optional>
11872 </define>
```

Эра

Элемент <number:era> определяет эру, в которой считаются годы.

Если этот элемент используется, то он должен быть включен в элемент `<number:date-style>`.

```
11873 <define name="number-era">
11874     <element name="number:era">
11875         <ref name="number-era-attlist"/>
11876         <ref name="common-calendar-attlist"/>
11877         <empty/>
11878     </element>
11879 </define>
```

Атрибут `number:style` может использоваться с этим элементом. Для получения информации об атрибутах, которые могут быть связаны с этим элементом, см. раздел 14.7.11.

Атрибут формата

Атрибут `number:style` определяет, в каком из двух форматов, кратком или полном, будет отображен элемент эры. Значение этого атрибута может быть `short` или `long`. Смысл этих значений зависит от значения атрибута `number:format-source`, который присоединен к стилю даты.

Если значение атрибута `number:format-source` для эры равно `fixed`, то:

- `short` означает, что используется сокращенное название эры;
- `long` означает, что используется полное название эры.

```
11880 <define name="number-era-attlist" combine="interleave">
11881     <optional>
11882         <attribute name="number:style" a:defaultValue="short">
11883             <choice>
11884                 <value>short</value>
11885                 <value>long</value>
11886             </choice>
11887         </attribute>
11888     </optional>
11889 </define>
```

День недели

Элемент `<number:day-of-week>` определяет день недели в дате.

Если этот элемент используется, он должен быть включен в элемент `<number:date-style>`.

```
11890 <define name="number-day-of-week">
11891     <element name="number:day-of-week">
11892         <ref name="number-day-of-week-attlist"/>
11893         <ref name="common-calendar-attlist"/>
11894         <empty/>
11895     </element>
11896 </define>
```

Атрибут number:style может использоваться с этим элементом. Для получения информации об атрибутах, которые могут быть связаны с этим элементом, см. раздел 14.7.11.

Атрибут формата

Атрибут number:style определяет в каком из двух форматов, коротком или полном, будет отображен элемент дня недели.

Значение атрибута может быть short или long. Смысл этих значений зависит от значения атрибута number:format-source, который присоединен к стилю даты.

Если значение атрибута number:format-source для дня недели равно fixed, то:

- short означает, что будет отображено сокращенное название дня;
- long означает, что будет отображено полное название дня.

```
<define name="number-day-of-week-attlist" combine="interleave">
    <optional>
        <attribute name="number:style" a:defaultValue="short">
            <choice>
                <value>short</value>
                <value>long</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Неделя года

Элемент <number:week-of-year> определяет неделю года в дате.

Если этот элемент используется, он должен быть включен в элемент <number:date-style>.

```
<define name="number-week-of-year">
    <element name="number:week-of-year">
        <ref name="common-calendar-attlist"/>
        <empty/>
    </element>
</define>
```

Для получения информации об атрибутах, которые могут быть связаны с этим элементом, см. раздел 14.7.11.

Квартал

Элемент <number:quarter> определяет квартал года в дате.

Если этот элемент используется, он должен быть включен в элемент `<number:date-style>`.

```
<define name="number-quarter">
    <element name="number:quarter">
        <ref name="number-quarter-attlist"/>
        <ref name="common-calendar-attlist"/>
        <empty/>
    </element>
</define>
```

Атрибут `number:style` может использоваться с этим элементом. Для получения информации об атрибутах, которые могут быть связаны с этим элементом,см. раздел 14.7.11.

Атрибут формата

Атрибут `number:style` определяет, в каком из двух форматов, кратком или полном, будет отображен элемент квартала.

Значение атрибута может быть `short` или `long`. Смысл этих значений зависит от значения атрибута `number:format-source` который присоединен к стилю даты.

Если значение атрибута `number:format-source` для квартала равно `fixed`, то:

- `short` означает, что будет отображено сокращенное название квартала, например Q1;
- `long` означает, что будет отображено полное название квартала, например Quarter 1;

```
<define name="number-quarter-attlist" combine="interleave">
    <optional>
        <attribute name="number:style" a:defaultValue="short">
            <choice>
                <value>short</value>
                <value>long</value>
            </choice>
        </attribute>
    </optional>
</define>
```

14.7.5 Стиль времени

Элемент `<number:time-style>` определяет стиль для значений времени.

Этот элемент может содержать *один* экземпляр каждого из следующих элементов: `<number:hours>`, `<number:minutes>`, `<number:seconds>` и `<number:am-pm>`.

Элемент `<number:time-style>` может также содержать элементы `<number:text>` , которые отображают дополнительный текст, но не может содержать два этих элемента подряд. Кроме того, он может содержать элемент `<style:text-properties>` и элемент `<style:map>`.

```
<define name="number-time-style">
    <element name="number:time-style">
        <ref name="number-time-style-attlist"/>
        <ref name="common-data-style-attlist"/>
        <ref name="common-format-source-attlist"/>
        <optional>
            <ref name="style-text-properties"/>
        </optional>
        <!-- This DTD does not reflect the fact that some elements must not -->
        <!-- occur more than once. -->
        <optional>
            <ref name="number-text"/>
        </optional>
        <oneOrMore>
            <ref name="any-time"/>
            <optional>
                <ref name="number-text"/>
            </optional>
        </oneOrMore>
        <zeroOrMore>
            <ref name="style-map"/>
        </zeroOrMore>
    </element>
</define>

<define name="any-time">
    <choice>
        <ref name="number-hours"/>
        <ref name="number-am-pm"/>
        <ref name="number-minutes"/>
        <ref name="number-seconds"/>
    </choice>
</define>
```

Для получения информации об атрибутах, которые могут быть связаны с элементами стиля времени, см. раздел 14.7.9.

Следующие элементы могут содержаться в элементе `<number:time-style>`:

- `<number:hours>` – часы;
- `<number:minutes>` – минуты;
- `<number:seconds>` – секунды;
- `<number:am-pm>` – am/pm (до полудня/после полудня).

Усечение значения времени

Если время или промежуток времени слишком большие, чтобы быть отображенными с использование диапазона по умолчанию для временных компонент (от 0 до 23 для `<number:hours>`), тогда атрибут `number:truncate-on-overflow` может быть использован для определения того, должны ли быть усечены значения времени или диапазона, или они должны быть расширены.

```
<define name="number-time-style-attlist" combine="interleave">
    <optional>
        <attribute name="number:truncate-on-overflow" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Часы

Элемент `<number:hours>` используется, если часы должны быть отображены как часть даты или времени.

```
<define name="number-hours">
    <element name="number:hours">
        <ref name="number-hours-attlist"/>
        <empty/>
    </element>
</define>
```

Атрибут формата

Атрибут `number:style` определяет, в каком из двух форматов, кратком или полном, будет отображен элемент часов.

Значение атрибута может быть `short` или `long`. Смысл этих значений зависит от значения атрибута `number:format-source`, который присоединен к стилю времени.

Если значение атрибута `number:format-source` для часов равно `fixed`, то:

- `short` означает, что в часах будет отображена как минимум одна цифра;
- `long` означает, что в часах будут отображены как минимум две цифры.

```
<define name="number-hours-attlist" combine="interleave">
    <optional>
        <attribute name="number:style" a:defaultValue="short">
            <choice>
                <value>short</value>
                <value>long</value>
            </choice>
        </attribute>
```

```
11984     </optional>
11985 </define>
```

Минуты

Элемент `<number:minutes>` используется, если минуты должны быть отображены как часть даты или времени.

```
11986 <define name="number-minutes">
11987     <element name="number:minutes">
11988         <ref name="number-minutes-attlist"/>
11989         <empty/>
11990     </element>
11991 </define>
```

Атрибут формата

Атрибут `number:style` определяет, в каком из двух форматов, коротком или полном, будет отображен элемент минут.

Значение атрибута может быть `short` или `long`. Смысл этих значений зависит от значения атрибута `number:format-source` который присоединен к стилю времени.

Если значение атрибута `number:format-source` для минут равно `fixed`, то:

- `short` означает, что в минутах будет отображена как минимум одна цифра;
- `long` означает, что в минутах будут отображены как минимум две цифры.

```
11992 <define name="number-minutes-attlist" combine="interleave">
11993     <optional>
11994         <attribute name="number:style" a:defaultValue="short">
11995             <choice>
11996                 <value>short</value>
11997                 <value>long</value>
11998             </choice>
11999         </attribute>
12000     </optional>
12001 </define>
```

Секунды

Элемент `<number:seconds>` используется, если секунды должны быть отображены как часть даты или времени.

```
12002 <define name="number-seconds">
12003     <element name="number:seconds">
12004         <ref name="number-seconds-attlist"/>
12005         <empty/>
12006     </element>
```

```
12007 </define>
```

Атрибут формата

Атрибут `number:style` определяет, в каком из двух форматов, коротком или полном, будет отображен элемент секунд.

Значение атрибута может быть `short` или `long`. Смысл этих значений зависит от значения атрибута `number:format-source` который присоединен к стилю времени.

Если значение атрибута `number:format-source` для секунд равно `fixed`, то:

- `short` означает, что в секундах будет отображена как минимум одна цифра;
- `long` означает, что в секундах будут отображены как минимум две цифры.

```
12008 <define name="number-seconds-attlist" combine="interleave">
12009     <optional>
12010         <attribute name="number:style" a:defaultValue="short">
12011             <choice>
12012                 <value>short</value>
12013                 <value>long</value>
12014             </choice>
12015         </attribute>
12016     </optional>
12017 </define>
```

Атрибут десятичных знаков

Атрибут `number:decimal-places` определяет число десятичных знаков, которые используются при отображении дробной части.

Если этот атрибут не представлен или значение этого атрибута равно `0`, дробная часть не отображается.

```
12018 <define name="number-seconds-attlist" combine="interleave">
12019     <optional>
12020         <attribute name="number:decimal-places" a:defaultValue="0">
12021             <ref name="integer"/>
12022         </attribute>
12023     </optional>
12024 </define>
```

AM/PM

Элемент `<number:am-pm>` определяет, включено ли AM/PM как часть даты или времени.

Если элемент <number:am-pm> содержится в стиле даты или времени, то часы будут отображаться значениями от 1 до 12.

```
<define name="number-am-pm">
    <element name="number:am-pm">
        <empty/>
    </element>
</define>
```

14.7.6 Логический стиль

Элемент <number:boolean-style> определяет стиль для логических значений.

Этот элемент может содержать один элемент <number:boolean>, до или после которого могут стоять элементы <number:text>. Кроме того, он может содержать элемент <style:text-properties> и элемент <style:map>.

```
<define name="number-boolean-style">
    <element name="number:boolean-style">
        <ref name="common-data-style-attlist"/>
        <optional>
            <ref name="style-text-properties"/>
        </optional>
        <optional>
            <ref name="number-text"/>
        </optional>
        <optional>
            <ref name="number-boolean"/>
            <optional>
                <ref name="number-text"/>
            </optional>
        </optional>
        <zeroOrMore>
            <ref name="style-map"/>
        </zeroOrMore>
    </element>
</define>
```

Логическое выражение

Элемент <number:boolean> содержит логическое выражение в логическом стиле.

```
<define name="number-boolean">
    <element name="number:boolean">
        <empty/>
    </element>
</define>
```

14.7.7 Текстовый стиль

Элемент <number:text-style> описывает стиль для отображения текста.

Этот элемент может содержать несколько элементов <number:text-content>. Он также содержит элементы <number:text>, которые отображают дополнительный текст, но он не может содержать два таких элемента подряд. Он может содержать также элемент <style:text-properties> и элемент <style:map>. Элементы <number:text-content> отражают переменное текстовое содержимое, которое будет отображено, тогда как элементы <number:text> содержат любой дополнительный фиксированный текст, который будет отображен.

```
<define name="number-text-style">
    <element name="number:text-style">
        <ref name="common-data-style-attlist"/>
        <optional>
            <ref name="style-text-properties"/>
        </optional>
        <optional>
            <ref name="number-text"/>
        </optional>
        <zeroOrMore>
            <ref name="number-text-content"/>
            <optional>
                <ref name="number-text"/>
            </optional>
        </zeroOrMore>
        <zeroOrMore>
            <ref name="style-map"/>
        </zeroOrMore>
    </element>
</define>
```

Для получения информации об атрибутах, которые могут быть связаны с элементами текстового стиля, см. раздел 14.7.9.

Фиксированный текст

Элемент <number:text> содержит несколько фиксированных текстов для стиля данных.

Этот элемент содержится во всех элементах стиля данных.

```
<define name="number-text">
    <element name="number:text">
        <text/>
    </element>
</define>
```

Текстовое содержимое

Элемент <number:text-content> содержит изменяемое текстовое содержимое текстового стиля.

```
12080 <define name="number-text-content">
12081     <element name="number:text-content">
12082         <empty/>
12083     </element>
12084 </define>
```

14.7.8 Общие элементы стиля данных

Следующие общие элементы могут быть включены в элементы стиля данных:

- свойства форматирования текста;
- соотнесение стилей.

Свойства форматирования

Элемент `<style:text-properties>` определяет параметры форматирования, которые применяются к любому тексту отображенному с использованием стиля данных. Для получения информации о параметрах форматирования элементов, см. раздел 15.4.

Назначением установки текстовых параметров форматирования внутри стиля данных в основном является выделение определенных значений (например, отрицательных) с использованием соотнесения стилей. Для этой цели стили данных обычно поддерживают только немногие текстовые параметры форматирования, например цвет текста. Также могут существовать ограничения для значений текстовых параметров форматирования. Например, только значение, разрешенное для цвета текста, может быть прочитано.

Соотнесение стилей

Элемент `<style:map>` устанавливает альтернативный стиль данных для соотнесения, если существует точное условие. Для получения информации об элементе `<style:map>` см. раздел 14.1.1.

Существуют следующие правила для использования элемента соотнесения стилей с элементами стиля данных.

- Стиль, на который ссылается атрибут `style:apply-style`, должен быть того же типа, как и стиль, содержащий соотнесение.

- Условие должно быть в формате value() op n, где op — это оператор отношения и n — это число. Для логического стиля значение условия должно быть «истина» или «ложь».

14.7.9 Общие атрибуты стиля данных

Многие из атрибутов стиля данных применимы к более чем одному элементу стиля данных. Следующие атрибуты стиля данных являются общими для многих элементов стиля данных:

- имя;
- язык;
- страна;
- заголовок;
- изменчивость;
- автоматический порядок;
- источник форматирования;
- транслитерация.

Имя

Атрибут `style:name` определяет имя стиля данных. Он может быть использован со всеми элементами стиля данных.

```
<define name="common-data-style-attlist" combine="interleave">
    <attribute name="style:name">
        <ref name="styleName"/>
    </attribute>
</define>
```

Отображаемое имя

Атрибут `style:display-name` определяет имя стиля в том виде, в котором оно должно появляться в пользовательском интерфейсе. В отличие от самого имени стиля, это имя может содержать произвольные символы. Если этот атрибут не используется, отображаемым именем становится имя стиля.

Атрибут `style:display-name` может быть использован со всеми элементами стиля данных.

```
<define name="style-data-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:display-name">
```

```
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Язык

Атрибут `number:language` определяет язык стиля. Значение атрибута — это код языка в соответствии с [RFC3066]. Код языка используется для получения информации о некоторых свойствах отображения, которые зависят от языка. Атрибут языка может быть использован со всеми элементами стиля данных.

Если код языка не указан, то используются либо параметры системы, либо параметры языка системы в зависимости от того, какое значение должно быть получено.

```
<define name="common-data-style-attlist" combine="interleave">
    <optional>
        <attribute name="number:language">
            <ref name="languageCode"/>
        </attribute>
    </optional>
</define>
```

Страна

Атрибут `number:country` определяет страну для стиля. Значение атрибута — это код страны в соответствии с [RFC3066]. Код страны используется для получения информации о некоторых свойствах отображения, которые зависят от страны. Атрибут страны может быть использован со всеми элементами стиля данных.

Если код страны не указан, то используются либо параметры системы, либо параметры страны системы в зависимости от того, какое значение должно быть получено.

```
<define name="common-data-style-attlist" combine="interleave">
    <optional>
        <attribute name="number:country">
            <ref name="countryCode"/>
        </attribute>
    </optional>
</define>
```

Заголовок

Атрибут `number:title` определяет заголовок стиля данных. Он может использоваться со всеми элементами стиля данных.

```
<define name="common-data-style-attlist" combine="interleave">
    <optional>
        <attribute name="number:title"/>
    </optional>
</define>
```

Изменчивость

Иногда, когда документ открыт, ссылки имеются не на все стили, содержащиеся в документе. Приложение может сохранить или отбросить эти неиспользуемые стили. Это можно контролировать с помощью атрибута `style:volatile`, который поддерживается всеми элементами стиля данных.

Если значение атрибута равно `true`, приложение сохранит стиль по возможности. Если значение равно `false`, приложение отбросит неиспользуемые стили.

```
<define name="common-data-style-attlist" combine="interleave">
    <optional>
        <attribute name="style:volatile">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Автоматический порядок

Атрибут `number:automatic-order` может быть использован для автоматического упорядочивания данных в соответствии с заданным по умолчанию порядком для языка и страны стиля данных. Этот атрибут используется со следующими элементами:

- `<number:currency-style>`, где число и символ валюты переупорядочиваются;
- `<number:date-style>`, где дочерние элементы `<number:date-style>` не являющиеся `<number:text>` или `<style:text-properties>` переупорядочиваются.

Значение атрибута может быть `true` или `false`.

```
<define name="common-auto-reorder-attlist" combine="interleave">
    <optional>
        <attribute name="number:automatic-order" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Источник форматирования

Атрибут number:format-source определяет источник short и long форматов отображения. Он используется со следующими элементами:

- <number:date-style>
- <number:time-style>

Значение этого атрибута может быть fixed или language.

Если значение равно fixed, то смысл значений short и long атрибута number:style соответствует описанному в этой спецификации.

Если значение атрибута number:format-source равно language, то смысл краткой и полной формы зависит от языка и страны стиля даты, или, если ни один из них не установлен, то приложения должны использовать параметры системы для краткого и полного форматов даты и времени.

```
12130 <define name="common-format-source-attlist">
12131     <optional>
12132         <attribute name="number:format-source" a:defaultValue="fixed">
12133             <choice>
12134                 <value>fixed</value>
12135                 <value>language</value>
12136             </choice>
12137         </attribute>
12138     </optional>
12139 </define>
```

14.7.10 Транслитерация

Некоторое количество атрибутов number:transliteration-* определяют собственную систему счисления стиля для отображения числа, используя, например, числовые символы CJK (China, Japan, Korea — Китай, Япония, Корея). Нотация заимствована из черновика W3C XSLT 2.0, см. § 12.3 в [XSLT2]. Однако для однозначного разграничения всех возможных собственных систем счисления, необходима совокупность дополнительных атрибутов. Например, в Корее используется 11 различных систем, где числа не всегда отличаются, но существуют короткие и длинные, формальные и неформальные формы.

Атрибуты транслитерации могут использоваться со всеми элементами стиля данных.

Формат транслитерации

Атрибут `number:transliteration-format` определяет, какое количество символов следует использовать. Значение этого атрибута равно «1», выраженное в собственной системе счисления.

Если формат не установлен, тогда по умолчанию используется ASCII представление арабских цифр, атрибуты других транслитераций в этом случае игнорируются.

```
12140 <define name="common-data-style-attlist" combine="interleave">
12141     <optional>
12142         <attribute name="number:transliteration-format" a:defaultValue="1">
12143             <ref name="string"/>
12144         </attribute>
12145     </optional>
12146 </define>
```

Язык транслитерации

Атрибут `number:transliteration-language` определяет язык, которому принадлежит собственная система счисления. Значения этого атрибута — код языка в соответствии с [RFC3066].

Если не установлена комбинация язык/страна (параметры локализации), то используются параметры локализации стиля данных.

```
12147 <define name="common-data-style-attlist" combine="interleave">
12148     <optional>
12149         <attribute name="number:transliteration-language">
12150             <ref name="countryCode"/>
12151         </attribute>
12152     </optional>
12153 </define>
```

Страна транслитерации

Атрибут `number:transliteration-country` определяет страну, которой принадлежит собственная система счисления. Значение этого атрибута — код страны в соответствии с [RFC3066].

Если не установлена комбинация язык/страна (параметры локализации), то используются параметры локализации стиля данных.

```
12154 <define name="common-data-style-attlist" combine="interleave">
12155     <optional>
12156         <attribute name="number:transliteration-country">
12157             <ref name="countryCode"/>
12158         </attribute>
12159     </optional>
```

```
12160 </define>
```

Стиль транслитерации

Атрибут `number:transliteration-style` определяет, какому стилю принадлежит собственная система счисления. Если более чем одна собственная система счисления совпадает с форматом транслитерации, тогда значением данного атрибута выбирается одна из них. Краткий стиль должен иметь однозначное отображение арабских цифр в собственную систему счисления, если это возможно.

```
12161 <define name="common-data-style-attlist" combine="interleave">
12162     <optional>
12163         <attribute name="number:transliteration-style" a:defaultValue="short">
12164             <choice>
12165                 <value>short</value>
12166                 <value>medium</value>
12167                 <value>long</value>
12168             </choice>
12169         </attribute>
12170     </optional>
12171 </define>
```

14.7.11 Атрибуты общих элементов стиля данных

Большое число атрибутов стиля применимо к более чем одному элементу стиля. Следующие атрибуты являются общими для большого числа элементов стиля:

- десятичные знаки;
- минимальное количество цифр в целом числе;
- группирующий разделитель;
- замещение в десятичных дробях;
- минимальное количество цифр в экспоненте;
- минимальное количество цифр в числителе;
- минимальное количество цифр в знаменателе;
- календарная система.

Десятичные знаки

Атрибут `number:decimal-places` определяет число десятичных знаков, которые будут отображаться. Этот атрибут поддерживается для следующих элементов:

- <number:number>
- <number:scientific-number>

Если этот атрибут не определен, то будет использоваться количество десятичных знаков по умолчанию.

```
12172 <define name="common-decimal-places-attlist">
12173     <optional>
12174         <attribute name="number:decimal-places">
12175             <ref name="integer"/>
12176         </attribute>
12177     </optional>
12178 </define>
```

Минимальное количество цифр в целом числе

Атрибут number:min-integer-digits определяет минимальное количество цифр в целом числе, которые будут отображаться в числе, числе в научном формате или в дроби. Этот атрибут поддерживается для следующих элементов:

- <number:number>
- <number:scientific-number>
- <number:fraction>

Если этот атрибут не определен, то будет использоваться количество цифр в представлении целого числа по умолчанию.

```
12179 <define name="common-number-attlist" combine="interleave">
12180     <optional>
12181         <attribute name="number:min-integer-digits">
12182             <ref name="integer"/>
12183         </attribute>
12184     </optional>
12185 </define>
```

Группирующий разделитель

Атрибут number:grouping определяет, должны ли быть сгруппированы цифры числа с использованием символа-разделителя. Атрибут поддерживает для следующих элементов:

- <number:number>
- <number:scientific-number>
- <number:fraction>

Группирующий символ, который используется, и число цифр, которые сгруппированы, зависят от языка и страны стиля.

```
12186 <define name="common-number-attlist" combine="interleave">
12187     <optional>
12188         <attribute name="number:grouping" a:defaultValue="false">
12189             <ref name="boolean"/>
12190         </attribute>
12191     </optional>
12192 </define>
```

Календарная система

Атрибут `number:calendar` определяет календарную систему, использующую части даты. Этот атрибут поддерживается для следующих элементов:

- `<number:day>`
- `<number:month>`
- `<number:year>`
- `<number:era>`
- `<number:day-of-week>`
- `<number:week-of-year>`
- `<number:quarter>`

Атрибут может принимать значения `gregorian`, `gengou`, `ROC`, `hanja_yoil`, `hanja`, `hijri`, `jewish`, `buddhist` или произвольное строковое значение. Если этот атрибут не определен, то используется календарная система по умолчанию.

```
12193 <define name="common-calendar-attlist" combine="interleave">
12194     <optional>
12195         <attribute name="number:calendar">
12196             <choice>
12197                 <value>gregorian</value>
12198                 <value>gengou</value>
12199                 <value>ROC</value>
12200                 <value>hanja_yoil</value>
12201                 <value>hanja</value>
12202                 <value>hijri</value>
12203                 <value>jewish</value>
12204                 <value>buddhist</value>
12205                 <ref name="string"/>
12206             </choice>
12207         </attribute>
12208     </optional>
12209 </define>
```

14.8 Стили текста

14.8.1 Стили текста

Стили текста — это элементы `<style:style>` из семейства `text`. Они могут использоваться во всех типах приложений для присвоения свойств форматирования фрагментам текста. Они поддерживают свойства текста, как описано в разделе 15.4.

```
<define name="style-style-content" combine="choice">
    <group>
        <attribute name="style:family">
            <value>text</value>
        </attribute>
        <optional>
            <ref name="style-text-properties"/>
        </optional>
    </group>
</define>
```

14.8.2 Стили абзацев

Стили абзацев — это элементы `<style:style>` из семейства `paragraph`. Они могут использоваться во всех типах приложений для присвоения свойств форматирования абзацам и заголовкам. Они поддерживают как свойства абзацев, описанные в разделе 15.5, так и свойства текста, описанные в разделе 15.4.

```
<define name="style-style-content" combine="choice">
    <group>
        <attribute name="style:family">
            <value>paragraph</value>
        </attribute>
        <optional>
            <ref name="style-paragraph-properties"/>
        </optional>
        <optional>
            <ref name="style-text-properties"/>
        </optional>
    </group>
</define>
```

14.8.3 Стили разделов

Стили разделов — это элементы `<style:style>` из семейства `section`. Они могут использоваться в текстовых приложениях для присвоения свойств форматирования разделу текста. Они поддерживают свойства разделов, как описано в разделе 15.7.

```
<define name="style-style-content" combine="choice">
```

```
12234     <group>
12235         <attribute name="style:family">
12236             <value>section</value>
12237         </attribute>
12238         <optional>
12239             <ref name="style-section-properties"/>
12240         </optional>
12241     </group>
12242 </define>
```

14.8.4 Стиль транскрипции

Стиль транскрипции определяет, как отображается текст транскрипции относительно основного текста. Он представлен элементом `<style:style>` из семейства `ruby`. Стиль транскрипции присваивается элементу транскрипции посредством атрибута `text:style-name`. Стили транскрипции поддерживают свойства форматирования, описанные в разделе 15.6.

```
12243 <define name="style-style-content" combine="choice">
12244     <group>
12245         <attribute name="style:family">
12246             <value>ruby</value>
12247         </attribute>
12248         <optional>
12249             <ref name="style-ruby-properties"/>
12250         </optional>
12251     </group>
12252 </define>
```

14.9 Расширенные стили текста

14.9.1 Конфигурация нумерации строк

Документ может не содержать *ни одного* или содержать *один* элемент конфигурации нумерации строк `<text:linenumbering-configuration>` внутри элемента `<office:styles>`. Если элемент не присутствует, используется конфигурация нумерации строк по умолчанию. Нумерация строк по умолчанию может меняться в зависимости от офисного приложения, но каждый документ, сохраняемый приложением, поддерживающим нумерацию строк, должен содержать элемент конфигурации нумерации строк.

```
12253 <define name="text-linenumbering-configuration">
12254     <element name="text:linenumbering-configuration">
12255         <ref name="text-linenumbering-configuration-attlist"/>
12256         <optional>
12257             <ref name="text-linenumbering-separator"/>
12258         </optional>
12259     </element>
12260 </define>
```

С элементом `<text:linenumbering-configuration>` могут быть связаны следующие атрибуты:

- включение нумерации строк;
- числовой формат;
- стиль текста;
- приращение;
- позиция;
- смещение;
- счетчик пустых строк;
- счетчик строк в текстовых блоках;
- перезапуск нумерации на каждой странице.

Следующий элемент может быть включен в элемент `<text:linenumbering-separator>`:

- разделитель.

Включение нумерации строк

Атрибут `text:number-lines` определяет, должны ли нумероваться строки.

```
<define name="text-linenumbering-configuration-attlist" combine="interleave">
    <optional>
        <attribute name="text:number-lines" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Числовой формат

В разделе 12.2 дана более подробная информация об атрибутах числового формата. Атрибуты, описанные в разделе 12.2, могут также быть связаны с элементом `<text:linenumbering-configuration>`.

```
<define name="text-linenumbering-configuration-attlist" combine="interleave">
    <optional>
        <ref name="common-num-format-attlist"/>
    </optional>
</define>
```

Стиль текста

Атрибут `text:style-name` определяет стиль текста для всех номеров строк. Значение этого атрибута — имя стиля текста, применяемого ко всем номерам строк.

```
<define name="text-linenumbering-configuration-attlist" combine="interleave">
    <optional>
        <attribute name="text:style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Приращение

Атрибут `text:increment` определяет, что номера строк, кратные данному приращению, подлежат нумерации. Например, если приращение равно 5, то нумероваться будут только пятая, десятая, пятнадцатая и так далее строки.

```
<define name="text-linenumbering-configuration-attlist" combine="interleave">
    <optional>
        <attribute name="text:increment">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Позиция

Атрибут `text:position` определяет, на каких полях печатаются номера строк: левых, правых, внутренних или внешних.

```
<define name="text-linenumbering-configuration-attlist" combine="interleave">
    <optional>
        <attribute name="text:number-position" a:defaultValue="left">
            <choice>
                <value>left</value>
                <value>rigth</value>
                <value>inner</value>
                <value>outer</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Смещение

Атрибут `text:offset` определяет расстояние между номером строки и полем.

```
<define name="text-linenumbering-configuration-attlist" combine="interleave">
```

```
12300     <optional>
12301         <attribute name="text:offset">
12302             <ref name="nonNegativeLength"/>
12303         </attribute>
12304     </optional>
12305 </define>
```

Счетчик пустых строк

Атрибут `text:count-empty-lines` определяет, учитываются ли пустые строки при подсчете строк. Если значение этого атрибута `true`, то пустые строки учитываются.

```
12306 <define name="text-linenumbering-configuration-attlist" combine="interleave">
12307     <optional>
12308         <attribute name="text:count-empty-lines" a:defaultValue="true">
12309             <ref name="boolean"/>
12310         </attribute>
12311     </optional>
12312 </define>
```

Счетчик строк в текстовых блоках

Атрибут `text:count-in-text-boxes` определяет, учитывается ли текст в текстовых окнах при подсчете числа строк. Если значение этого атрибута `true`, то текст в текстовых окнах учитывается.

```
12313 <define name="text-linenumbering-configuration-attlist" combine="interleave">
12314     <optional>
12315         <attribute name="text:count-in-text-boxes" a:defaultValue="false">
12316             <ref name="boolean"/>
12317         </attribute>
12318     </optional>
12319 </define>
```

Перезапуск нумерации на каждой странице

Атрибут `text:restart-on-page` определяет, должен ли счетчик строк устанавливаться в 1 в начале каждой страницы.

Если значение этого атрибута равно `true`, то счетчик строк устанавливается в 1 в начале каждой страницы, то есть нумерация строк независима на каждой странице. По умолчанию значение этого атрибута ложь (`false`), то есть нумерация строк едина в пределах всего документа.

```
12320 <define name="text-linenumbering-configuration-attlist" combine="interleave">
12321     <optional>
12322         <attribute name="text:restart-on-page" a:defaultValue="false">
12323             <ref name="boolean"/>
12324         </attribute>
12325     </optional>
```

```
12326 </define>
```

Разделитель

Элемент `<text:linenumbering-separator>` содержит текст, который отображается в качестве разделителя. Разделитель — это текст, отображаемый вместо номера строки в строках, где не отображается номер.

Этот элемент содержится в элементе конфигурации нумерации строк. Если элемент отсутствует, разделитель не отображается.

Атрибут `text:increment` элемента вызывает отображение разделителя на строках, кратных заданному приращению. Например, если приращение равно 2, то разделитель получат только вторая, четвертая, шестая и так далее строки, если только на них уже не отображается номер.

```
12327 <define name="text-linenumbering-separator">
12328     <element name="text:linenumbering-separator">
12329         <optional>
12330             <attribute name="text:increment">
12331                 <ref name="nonNegativeInteger"/>
12332             </attribute>
12333         </optional>
12334         <text/>
12335     </element>
12336 </define>
```

14.9.2 Элемент конфигурации заметок

Документ в формате OpenDocument содержит не более одного элемента конфигурации заметок для каждого класса заметок, используемого в документе. Если элемент конфигурации заметок отсутствует, используется конфигурация заметок по умолчанию.

```
12337 <define name="text-notes-configuration">
12338     <element name="text:notes-configuration">
12339         <ref name="text-notes-configuration-content"/>
12340     </element>
12341 </define>
```

С элементом `<text:notes-configuration>` могут быть связаны следующие атрибуты:

- класс заметок;
- стиль цитаты в сноске;
- стиль цитаты в тексте;
- стиль абзаца в сноске по умолчанию;

- мастер-страница;
- начальное значение;
- числовой формат;
- схема нумерации;
- расположение сноски.

Следующий элемент может быть включен в элемент `<text:footnotes-configuration>`:

- извещение о продолжении сноски (вперед и назад).

Класс заметок

Атрибут класс заметок определяет, к каким элементам заметок применяется эта конфигурация заметок.

```
12342 <define name="text-notes-configuration-content" combine="interleave">
12343     <ref name="text-note-class"/>
12344 </define>
```

Стиль цитаты в сноске

Атрибут `text:citation-style` определяет стиль текста, который должен использоваться для цитаты в сноске.

```
12345 <define name="text-notes-configuration-content" combine="interleave">
12346     <optional>
12347         <attribute name="text:citation-style-name">
12348             <ref name="styleNameRef"/>
12349         </attribute>
12350     </optional>
12351 </define>
```

Стиль цитаты в тексте

Атрибут `text:citation-body-style-name` определяет стиль текста, который должен использоваться для цитат по ходу текста.

```
12352 <define name="text-notes-configuration-content" combine="interleave">
12353     <optional>
12354         <attribute name="text:citation-body-style-name">
12355             <ref name="styleNameRef"/>
12356         </attribute>
12357     </optional>
12358 </define>
```

Стиль абзаца в сноске по умолчанию

Стиль абзаца в сноске по умолчанию используется только для сносок, вставляемых в существующий документ. Он не используется для сносок, которые уже существуют.

```
<define name="text-notes-configuration-content" combine="interleave">
    <optional>
        <attribute name="text:default-style-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Мастер-страница

Для отображения сносок в конце документа страницы, содержащие сноски, должны быть экземплярами мастер-страницы, определенными атрибутом `text:master-page-name`.

```
<define name="text-notes-configuration-content" combine="interleave">
    <optional>
        <attribute name="text:master-page-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

Начальное значение

Атрибут `start:value` определяет значение, с которого будет начинаться нумерация сносок.

```
<define name="text-notes-configuration-content" combine="interleave">
    <optional>
        <attribute name="text:start-value">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Числовой формат

В разделе 12.2 дана информация о числовом формате для сносок.

```
<define name="text-notes-configuration-content" combine="interleave">
    <ref name="common-num-format-prefix-suffix-attlist"/>
    <optional>
        <ref name="common-num-format-attlist"/>
    </optional>
</define>
```

Схема нумерации

Атрибут `text:start-numbering-at` определяет, начинается ли заново нумерация сносок в начале документа, главы или страницы.

Примечание — [XSLT] не обладает возможностью начинать заново нумерацию сносок на каждой странице.

```
<define name="text-notes-configuration-content" combine="interleave">
    <optional>
        <attribute name="text:start-numbering-at">
            <choice>
                <value>document</value>
                <value>chapter</value>
                <value>page</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Расположение сносок

Атрибут `text:footnotes-position` определяет одну из следующих позиций для сносок:

- `text` — на странице, где расположена ссылка на сноску, прямо под текстом на странице;
- `page` — внизу страницы, где расположена ссылка на сноску;
- `section` — в конце раздела;
- `document` — в конце документа.

Примечание — [XSL] не обладает возможностью отображения сносок в конце документа. Однако таблица стилей [XSLT] может генерировать некоторые другие объекты для отображения подобных сносок.

```
<define name="text-notes-configuration-content" combine="interleave">
    <optional>
        <attribute name="text:footnotes-position">
            <choice>
                <value>text</value>
                <value>page</value>
                <value>section</value>
                <value>document</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Продолжение сноски

Элементы продолжения сносок определяют:

- текст, отображаемый в конце сноски, которая продолжается на следующей странице;
- текст, отображаемый перед продолжением сноски.

```
<define name="text-notes-configuration-content" combine="interleave">
    <optional>
        <element name="text:note-continuation-notice-forward">
            <text/>
        </element>
    </optional>
</define>
<define name="text-notes-configuration-content" combine="interleave">
    <optional>
        <element name="text:note-continuation-notice-backward">
            <text/>
        </element>
    </optional>
</define>
```

Пример — Конфигурация сносок:

```
<text:notes-configuration
    text:notes-type="footnote"
    text:citation-style="Footnote symbol"
    text:default-style="Footnote">
    <text:note-continuation-notice-forward>" .."
    </text:note-continuation-notice-forward>
    <text:note-continuation-notice-forward>".. "
    </text:note-continuation-notice-forward>
</text:notes-configuration>
```

14.9.3 Конфигурация библиографии

Элемент конфигурации библиографии `<text:bibliography-configuration>` содержится в разделе стиля документа. Он содержит информацию о том, каким образом компоненты библиографии отображаются подряд и в библиографических индексах.

```
<define name="text-bibliography-configuration">
    <element name="text:bibliography-configuration">
        <ref name="text-bibliography-configuration-attlist"/>
        <zeroOrMore>
            <ref name="text-sort-key"/>
        </zeroOrMore>
    </element>
</define>
```

Префикс и суффикс

Атрибуты `text:prefix` и `text:suffix` содержат строку, отображаемую до и после сокращенного названия или номера компонента библиографии, если он появляется в теле документа.

```
<define name="text-bibliography-configuration-attlist" combine="interleave">
```

```
12432     <optional>
12433         <attribute name="text:prefix">
12434             <ref name="string"/>
12435         </attribute>
12436     </optional>
12437     <optional>
12438         <attribute name="text:suffix">
12439             <ref name="string"/>
12440         </attribute>
12441     </optional>
12442 </define>
```

Пронумерованные компоненты

Атрибут `text:numbered-entry` определяет, отображаются ли номера компонентов библиографии вместо их сокращенных названий.

Пример —

С префиксом и суффиксом «[» и «]» компонент библиографии с сокращенным названием "Abc123" будет отображаться как «[Abc123]» в теле документа, если `text:numbered-entry` имеет значение `false`, и, например, как «[5]», если он имеет значение `true`.

```
12443 <define name="text-bibliography-configuration-attlist" combine="interleave">
12444     <optional>
12445         <attribute name="text:numbered-entries" a:defaultValue="false">
12446             <ref name="boolean"/>
12447         </attribute>
12448     </optional>
12449 </define>
```

Сортировка

Атрибут `text:sort-by-position` определяет, отображаются ли компоненты библиографии в порядке их расположения в документе или по некоторым полям компонентов, например имя автора или год издания. В последнем случае порядок сортировки компонентов определяется тройкой язык/страна/алгоритм сортировки, как указано в атрибутах `fo:language`, `fo:country` и `text:sort-algorithm` (см. также раздел 7.8).

```
12450 <define name="text-bibliography-configuration-attlist" combine="interleave">
12451     <optional>
12452         <attribute name="text:sort-by-position" a:defaultValue="true">
12453             <ref name="boolean"/>
12454         </attribute>
12455     </optional>
12456     <optional>
12457         <attribute name="fo:language">
12458             <ref name="languageCode"/>
12459         </attribute>
12460     </optional>
```

```
12461     <optional>
12462         <attribute name="fo:country">
12463             <ref name="countryCode"/>
12464         </attribute>
12465     </optional>
12466     <optional>
12467         <attribute name="text:sort-algorithm">
12468             <ref name="string"/>
12469         </attribute>
12470     </optional>
12471 </define>
```

Ключи сортировки

Элемент `<text:sort-key>` определяет единый ключ сортировки, если компоненты библиографии отображаются не в порядке их следования в документе. У него есть атрибут `text:key`, содержащий тип данных индекса компонента, который должен использоваться для сортировки (см. раздел 7.1.4), и атрибут `text:sort-ascending` , указывающий, идет ли сортировка в порядке возрастания или убывания.

```
12472 <define name="text-sort-key">
12473     <element name="text:sort-key">
12474         <ref name="text-sort-key-attlist"/>
12475         <empty/>
12476     </element>
12477 </define>
12478
12479 <define name="text-sort-key-attlist" combine="interleave">
12480     <attribute name="text:key">
12481         <choice>
12482             <value>address</value>
12483             <value>annote</value>
12484             <value>author</value>
12485             <value>bibliography-type</value>
12486             <value>booktitle</value>
12487             <value>chapter</value>
12488             <value>custom1</value>
12489             <value>custom2</value>
12490             <value>custom3</value>
12491             <value>custom4</value>
12492             <value>custom5</value>
12493             <value>edition</value>
12494             <value>editor</value>
12495             <value>howpublished</value>
12496             <value>identifier</value>
12497             <value>institution</value>
12498             <value>isbn</value>
12499             <value>issn</value>
12500             <value>journal</value>
12501             <value>month</value>
12502             <value>note</value>
12503             <value>number</value>
12504             <value>organizations</value>
12505             <value>pages</value>
12506             <value>publisher</value>
12507             <value>report-type</value>
```

```
12508            <value>school</value>
12509            <value>series</value>
12510            <value>title</value>
12511            <value>url</value>
12512            <value>volume</value>
12513            <value>year</value>
12514        </choice>
12515    </attribute>
12516    <optional>
12517        <attribute name="text:sort-ascending" a:defaultValue="true">
12518            <ref name="boolean"/>
12519        </attribute>
12520    </optional>
12521 </define>
```

14.10 Стили списков

Стили списков определяют свойства форматирования для списков. Элемент `<text:list-style>` содержит набор элементов стиля для каждого уровня списка, которые называются стили уровня списка. Есть три различных стиля уровней списка в зависимости от того, должна ли быть метка этого уровня числом, маркером или изображением.

Если стиль списка применяется к списку, но не содержит требований к оформлению какого-либо уровня списка, используется стиль следующего, ниже по иерархии, уровня списка. Если такого уровня списка не существует, используется стиль по умолчанию.

```
12522 <define name="text-list-style">
12523     <element name="text:list-style">
12524         <ref name="text-list-style-attr"/>
12525         <zeroOrMore>
12526             <ref name="text-list-style-content"/>
12527         </zeroOrMore>
12528     </element>
12529 </define>
```

Примечание — Стили списка содержат свойства, отличающиеся от стилей текста или абзаца. Поэтому они представляются другим элементом.

С элементом `<text:list-style>` могут быть связаны следующие атрибуты:

- имя;
- отображаемое имя;
- последовательная нумерация.

Имя

Атрибут `style:name` определяет имя стиля списка.

```
12530 <define name="text-list-style-attr" combine="interleave">
12531     <attribute name="style:name">
12532         <ref name="styleName"/>
12533     </attribute>
12534 </define>
```

Отображаемое имя

Атрибут `style:display-name` определяет имя стиля списка, каким оно отображается в пользовательском интерфейсе. В отличие от имени стиля как такового, это имя может содержать произвольные символы. Если этот атрибут не указан, то имя отображения совпадает с именем стиля.

```
12535 <define name="text-list-style-attr" combine="interleave">
12536     <optional>
12537         <attribute name="style:display-name">
12538             <ref name="string"/>
12539         </attribute>
12540     </optional>
12541 </define>
```

Последовательная нумерация

Атрибут `text:consecutive-numbering` определяет, используется ли последовательная нумерация для всех уровней стиля, или нумерация начинается заново на каждом уровне.

```
12542 <define name="text-list-style-attr" combine="interleave">
12543     <optional>
12544         <attribute name="text:consecutive-numbering" a:defaultValue="false">
12545             <ref name="boolean"/>
12546         </attribute>
12547     </optional>
12548 </define>
```

14.10.1 Общий атрибут стилей уровня списка

Следующий атрибут может быть в стилях всех уровней списка:

Уровень

Атрибут `text:level` определяет стиль уровня списка.

```
12549 <define name="text-list-level-style-attr">
12550     <attribute name="text:level">
12551         <ref name="positiveInteger"/>
12552     </attribute>
12553 </define>
```

14.10.2 Стиль нумерованного списка

Стиль нумерованного списка определяет стиль списка, где абзацы списка начинаются с чисел.

```
12554 <define name="text-list-style-content" combine="choice">
12555     <element name="text:list-level-style-number">
12556         <ref name="text-list-level-style-attr"/>
12557         <ref name="text-list-level-style-number-attr"/>
12558         <optional>
12559             <ref name="style-list-level-properties"/>
12560         </optional>
12561         <optional>
12562             <ref name="style-text-properties"/>
12563         </optional>
12564     </element>
12565 </define>
```

С элементом `<text:list-level-style-number>` могут быть связаны следующие атрибуты:

- уровень (см. раздел 14.10.1);
- стиль текста;
- числовой формат;
- отображаемые уровни;
- начальное значение.

Дополнительные свойства форматирования могут содержаться в элементах `<style:list-level-properties>` и `<style:text-properties>`. Для получения дополнительной информации см. разделы 15.12 и 15.4).

Стиль текста

Атрибут `text:style-name` определяет тип шрифта, используемый для нумерации в списке.

```
12566 <define name="text-list-level-style-number-attr" combine="interleave">
12567     <optional>
12568         <attribute name="text:style-name">
12569             <ref name="styleNameRef"/>
12570         </attribute>
12571     </optional>
12572 </define>
```

Числовой формат

Подробная информация об атрибутах числового формата дана в разделе 12.2. Атрибуты, описанные в разделе 12.2, могут также быть связаны с элементом

`<text:list-level-style-number>`. Атрибут `style:num-format` может быть пустым. В этом случае номер не отображается.

```
<define name="text-list-level-style-number-attr" combine="interleave">
    <ref name="common-num-format-attlist"/>
    <ref name="common-num-format-prefix-suffix-attlist"/>
</define>
```

Отображаемые уровни

Атрибут `text:display-levels` определяет число уровней, номера которых отображаются на текущем уровне.

```
<define name="text-list-level-style-number-attr" combine="interleave">
    <optional>
        <attribute name="text:display-levels" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Пример —

Задан номер главы третьего уровня 1.2.3. Использование значений атрибута `text:display-levels` от 1 до 3 приведет к следующим результатам:

`text:display-number`	***display***
1	1
2	1.2
3	1.2.3

Начальное значение

Атрибут `text:start-value` определяет первый номер пункта списка на текущем уровне.

```
<define name="text-list-level-style-number-attr" combine="interleave">
    <optional>
        <attribute name="text:start-value" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

14.10.3 Стиль маркированного списка

Элемент стиль маркированного списка определяет стиль списка, где в начале каждого абзаца списка ставится маркер.

```
<define name="text-list-style-content" combine="choice">
    <element name="text:list-level-style-bullet">
        <ref name="text-list-level-style-attr"/>
```

```
12594         <ref name="text-list-level-style-bullet-attr"/>
12595         <optional>
12596             <ref name="style-list-level-properties"/>
12597         </optional>
12598         <optional>
12599             <ref name="style-text-properties"/>
12600         </optional>
12601     </element>
12602 </define>
```

С элементом `<text:list-level-style-bullet>` могут быть связаны следующие атрибуты:

- уровень (см. раздел 14.10.1);
- стиль текста;
- символ маркера;
- префикс и суффикс;
- относительный размер маркера.

Дополнительные свойства форматирования могут содержаться в элементах `<style:list-level-properties>` и `<style:text-properties>`. Для получения подробной информацией см. разделы 15.12 и 15.4.

Стиль текста

Атрибут `text:style-name` содержит имя стиля символа, используемого для форматирования маркера списка.

```
12603 <define name="text-list-level-style-bullet-attr" combine="interleave">
12604     <optional>
12605         <attribute name="text:style-name">
12606             <ref name="styleNameRef"/>
12607         </attribute>
12608     </optional>
12609 </define>
```

Символ маркера

Атрибут символ маркера определяет символ [UNICODE], используемый в качестве маркера в стиле маркированного списка.

Типичными символами маркера являются:

- U+2022
- ● U+25CF
- ➔ U+2794
- ➢ U+27A2
- ✗ U+2717

✔ U+2714

Эти символы могут быть недоступны в некоторых типах шрифта.

```
12610 <define name="text-list-level-style-bullet-attr" combine="interleave">
12611     <attribute name="text:bullet-char">
12612         <ref name="character"/>
12613     </attribute>
12614 </define>
```

Префикс и суффикс

Атрибуты `tyle:num-format-prefix` и `style:num-format-suffix`, упомянутые в разделе 12.2, могут быть использованы для добавления символов до или после символа маркера.

```
12615 <define name="text-list-level-style-bullet-attr" combine="interleave">
12616     <ref name="common-num-format-prefix-suffix-attlist"/>
12617 </define>
```

Относительный размер маркера

Атрибут `text:bullet-relative-size` определяет процентное соотношение размера маркера к размеру шрифта, используемого в отмечаемом маркером абзаце. Например, если значение атрибута `text:bullet-relative-size` равно 75, то размер маркера, используемого в списке, составляет 75 % от размера шрифта в абзаце.

```
12618 <define name="text-list-level-style-bullet-attr" combine="interleave">
12619     <optional>
12620         <attribute name="text:bullet-relative-size">
12621             <ref name="percent"/>
12622         </attribute>
12623     </optional>
12624 </define>
```

14.10.4 Стиль списка, отмечаемого изображениями

Элемент стиля списка, отмечаемого изображениями, определяет стиль списка, где пунктам списка предшествуют изображения. Элемент может быть [XLink] и содержится только в элементах стиля.

```
12625 <define name="text-list-style-content" combine="choice">
12626     <element name="text:list-level-style-image">
12627         <ref name="text-list-level-style-attr"/>
12628         <ref name="text-list-level-style-image-attr"/>
12629         <optional>
12630             <ref name="style-list-level-properties"/>
```

```
        </optional>
    </element>
</define>
```

С элементом `<text:list-level-style-image>` могут быть связаны следующие атрибуты:

- уровень (см. раздел 14.10.1);
- расположение изображения.

Дополнительные свойства форматирования могут содержаться в элементе `<style:list-level-properties>`. Для получения подробной информацией см. раздел 15.12.

Расположение изображения

Данные об изображении могут храниться одним из следующих способов (см. раздел 9.3.2).

- Данные об изображении располагаются во внешнем файле. Используйте описанный ниже атрибут `xlink:href` для указания адреса файла.
- Данные об изображении содержатся в элементе `<text:list-level-style-image>`. Элемент `<text:list-level-style-image>` должен включать элемент `<office:binary-data>` , содержащий данные об изображении в кодировке BASE64. В этом случае атрибут `xlink:href` не требуется.

```
<define name="text-list-level-style-image-attr" combine="interleave">
    <choice>
        <ref name="common-draw-data-attlist"/>
        <ref name="office-binary-data"/>
    </choice>
</define>
```

14.10.5 Пример стиля уровня списка

Пример — Стиль уровня списка:

```
<text:list-style style:name="List 1">
    <text:list-level-style-number text:level="1"
        fo:num-format="1"/>
    <text:list-level-style-bullet text:level="2"
        text:bullet-char="-"
        text:style-name="Bullet Char"/>
    <text:list-level-style-image text:level="3" xlink:href="bullet.gif">
        <style:list-level-properties fo:width=".27cm" fo:height=".27cm"
            style:vertical-pos="middle" style:vertical-rel="line"/>
    </text:list-level-style-image>
</text:list-style>
```

Следующее будет результатом реализации указанного примера:

1. Это первый пункт списка.

   Это продолжение первого пункта списка.

2. Это второй пункт списка. Он содержит неупорядоченный подсписок.
   - Это пункт подсписка.
   - Это пункт подсписка.
   - Это пункт подсписка.
     - Это пункт подподсписка.
     - Это пункт подподсписка.
3. Это третий пункт списка.

14.11 Стили структур

Стиль структуры — это стиль списка, применяемый ко всем заголовкам в текстовом документе, где стиль абзаца заголовка сам не определяет, какой стиль списка необходимо применить.

Способ представления стилей нумерации структуры в формате OpenDocument имеет много общего со способом представления стилей списка. Элемент `<text:outline-style>` содержит элементы, определяющие стиль каждого уровня структуры. Он может содержаться только внутри элемента `<office:styles`.

```
<define name="text-outline-style">
    <element name="text:outline-style">
        <oneOrMore>
            <ref name="text-outline-level-style"/>
        </oneOrMore>
    </element>
</define>
```

14.11.1 Стиль уровня структуры

Элемент `<text:outline-level-style>` определяет стиль каждого уровня структуры. Этот элемент содержится только в элементах `<text:outline-style>`.

```
<define name="text-outline-level-style">
    <element name="text:outline-level-style">
        <ref name="text-outline-level-style-attlist"/>
        <optional>
            <ref name="style-list-level-properties"/>
        </optional>
```

```
12653         <optional>
12654             <ref name="style-text-properties"/>
12655         </optional>
12656     </element>
12657 </define>
```

С элементом `<text:outline-level-style>` могут быть связаны следующие атрибуты:

- уровень;
- стиль текста;
- числовой формат;
- отображаемые уровни;
- начальное значение.

Дополнительные свойства форматирования могут содержаться в элементах `<style:list-level-properties>` и `<style:text-properties>`. Для получения дополнительной информации см. разделы 15.12 и 15.4.

Уровень

Атрибут `text:level` определяет уровень стиля структуры.

```
12658 <define name="text-outline-level-style-attlist" combine="interleave">
12659     <attribute name="text:level">
12660         <ref name="positiveInteger"/>
12661     </attribute>
12662 </define>
```

Стиль текста

Атрибут `text:style-name` определяет имя некоторого стиля, используемого для форматирования номера заголовка.

```
12663 <define name="text-outline-level-style-attlist" combine="interleave">
12664     <optional>
12665         <attribute name="text:style-name">
12666             <ref name="styleNameRef"/>
12667         </attribute>
12668     </optional>
12669 </define>
```

Числовой формат

Информация об атрибутах числового формата дана в разделе 14.10.2.

```
12670 <define name="text-outline-level-style-attlist" combine="interleave">
12671     <ref name="common-num-format-attlist"/>
12672     <ref name="common-num-format-prefix-suffix-attlist"/>
12673 </define>
```

Отображаемые уровни

Атрибут `text:display-levels` определяет число уровней, номера которых отображаются на текущем уровне (см. раздел 14.10.2).

```
<define name="text-outline-level-style-attlist" combine="interleave">
    <optional>
        <attribute name="text:display-levels" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

Начальное значение

Атрибут `text:start-value` определяет первый номер абзаца списка на текущем уровне.

```
<define name="text-outline-level-style-attlist" combine="interleave">
    <optional>
        <attribute name="text:start-value" a:defaultValue="1">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

14.12 Стили таблиц

14.12.1 Стили таблиц

Стили таблиц — это элементы `<style:style>` семейства `table`. Они могут быть использованы во всех типах приложений для определения свойств форматирования таблиц. Они поддерживают свойства таблиц, как описано в разделе 15.8.

```
<define name="style-style-content" combine="choice">
    <group>
        <attribute name="style:family">
            <value>table</value>
        </attribute>
        <optional>
            <ref name="style-table-properties"/>
        </optional>
    </group>
</define>
```

14.12.2 Стили столбцов таблицы

Стили столбцов таблицы — это элементы `<style:style>` семейства `table-column`. Они могут быть использованы во всех типах приложений для

определения свойств форматирования столбцов таблиц. Они поддерживают свойства столбцов таблиц, как описано в разделе 15.9.

```
<define name="style-style-content" combine="choice">
    <group>
        <attribute name="style:family">
            <value>table-column</value>
        </attribute>
        <optional>
            <ref name="style-table-column-properties"/>
        </optional>
    </group>
</define>
```

14.12.3 Стили строк таблицы

Стили строк таблицы — это элементы `<style:style>` семейства `table-row` . Они могут быть использованы во всех типах приложений для определения свойств форматирования строк таблиц. Они поддерживают свойства строк таблиц, как описано в разделе 15.10.

```
<define name="style-style-content" combine="choice">
    <group>
        <attribute name="style:family">
            <value>table-row</value>
        </attribute>
        <optional>
            <ref name="style-table-row-properties"/>
        </optional>
    </group>
</define>
```

14.12.4 Стили ячеек таблицы

Стили ячеек таблицы — это элементы `<style:style>` семейства `table-cell` . Они могут быть использованы во всех типах приложений для определения свойств форматирования ячеек таблиц. Они поддерживают как свойства таблиц, описанные в разделе 15.11, так и свойства текста, описанные в разделах 15.5 и 15.4.

```
<define name="style-style-content" combine="choice">
    <group>
        <attribute name="style:family">
            <value>table-cell</value>
        </attribute>
        <optional>
            <ref name="style-table-cell-properties"/>
        </optional>
        <optional>
            <ref name="style-paragraph-properties"/>
        </optional>
        <optional>
```

```
            <ref name="style-text-properties"/>
        </optional>
    </group>
</define>
```

14.13 Графические стили

14.13.1 Графические стили и стили презентации

Графические стили и стили презентации являются элементами `<style:style>`, принадлежащими семействам `graphic` или `presentation`. Графические стили семейства `graphic` могут встречаться во всех типах приложений, графические стили семейства `presentation` могут встречаться только в документах презентаций. Оба вида стилей поддерживают графические свойства, описанные в разделе 15.17. Они могут также содержать свойства абзацев и текста, описанные в разделах 15.5 и 15.4.

```
<define name="style-style-content" combine="choice">
    <group>
        <attribute name="style:family">
            <choice>
                <value>graphic</value>
                <value>presentation</value>
            </choice>
        </attribute>
        <optional>
            <ref name="style-graphic-properties"/>
        </optional>
        <optional>
            <ref name="style-paragraph-properties"/>
        </optional>
        <optional>
            <ref name="style-text-properties"/>
        </optional>
    </group>
</define>

<define name="style-graphic-properties">
    <element name="style:graphic-properties">
        <ref name="style-graphic-properties-content"/>
    </element>
</define>

<define name="style-graphic-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-graphic-properties-content-strict">
    <ref name="style-graphic-properties-attlist"/>
    <ref name="style-graphic-fill-properties-attlist"/>
    <ref name="style-graphic-properties-elements"/>
</define>

<define name=" style-graphic-properties-elements">
```

```
    <empty/>
</define>
```

14.13.2 Стили графических страниц

Стиль графической страницы — это элемент `<style:style>` из семейства `drawing-page`. В графических приложениях стили графических страниц могут использоваться для изменения фона графических страниц. Если фон задан с помощью стиля графической страницы, он перекроет фон мастер-страницы, связанной со страницей рисования, но не фигуры, находящиеся на мастер-странице. В приложениях-презентациях стиль графической страницы может содержать дополнительные свойства презентации, например продолжительность демонстрации страницы или эффекты затухания.

Свойства, которые могут использоваться в стиле графической страницы для изменения фона, описаны в разделе 15.14.

Свойства презентации, которые могут использоваться в стиле графической страницы, описаны в разделе 15.36.

```
<define name="style-style-content" combine="choice">
    <group>
        <attribute name="style:family">
            <value>drawing-page</value>
        </attribute>
        <optional>
            <ref name="style-drawing-page-properties"/>
        </optional>
    </group>
</define>

<define name="style-drawing-page-properties">
    <element name="style:drawing-page-properties">
        <ref name="style-drawing-page-properties-content"/>
    </element>
</define>

<define name="style-drawing-page-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-drawing-page-properties-content-strict">
    <ref name="style-graphic-fill-properties-attlist"/>
    <ref name="style-drawing-page-properties-attlist"/>
    <ref name="style-drawing-page-properties-elements"/>
</define>
```

14.14 Элементы расширенных графических стилей

Элементы, описанные в данном разделе, относятся к расширенному графическому стилю. Они не могут быть использованы в качестве автоматических

стилей, для этого они должны быть расположены в разделе `<office:styles>` документа. Как и все остальные элементы стиля, они доступны по уникальному имени. Для заполняемых графических объектов доступны следующие стили:

- градиент;
- SVG-градиент;
- штриховка;
- заполнение изображением;
- непрозрачность градиента;
- маркер;
- штрих контурной линии;
- разметка страницы презентации.

14.14.1 Градиент

Элемент `<draw:gradient>` определяет градиент для заполнения графического объекта. Градиенты не доступны в качестве автоматических стилей.

```
<define name="draw-gradient">
    <element name="draw:gradient">
        <ref name="common-draw-gradient-attlist"/>
        <ref name="draw-gradient-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом градиента могут быть связаны следующие атрибуты:

- имя;
- отображаемое имя;
- стиль градиента;
- центр градиента;
- цвета;
- яркость;
- угол;
- обрамление.

Имя

Атрибут `draw:name` однозначно идентифицирует градиент внутри элемента `<office:styles>`.

```
<define name="common-draw-gradient-attlist" combine="interleave">
    <optional>
        <attribute name="draw:name">
            <ref name="styleName"/>
        </attribute>
    </optional>
</define>
```

Отображаемое имя

Атрибут `draw:display-name` определяет имя градиента, каким оно будет отображено пользователю. В отличие от имени стиля, такое имя может содержать произвольные символы. Если этот атрибут не определен, то отображаемое имя градиента будет таким же, как и имя стиля.

```
<define name="common-draw-gradient-attlist" combine="interleave">
    <optional>
        <attribute name="draw:display-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Стиль градиента

Атрибут `draw:style` определяет стиль градиента. Офисное приложение должно поддерживать такие стили градиента, как `linear`, `axial`, `radial`, `ellipsoid`, `square` и `rectangular`.

```
<define name="common-draw-gradient-attlist" combine="interleave">
    <attribute name="draw:style">
        <ref name="gradient-style"/>
    </attribute>
</define>
<define name="gradient-style">
    <choice>
        <value>linear</value>
        <value>axial</value>
        <value>radial</value>
        <value>ellipsoid</value>
        <value>square</value>
        <value>rectangular</value>
    </choice>
</define>
```

Центр градиента

Если стиль градиента `radial`, `ellipsoid`, `square` или `rectangular`, то атрибуты центра градиента `draw:cx` и `draw:cy` задают центр геометрического построения, применяемого к градиенту. Значения этих атрибутов всегда задаются в процентах.

```
12835 <define name="common-draw-gradient-attlist" combine="interleave">
12836     <optional>
12837         <attribute name="draw:cx">
12838             <ref name="percent"/>
12839         </attribute>
12840     </optional>
12841     <optional>
12842         <attribute name="draw:cy">
12843             <ref name="percent"/>
12844         </attribute>
12845     </optional>
12846 </define>
```

Цвета

Градиент представляет собой плавный переход между начальным и конечным значениями цветов, которые можно задать с помощью атрибутов `draw:start-color` и `draw:end-color`.

```
12847 <define name="draw-gradient-attlist" combine="interleave">
12848     <optional>
12849         <attribute name="draw:start-color">
12850             <ref name="color"/>
12851         </attribute>
12852     </optional>
12853     <optional>
12854         <attribute name="draw:end-color">
12855             <ref name="color"/>
12856         </attribute>
12857     </optional>
12858 </define>
```

Яркость

Атрибуты `draw:start-intensity` и `draw:end-intensity` определяют яркость начального и конечного цвета градиента в процентах. Эти атрибуты являются необязательными. Если атрибуты не указаны, цвета будут использоваться так, как есть, что соответствует 100 % яркости.

```
12859 <define name="draw-gradient-attlist" combine="interleave">
12860     <optional>
12861         <attribute name="draw:start-intensity">
12862             <ref name="percent"/>
12863         </attribute>
12864     </optional>
12865     <optional>
12866         <attribute name="draw:end-intensity">
12867             <ref name="percent"/>
12868         </attribute>
12869     </optional>
12870 </define>
```

Угол

Атрибут draw:angle определяет угол поворота оси, на которой будет создаваться плавный переход цвета градиента. Этот атрибут игнорируется для градиентов радиального стиля.

```
12871 <define name="common-draw-gradient-attlist" combine="interleave">
12872     <optional>
12873         <attribute name="draw:angle">
12874             <ref name="integer"/>
12875         </attribute>
12876     </optional>
12877 </define>
```

Обрамление

В зависимости от стиля градиента, атрибут draw:border задает значение в процентах, которое используется для масштабирования границы, заполненной только начальным цветом либо только конечным цветом градиента.

Например, данный атрибут, равный 10 %, означает, что первые 10 % градиента полностью окрашены в начальный цвет, а оставшиеся 90 % будут плавно распределены между начальным и конечным значениями цвета.

```
12878 <define name="common-draw-gradient-attlist" combine="interleave">
12879     <optional>
12880         <attribute name="draw:border">
12881             <ref name="percent"/>
12882         </attribute>
12883     </optional>
12884 </define>
```

14.14.2 SVG-градиенты

В дополнение к градиентам, описанным в разделе 14.14.1, градиент может быть определен в элементах градиентов SVG <linearGradient> и <radialGradient> как указано в § 13.2 [SVG].

Следующие правила применяются к SVG градиентам, если они используются в документах формата OpenDocument:

- градиенты должны иметь название, что определяется атрибутом draw:name;
- для <linearGradient> будут вычисляться только атрибуты gradientTransform, x1, y1, x2, y2 и spreadMethod;

- для <radialGradient> будут вычисляться только атрибуты gradientTransform, cx, cy, r, fx, fy и spreadMethod;
- градиент будет рассчитываться как обладающий свойством gradientUnits из objectBoundingBox, вне зависимости от фактического значения атрибута;
- единственный дочерний элемент, который вычисляется, — это элемент <stop>;
- для <stop> будут вычисляться только атрибуты offset, stop-color и stop-opacity.

```
<define name="svg-linearGradient">
    <element name="svg:linearGradient">
        <ref name="common-svg-gradient-attlist"/>
        <optional>
            <attribute name="svg:x1" a:defaultValue="0%">
                <choice>
                    <ref name="coordinate"/>
                    <ref name="percent"/>
                </choice>
            </attribute>
        </optional>
        <optional>
            <attribute name="svg:y1" a:defaultValue="0%">
                <choice>
                    <ref name="coordinate"/>
                    <ref name="percent"/>
                </choice>
            </attribute>
        </optional>
        <optional>
            <attribute name="svg:x2" a:defaultValue="100%">
                <choice>
                    <ref name="coordinate"/>
                    <ref name="percent"/>
                </choice>
            </attribute>
        </optional>
        <optional>
            <attribute name="svg:y2" a:defaultValue="100%">
                <choice>
                    <ref name="coordinate"/>
                    <ref name="percent"/>
                </choice>
            </attribute>
        </optional>
        <zeroOrMore>
            <ref name="svg-stop"/>
        </zeroOrMore>
    </element>
</define>

<define name="svg-radialGradient">
    <element name="svg:radialGradient">
        <ref name="common-svg-gradient-attlist"/>
        <optional>
```

```
12930            <attribute name="svg:cx" a:defaultValue="50%">
12931                <choice>
12932                    <ref name="coordinate"/>
12933                    <ref name="percent"/>
12934                </choice>
12935            </attribute>
12936        </optional>
12937        <optional>
12938            <attribute name="svg:cy" a:defaultValue="50%">
12939                <choice>
12940                    <ref name="coordinate"/>
12941                    <ref name="percent"/>
12942                </choice>
12943            </attribute>
12944        </optional>
12945        <optional>
12946            <attribute name="svg:r" a:defaultValue="50%">
12947                <choice>
12948                    <ref name="coordinate"/>
12949                    <ref name="percent"/>
12950                </choice>
12951            </attribute>
12952        </optional>
12953        <optional>
12954            <attribute name="svg:fx">
12955                <choice>
12956                    <ref name="coordinate"/>
12957                    <ref name="percent"/>
12958                </choice>
12959            </attribute>
12960        </optional>
12961        <optional>
12962            <attribute name="svg:fy">
12963                <choice>
12964                    <ref name="coordinate"/>
12965                    <ref name="percent"/>
12966                </choice>
12967            </attribute>
12968        </optional>
12969        <zeroOrMore>
12970            <ref name="svg-stop"/>
12971        </zeroOrMore>
12972    </element>
12973 </define>
12974
12975 <define name="svg-stop">
12976     <element name="svg:stop">
12977         <attribute name="svg:offset">
12978             <choice>
12979                 <ref name="double"/>
12980                 <ref name="percent"/>
12981             </choice>
12982         </attribute>
12983         <optional>
12984             <attribute name="svg:stop-color">
12985                 <ref name="color"/>
12986             </attribute>
12987         </optional>
12988         <optional>
12989             <attribute name="svg:stop-opacity">
12990                 <ref name="double"/>
12991             </attribute>
12992         </optional>
```

```
12993     </element>
12994 </define>
12995
12996 <define name="common-svg-gradient-attlist" combine="interleave">
12997     <optional>
12998         <attribute name="svg:gradientUnits" a:defaultValue="objectBoundingBox">
12999             <value>objectBoundingBox</value>
13000         </attribute>
13001     </optional>
13002     <optional>
13003         <attribute name="svg:gradientTransform">
13004             <ref name="string"/>
13005         </attribute>
13006     </optional>
13007     <optional>
13008         <attribute name="svg:spreadMethod" a:defaultValue="pad">
13009             <choice>
13010                 <value>pad</value>
13011                 <value>reflect</value>
13012                 <value>repeat</value>
13013             </choice>
13014         </attribute>
13015     </optional>
13016 </define>
```

Имя

Атрибут `draw:name` однозначно идентифицирует градиент внутри элемента `<office:styles>`. Как и элементы `<draw:gradient>`, SVG-градиенты доступны по этому имени с использованием атрибута `draw:fill-gradient-name` в графическом стиле. SVG-градиенты не могут быть доступны по атрибуту `draw:opacity-name`. Результат обращения к SVG-градиенту с атрибутом `draw:fill-gradient-name` и непрозрачному градиенту с атрибутом `draw:opacity-name` одновременно, не определен.

```
13017 <define name="common-svg-gradient-attlist" combine="interleave">
13018     <attribute name="draw:name">
13019         <ref name="styleName"/>
13020     </attribute>
13021 </define>
```

Отображаемое имя

Атрибут `draw:display-name` определяет имя градиента, каким оно будет показано пользователю. В отличие от имени стиля, такое имя может содержать произвольные символы. Если этот атрибут не определен, то отображаемое имя градиента будет таким же, как и имя стиля.

```
13022 <define name="common-svg-gradient-attlist" combine="interleave">
13023     <optional>
13024         <attribute name="draw:display-name">
```

```
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

14.14.3 Штриховка

Элемент `<draw:hatch>` определяет штриховку для заполняемых графических объектов. Штриховка является простым шаблоном, заполняющим выделенную область повторяющимися прямыми линиями. Штриховка недоступна в качестве автоматических стилей.

```
<define name="draw-hatch">
    <element name="draw:hatch">
        <ref name="draw-hatch-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом штриховки могут быть связаны следующие атрибуты:

- имя;
- отображаемое имя;
- стиль;
- цвет;
- расстояние;
- угол;
- фон.

Имя

Атрибут `draw:name` однозначно идентифицирует элемент штриха внутри элемента `<office:styles>`.

```
<define name="draw-hatch-attlist" combine="interleave">
    <attribute name="draw:name">
        <ref name="styleName"/>
    </attribute>
</define>
```

Отображаемое имя

Атрибут `draw:display-name` определяет имя градиента, каким оно будет показано пользователю. В отличие от имени стиля, такое имя может содержать произвольные символы. Если этот атрибут не определен, то отображаемое имя градиента будет таким же, как и имя стиля.

```
13040 <define name="draw-hatch-attlist" combine="interleave">
13041     <optional>
13042         <attribute name="draw:display-name">
13043             <ref name="string"/>
13044         </attribute>
13045     </optional>
13046 </define>
```

Стиль

Атрибут `draw:style` определяет стиль штриха.

Стиль штриха может быть одним из трех: одинарный, двойной, тройной.

```
13047 <define name="draw-hatch-attlist" combine="interleave">
13048     <attribute name="draw:style">
13049         <choice>
13050             <value>single</value>
13051             <value>double</value>
13052             <value>triple</value>
13053         </choice>
13054     </attribute>
13055 </define>
```

Цвет

Атрибут `draw:color` определяет цвет линий штриха.

```
13056 <define name="draw-hatch-attlist" combine="interleave">
13057     <optional>
13058         <attribute name="draw:color">
13059             <ref name="color"/>
13060         </attribute>
13061     </optional>
13062 </define>
```

Расстояние

Атрибут `draw:distance` определяет расстояние между двумя линиями штриха.

```
13063 <define name="draw-hatch-attlist" combine="interleave">
13064     <optional>
13065         <attribute name="draw:distance">
13066             <ref name="length"/>
13067         </attribute>
13068     </optional>
13069 </define>
```

Угол

Атрибут `draw:rotation` определяет угол поворота линий штриха.

```
13070 <define name="draw-hatch-attlist" combine="interleave">
13071     <optional>
13072         <attribute name="draw:rotation">
```

```
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

14.14.4 Заполнение изображением

Элемент `<draw:fill-image>` определяет путь к растровому ресурсу, например файлу PNG. Этот элемент подчиняется спецификации XLink. Заполнение изображением недоступно в качестве автоматических стилей.

```
<define name="draw-fill-image">
    <element name="draw:fill-image">
        <ref name="draw-fill-image-attlist"/>
        <attribute name="xlink:href">
            <ref name="anyURI"/>
        </attribute>
        <optional>
            <attribute name="xlink:type" a:defaultValue="simple">
                <choice>
                    <value>simple</value>
                </choice>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:show" a:defaultValue="embed">
                <choice>
                    <value>embed</value>
                </choice>
            </attribute>
        </optional>
        <optional>
            <attribute name="xlink:actuate" a:defaultValue="onLoad">
                <choice>
                    <value>onLoad</value>
                </choice>
            </attribute>
        </optional>
        <empty/>
    </element>
</define>
```

С элементом заполнения изображением могут быть связаны следующие атрибуты:

- имя;
- отображаемое имя;
- размер.

Имя

Атрибут `draw:name` однозначно идентифицирует заполнение изображения внутри `<office:styles>` элемента.

```
13107 <define name="draw-fill-image-attlist" combine="interleave">
13108     <attribute name="draw:name">
13109         <ref name="styleName"/>
13110     </attribute>
13111 </define>
```

Отображаемое имя

Атрибут `draw:display-name` определяет имя заполняющегося изображения, каким оно отобразится в пользовательском интерфейсе. В отличие от самого имени стиля, такое имя может содержать произвольные символы. Если этот атрибут не определен, то отображаемое имя соответствует имени стиля.

```
13112 <define name="draw-fill-image-attlist" combine="interleave">
13113     <optional>
13114         <attribute name="draw:display-name">
13115             <ref name="string"/>
13116         </attribute>
13117     </optional>
13118 </define>
```

Размер

Дополнительные атрибуты `svg:width` и `svg:height` определяют размер указанного изображения. Эти значения необязательны и перекрываются физическим размером указанного ресурса изображения. Они могут быть использованы для получения физического размера изображения до того, как будут загружены.

```
13119 <define name="draw-fill-image-attlist" combine="interleave">
13120     <optional>
13121         <attribute name="svg:width">
13122             <ref name="length"/>
13123         </attribute>
13124     </optional>
13125     <optional>
13126         <attribute name="svg:height">
13127             <ref name="length"/>
13128         </attribute>
13129     </optional>
13130 </define>
```

14.14.5 Непрозрачность градиента

Элемент `<draw:opacity>` определяет уровень непрозрачности градиента для графического объекта. Уровень непрозрачности градиента работает подобно градиенту, за исключением того, что уровень непрозрачности интерполирован

вместо цвета. Уровень непрозрачности градиентов недоступен в качестве автоматических стилей.

```
<define name="draw-opacity">
    <element name="draw:opacity">
        <ref name="common-draw-gradient-attlist"/>
        <ref name="draw-opacity-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<draw:opacity>` могут быть связаны следующие атрибуты:

- имя, отображаемое имя, стиль, центр непрозрачности, угол, обрамление (см. раздел 14.14.1);
- уровень непрозрачности.

Уровень непрозрачности

Непрозрачность значений интерполируется между началом и концом.

Значения атрибутов `draw:start` и `draw:end` задаются в процентах, при этом 0 % — полная прозрачность, 100 % — полная непрозрачность.

```
<define name="draw-opacity-attlist" combine="interleave">
    <optional>
        <attribute name="draw:start">
            <ref name="percent"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="draw:end">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

14.14.6 Маркер

Элемент `<draw:marker>` определяет маркер, который используется для прорисовки начальных и конечных точек штрихов. Маркер недоступен в качестве автоматических стилей.

```
<define name="draw-marker">
    <element name="draw:marker">
        <ref name="draw-marker-attlist"/>
        <ref name="common-draw-viewbox-attlist"/>
        <ref name="common-draw-path-data-attlist"/>
        <empty/>
    </element>
</define>
```

Для получения информации относительно атрибутов данных траектории и области отображения, которые могут быть связаны с элементом `<draw:marker>`, см. разделы 9.2.4 и 9.2.15.

Имя

Атрибут `draw:name` однозначно идентифицирует заполнение изображения внутри `<office:styles>` элемента.

```
<define name="draw-marker-attlist" combine="interleave">
    <attribute name="draw:name">
        <ref name="styleName"/>
    </attribute>
</define>
```

Отображаемое имя

Атрибут `draw:display-name` определяет имя маркера, каким оно отобразится в пользовательском интерфейсе. В отличие от самого имени стиля, такое имя может содержать произвольные символы. Если этот атрибут не определен, то отображаемое имя соответствует имени стиля.

```
<define name="draw-marker-attlist" combine="interleave">
    <optional>
        <attribute name="draw:display-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

14.14.7 Штрих контурной линии

Элемент контурной линии `<draw:stroke-dash>` определяет стиль контурной линии, которая используется для задания штрихов формы. Штрихи контурной линии недоступны в качестве автоматических стилей.

```
<define name="draw-stroke-dash">
    <element name="draw:stroke-dash">
        <ref name="draw-stroke-dash-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<draw:stroke-dash>` могут быть связаны следующие атрибуты:

- имя;
- отображаемое имя;

- стиль;
- точки;
- интервал.

Имя

Атрибут `draw:name` однозначно идентифицирует контурную линию внутри `<office:styles>` элемента.

```
<define name="draw-stroke-dash-attlist" combine="interleave">
    <attribute name="draw:name">
        <ref name="styleName"/>
    </attribute>
</define>
```

Отображаемое имя

Атрибут `draw:display-name` задает имя контурной линии, каким оно отобразится в пользовательском интерфейсе. В отличии от самого имени стиля, такое имя может содержать произвольные символы. Если этот атрибут не определен, то отображаемое имя соответствует имени стиля.

```
<define name="draw-stroke-dash-attlist" combine="interleave">
    <optional>
        <attribute name="draw:display-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Стиль

Атрибут `draw:style` определяет, являются ли точки контурной линии круглыми или прямоугольными.

```
<define name="draw-stroke-dash-attlist" combine="interleave">
    <optional>
        <attribute name="draw:style">
            <choice>
                <value>rect</value>
                <value>round</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Точки

Каждая из пар атрибутов draw:dots1, draw:dots1-length и draw:dots2, draw:dots2-length определяет повторяющуюся последовательность точек, которая используется для отделения контурной линии. Обе последовательности используются чередуясь. Атрибуты draw:dots1 и draw:dots2 задают число точек для прорисовки обеих последовательностей, атрибуты draw:dots1-length и draw:dots2-length задают длину для каждой точки.

```
13198 <define name="draw-stroke-dash-attlist" combine="interleave">
13199     <optional>
13200         <attribute name="draw:dots1">
13201             <ref name="integer"/>
13202         </attribute>
13203     </optional>
13204     <optional>
13205         <attribute name="draw:dots1-length">
13206             <ref name="length"/>
13207         </attribute>
13208     </optional>
13209     <optional>
13210         <attribute name="draw:dots2">
13211             <ref name="integer"/>
13212         </attribute>
13213     </optional>
13214     <optional>
13215         <attribute name="draw:dots2-length">
13216             <ref name="length"/>
13217         </attribute>
13218     </optional>
13219 </define>
```

Интервал

Атрибут draw:distance определяет интервал между точками контурной линии.

```
13220 <define name="draw-stroke-dash-attlist" combine="interleave">
13221     <optional>
13222         <attribute name="draw:distance">
13223             <ref name="length"/>
13224         </attribute>
13225     </optional>
13226 </define>
```

14.15 Разметка страницы презентации

Элемент <style:presentation-page-layout> является контейнером для полей для заполнения, который определяет набор пустых объектов презентации, например название контура. Эти поля для заполнения используются как шаблоны для создания новых объектов презентации и для фиксации размера и позиции

объекта, если разметка страницы презентации для графической страницы изменена.

Элемент `<style:presentation-page-layout>` имеет атрибут `style:name`. Он определяет имя макета страницы. Если графическая страница была создана с использованием макета страницы презентации, то имя макета будет содержаться в атрибутах графической страницы `presentation:presentation-page-layout-name`. Необязательный атрибут `style:display-name` задает имя макета страницы презентации, каким оно отобразится в пользовательском интерфейсе.

```
<define name="style-presentation-page-layout">
    <element name="style:presentation-page-layout">
        <attribute name="style:name">
            <ref name="styleName"/>
        </attribute>
        <optional>
            <attribute name="style:display-name">
                <ref name="string"/>
            </attribute>
        </optional>
        <zeroOrMore>
            <ref name="presentation-placeholder"/>
        </zeroOrMore>
    </element>
</define>
```

14.15.1 Поле для заполнения в презентации

Элемент `<presentation:placeholder>` определяет поле для заполнения для объектов презентации, например заголовка или структуры.

Элемент имеет следующие атрибуты:

- `object` — задает вид объекта для элемента поле для заполнения, значение идентично одному из значений атрибутов `presentation:class` фигур презентации (см. раздел 9.6);
- `svg:x`, `svg:y`, `svg:width`, `svg:height` — положение и размер атрибутов указаны в разделе 9.2.15, за исключением значений в процентах, допустимых для полей для заполнения.

```
<define name="presentation-placeholder">
    <element name="presentation:placeholder">
        <attribute name="presentation:object">
            <ref name="presentation-classes"/>
        </attribute>
        <attribute name="svg:x">
            <choice>
                <ref name="coordinate"/>
```

```
13250                <ref name="percent"/>
13251            </choice>
13252        </attribute>
13253        <attribute name="svg:y">
13254            <choice>
13255                <ref name="coordinate"/>
13256                <ref name="percent"/>
13257            </choice>
13258        </attribute>
13259        <attribute name="svg:width">
13260            <choice>
13261                <ref name="length"/>
13262                <ref name="percent"/>
13263            </choice>
13264        </attribute>
13265        <attribute name="svg:height">
13266            <choice>
13267                <ref name="length"/>
13268                <ref name="percent"/>
13269            </choice>
13270        </attribute>
13271        <empty/>
13272    </element>
13273 </define>
```

14.16 Стили диаграмм

Элементы стиля диаграммы `<style:style>` принадлежат семейству `chart`. Они могут использоваться в пределах документов диаграммы, для определения свойств форматирования диаграммы, а также для некоторых объектов в пределах диаграммы. Они поддерживают свойства диаграммы, которые описаны в разделе 15.29, также графические свойства, свойства абзаца, текстовые свойства, описанные в разделах 15.17, 15.5 и 15.4.

```
13274 <define name="style-style-content" combine="choice">
13275     <group>
13276         <attribute name="style:family">
13277             <value>chart</value>
13278         </attribute>
13279         <optional>
13280             <ref name="style-chart-properties"/>
13281         </optional>
13282         <optional>
13283             <ref name="style-graphic-properties"/>
13284         </optional>
13285         <optional>
13286             <ref name="style-paragraph-properties"/>
13287         </optional>
13288         <optional>
13289             <ref name="style-text-properties"/>
13290         </optional>
13291     </group>
13292 </define>
```

15 Свойства форматирования

Документ может содержать несколько элементов стиля. Чтобы задать единый набор свойств форматирования, все свойства форматирования содержатся в элементах свойств форматирования, которые включены как дочерние элементы любого элемента стиля. Эти контейнерные элементы обладают двумя следующими важными преимуществами.

- К свойствам форматирования можно обращаться как к таблицам стилей [CSS2] или [XSLT] независимо от типа стиля.
- Стили содержат дополнительную информацию, которая не является свойством форматирования, например, имя стиля и родительский стиль. Хорошей практикой считается отделение этого типа информации.

Существуют следующие элементы свойств форматирования:

- `<style:page-layout-properties>` — для свойств разметки страницы;
- `<style:header-footer-properties>` — для свойств верхнего и нижнего колонтитулов;
- `<style:text-properties>` — для свойств текста;
- `<style:paragraph-properties>` — для свойств абзацев;
- `<style:section-properties>` — для свойств разделов текста;
- `<style:ruby-properties>` — для свойств транскрипции;
- `<style:list-level-properties>` — для свойств списков;
- `<style:table-properties>` — для свойств таблиц;
- `<style:table-column-properties>` — для свойств столбцов таблиц;
- `<style:table-row-properties>` — для свойств строк таблиц;
- `<style:table-cell-properties>` — для свойств ячеек таблиц;
- `<style:graphic-properties>` — для свойств графических объектов.

15.1 Простые и составные свойства форматирования

15.1.1 Простые свойства форматирования

Большинство свойств форматирования являются простыми и могут быть представлены как атрибуты свойств форматирования*. Для представления свойств форматирования, где возможно, используются [XSL] атрибуты или атрибуты из других спецификаций. В этой спецификации префикс пространства имен `fo` используется для свойств XSL, являющихся частью пространства имен XSL.

Очень часто в офисных приложениях свойства форматирования не могут быть определены независимо от других свойств форматирования. В этом случае, если некоторые из требуемых свойств пропущены, приложения присваивают приемлемые значения по умолчанию.

Пример — Простые свойства стиля.

Этот пример показывает контейнер свойств форматирования, который определяет верхнее поле абзаца, равное 1 см, и нижнее поле 0.5 см:

```
<style:paragraph-properties fo:margin-left="1cm" fo:margin-bottom=".5cm"/>
```

15.1.2 Составные свойства форматирования

Если свойство форматирования слишком сложно для представления атрибутами XML, оно представляется элементом XML. Каждое такое свойство представляется собственным типом элемента.

Пример — Составные свойства форматирования.

Ниже приведен пример контейнера свойства форматирования, который определяет верхнее и нижнее поле позиции табуляции соответственно 2 см и 4 см:

```
<style:paragraph-properties>
    <style:tab-stops>
        <style:tab-stop style:position="2cm"/>
        <style:tab-stop style:position="4cm"/>
    </style:tab-stops>
</style:paragraph-properties>
```

*В данной главе на равных правах используются термины «свойство» и «атрибут», что соответствует тексту оригинала ISO/IEC 26300:2006 «Information technology — Open Document Format for Office Applications (OpenDocument) v1.0». С содержательной точки зрения — это свойство (форматирования), представлением которого в формате ODF является соответствующий атрибут.

15.1.3 Правила обработки свойств форматирования

В структуре OpenDocument различные элементы `<style:*-properties>` могут содержать предопределенные атрибуты и элементы форматирования, а также особые атрибуты и элементы форматирования. Предопределенные элементы и атрибуты имеют определенную семантику и описаны в этом разделе.

Особые атрибуты и элементы — произвольные атрибуты и элементы внутри элементов `<style:*-properties>` . Их семантика не определена в данной спецификации.

Соответствующим приложениям **следует** сохранять предопределенные и особые атрибуты и элементы в процессе редактирования документа.

```
13293 <define name="style-properties-content">
13294     <ref name="anyAttListOrElements"/>
13295 </define>
```

15.2 Свойства форматирования для разметки страницы

Свойства, описанные в этом разделе, могут содержаться внутри стиля разметки страницы (см. раздел 14.3). Они содержатся в элементе `<style:page-layout-properties>`:

- размер страницы;
- формат номера страницы;
- лоток для бумаги;
- ориентация печати;
- поля;
- обрамление;
- толщина линии обрамления;
- заполнение;
- тень;
- фон;
- колонки;
- приводка;
- печать;
- порядок страниц при печати;
- нумерация первой страницы;
- масштаб;

- выравнивание таблицы;
- максимальная высота сноски;
- разделитель сноски.

```
<define name="style-page-layout-properties">
    <element name="style:page-layout-properties">
        <ref name="style-page-layout-properties-content"/>
    </element>
</define>

<define name="style-page-layout-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-page-layout-properties-content-strict">
    <ref name="style-page-layout-properties-attlist"/>
    <ref name="style-page-layout-properties-elements"/>
</define>
```

15.2.1 Размер страницы

Атрибуты `fo:page-width` и `fo:page-height` определяют физический размер страницы.

Атрибут `fo:page-width` должен соответствовать ориентации страницы. Например, если страница имеет книжную ориентацию, то атрибут `fo:page-width` определят ширину более короткой стороны страницы. Если страница имеет альбомную ориентацию, то атрибут `fo:page-width` определяет ширину более длинной стороны страницы.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:page-width">
            <ref name="length"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:page-height">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.2.2 Формат номера страницы

Атрибуты `style:num-format`, `style:num-prefix` и `style:num-suffix` определяют формат номера по умолчанию для стиля страницы, который используется для отображения номеров страниц внутри верхних и нижних колонтитулов Для получения дополнительной информации относительно атрибутов формата номера см. раздел 12.2.

Атрибут `style:num-format` может быть пустым. В этом случае по умолчанию номер страницы отображен не будет.

```
13322 <define name="style-page-layout-properties-attlist" combine="interleave">
13323     <optional>
13324         <ref name="common-num-format-attlist"/>
13325     </optional>
13326     <ref name="common-num-format-prefix-suffix-attlist"/>
13327 </define>
```

15.2.3 Лоток для бумаги

Атрибут `style:paper-tray-name` определяет используемый лоток для бумаги при печати документа. Названия, связанные с лотками принтера, зависят от принтера. Если значение этого атрибута `default`, будет использоваться лоток по умолчанию, определенный в параметрах конфигурации принтера.

```
13328 <define name="style-page-layout-properties-attlist" combine="interleave">
13329     <optional>
13330         <attribute name="style:paper-tray-name">
13331             <choice>
13332                 <value>default</value>
13333                 <ref name="string"/>
13334             </choice>
13335         </attribute>
13336     </optional>
13337 </define>
```

15.2.4 Ориентация печати

Атрибут `style:print-orientation` определяет ориентацию печатной страницы. Значение этого атрибута может быть `portrait` или `landscape`.

```
13338 <define name="style-page-layout-properties-attlist" combine="interleave">
13339     <optional>
13340         <attribute name="style:print-orientation">
13341             <choice>
13342                 <value>portrait</value>
13343                 <value>landscape</value>
13344             </choice>
13345         </attribute>
13346     </optional>
13347 </define>
```

15.2.5 Поля

Атрибуты полей `fo:margin`, `fo:margin-top`, `fo:margin-bottom`, `fo:margin-left` и `fo:margin-right` определяют размер полей страницы. Для

получения дополнительной информации по этим атрибутам см. разделы 15.5.17, 15.5.20 и 15.5.21. Значения в процентах не поддерживаются.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <ref name="common-horizontal-margin-attlist"/>
    <ref name="common-vertical-margin-attlist"/>
    <ref name="common-margin-attlist"/>
</define>
```

15.2.6 Обрамление

Атрибуты обрамления `fo:border`, `fo:border-top`, `fo:border-bottom`, `fo:border-left` и `fo:border-right` определяют свойства обрамления страницы. Для получения дополнительной информации по этим атрибутам см. раздел 15.5.25.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <ref name="common-border-attlist"/>
</define>
```

15.2.7 Толщина линии обрамления

Если страница содержит обрамления, то атрибуты толщины линии обрамления `style:border-line-width`, `style:border-line-width-top`, `style:border-line-width-bottom`, `style:border-line-width-left` и `style:border-line-width-right` определяют свойства линий обрамления страницы. Для получения дополнительной информации по этим атрибутам см. раздел 15.5.26.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <ref name="common-border-line-width-attlist"/>
</define>
```

15.2.8 Заполнение

Атрибуты заполнения `fo:padding`, `fo:padding-top`, `fo:padding-bottom`, `fo:padding-left` и `fo:padding-right` определяют свойства заполнения страницы. Для получения дополнительной информации по этим атрибутам см. раздел 15.5.27.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <ref name="common-padding-attlist"/>
```

```
13361 </define>
```

15.2.9 Тень

Атрибут тени `style:shadow` определяет тень страницы. Для получения дополнительной информации по этому атрибуту см. раздел 15.5.28.

```
13362 <define name="style-page-layout-properties-attlist" combine="interleave">
13363     <ref name="common-shadow-attlist"/>
13364 </define>
```

15.2.10 Фон

Атрибут фона `fo:background-color` и элемент фона `<style:background-image>` определяют свойства фона страницы. Для получения дополнительной информации по данному атрибуту и элементу см. разделы 15.5.23 и 15.5.24.

```
13365 <define name="style-page-layout-properties-attlist" combine="interleave">
13366     <ref name="common-background-color-attlist"/>
13367 </define>
13368 <define name="style-page-layout-properties-elements" combine="interleave">
13369     <ref name="style-background-image"/>
13370 </define>
```

15.2.11 Колонки

Элемент `<style:columns>` определяет, содержит ли страница колонки. Для получения дополнительной информации по этому элементу см. раздел 15.7.3.

```
13371 <define name="style-page-layout-properties-elements" combine="interleave">
13372     <ref name="style-columns"/>
13373 </define>
```

15.2.12 Приводка

Атрибут `style:register-truth-ref-style-name` ссылается на стиль абзаца. Междустрочное расстояние, определяемое стилем абзаца, используется в качестве ссылки на расстояние между строками для всех абзацев, для которых включена приводка.

```
13374 <define name="style-page-layout-properties-attlist" combine="interleave">
13375     <optional>
```

```
13376        <attribute name="style:register-truth-ref-style-name">
13377            <ref name="styleNameRef"/>
13378        </attribute>
13379    </optional>
13380 </define>
```

15.2.13 Печать

Атрибут `style:print` определяет, какие компоненты в электронной таблице печатать.

Значением этого атрибута может быть список следующих значений, разделенных пробелами:

- `headers`
- `grid`
- `annotations`
- `objects` (включая графические)
- `charts`
- `drawings`
- `formulas`
- `zero-values`

```
13381 <define name="style-page-layout-properties-attlist" combine="interleave">
13382     <optional>
13383         <attribute name="style:print">
13384             <list>
13385                 <zeroOrMore>
13386                     <choice>
13387                         <value>headers</value>
13388                         <value>grid</value>
13389                         <value>annotations</value>
13390                         <value>objects</value>
13391                         <value>charts</value>
13392                         <value>drawings</value>
13393                         <value>formulas</value>
13394                         <value>zero-values</value>
13395                     </choice>
13396                 </zeroOrMore>
13397             </list>
13398         </attribute>
13399     </optional>
13400 </define>
```

15.2.14 Порядок страниц при печати

Атрибут `style:print-page-order` определяет порядок, в котором данные в электронной таблице нумеруются и печатаются, когда данные не помещаются на одной печатной странице.

Значением этого атрибута может быть `ttb` или `ltr`. Используйте `ttb` для печати данных вертикально от левого столбца до нижней строки документа. Используйте `ltr` для печати данных горизонтально от верхней строки до правой колонки документа.

```
13401 <define name="style-page-layout-properties-attlist" combine="interleave">
13402     <optional>
13403         <attribute name="style:print-page-order">
13404             <choice>
13405                 <value>ttb</value>
13406                 <value>ltr</value>
13407             </choice>
13408         </attribute>
13409     </optional>
13410 </define>
```

15.2.15 Нумерация первой страницы

Атрибут `style:first-page-number` определяет номер первой страницы текстового или графического документа или первой страницы в таблице внутри документа электронной таблицы.

Значение этого атрибута может быть целым числом или `continue`. Если значение `continue`, то номер страницы будет номером предыдущей страницы, увеличенным на 1. По умолчанию номер первой страницы равен 1.

```
13411 <define name="style-page-layout-properties-attlist" combine="interleave">
13412     <optional>
13413         <attribute name="style:first-page-number">
13414             <choice>
13415                 <ref name="positiveInteger"/>
13416                 <value>continue</value>
13417             </choice>
13418         </attribute>
13419     </optional>
13420 </define>
```

15.2.16 Масштаб

Атрибуты `style:scale-to` и `style:scale-to-pages` определяют, как приложение должно масштабировать документы электронной таблицы для печати.

Атрибут `style:scale-to` определяет, что документ масштабирован на значение, указанное в процентах, где 100 % соответствует отсутствию масштабирования. При использовании этого атрибута все страницы при печати будут увеличены или уменьшены в размере.

Атрибут `style:scale-to-pages` определяет число страниц, на которых документ должен быть напечатан.

Если ни один из этих атрибутов не указан, то документ не будет масштабирован.

```
13421 <define name="style-page-layout-properties-attlist" combine="interleave">
13422     <optional>
13423         <attribute name="style:scale-to">
13424             <ref name="percent"/>
13425         </attribute>
13426     </optional>
13427     <optional>
13428         <attribute name="style:scale-to-pages">
13429             <ref name="positiveInteger"/>
13430         </attribute>
13431     </optional>
13432 </define>
```

15.2.17 Выравнивание таблицы

Атрибут `style:table-centering` определяет, каким образом приложение должно выровнять таблицы на странице. Этот атрибут относится только к электронным таблицам.

Значение этого атрибута может быть `horizontal`, `vertical`, `both` или `none`. Если этот атрибут не указан, то таблица не будет выровнена.

```
13433 <define name="style-page-layout-properties-attlist" combine="interleave">
13434     <optional>
13435         <attribute name="style:table-centering">
13436             <choice>
13437                 <value>horizontal</value>
13438                 <value>vertical</value>
13439                 <value>both</value>
13440                 <value>none</value>
13441             </choice>
13442         </attribute>
13443     </optional>
13444 </define>
```

15.2.18 Максимальная высота сноски

Атрибут `style:footnote-max-height` определяет максимальное пространство на странице, которое может занимать сноска. Значением этого

атрибута является величина, которая определяет максимальную высоту области сноски.

Если значение этого атрибута установлено в `0in`, то нет никаких ограничений на пространство, которое может занимать сноска.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:footnote-max-height">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.2.19 Режим записи

Атрибут `style:writing mode` определяет режим записи, который должен использоваться во всех абзацах, находящихся на странице (см. раздел 15.5.36). Значение `page` недопустимо при страничной разметке.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <ref name="common-writing-mode-attlist"/>
</define>
```

15.2.20 Разделитель сноски

Элемент `<style:footnote-sep>` описывает линию, которая отделяет область сноски от области основного текста на странице.

Элемент `<style:footnote-sep>` поддерживает следующие атрибуты:

- `style:width` — определяет ширину или толщину линии;
- `style:rel-width` — определяет длину линии в процентном отношении от области основного текста;
- `style:color` — определяет цвет линии;
- `style:adjustment` — определяет, как линия выровнена на странице, слева, справа или по центру;
- `style:distance-before-sep` — определяет пространство между областью основного текста и линией сноски;
- `style:distance-after-sep` — определяет расстояние между линией сноски и текстом сноски;
- `style:line-style` — определяет стиль линии.

```
<define name="style-page-layout-properties-elements" combine="interleave">
    <ref name="style-footnote-sep"/>
</define>
```

```
<define name="style-footnote-sep">
    <optional>
        <element name="style:footnote-sep">
            <ref name="style-footnote-sep-attlist"/>
            <empty/>
        </element>
    </optional>
</define>
<define name="style-footnote-sep-attlist" combine="interleave">
    <optional>
        <attribute name="style:width">
            <ref name="length"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:rel-width">
            <ref name="percent"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:color">
            <ref name="color"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:line-style">
            <ref name="lineStyle"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:adjustment" a:defaultValue="left">
            <choice>
                <value>left</value>
                <value>center</value>
                <value>right</value>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:distance-before-sep">
            <ref name="length"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:distance-after-sep">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.2.21 Сеточная разметка

Свойство `style:layout-grid-mode` включает азиатскую сеточную разметку. Оно имеет следующие значения.

- `none` — Отключает сеточную разметку.
- `lines` — При включенной сеточной разметке страница разделена на фиксированное количество строк. Точное число строк зависит от других

свойств сеточной разметки, описанных ниже. Между строк сеточной разметки нет никакого пространства. Сеточная разметка центрирована на странице.

- `both` — Подобно `lines`, за исключением того, что строки разделены на квадратные ячейки. Число ячеек в строке зависит от высоты строки, где высота строки — это сумма базовой высоты и высоты транскрипции, как определено ниже. В пределах разметки в ячейке отображается не более одного азиатского символа [UNICODE]. Азиатские символы, которые не помещаются в одну ячейку, отображаются по центру внутри необходимого количества ячеек. Неазиатский текст центрирован внутри необходимого количества ячеек.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:layout-grid-mode">
            <choice>
                <value>none</value>
                <value>line</value>
                <value>both</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.2.22 Базовая высота сеточной разметки

Атрибут `style:layout-grid-base-height` определяет высоту, необходимую в строках сеточной разметки для текста без транскрипции.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:layout-grid-base-height">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.2.23 Высота транскрипции сеточной разметки

Атрибут `style:layout-grid-ruby-height` определяет высоту, необходимую для транскрипции сеточной разметки.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:layout-grid-ruby-height">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.2.24 Строки сеточной разметки

Атрибут `style:layout-grid-lines` определяет количество строк на странице для сеточной разметки. Фактически количество отображенных строк может быть меньше указанного, если на странице недостаточно места для отображения количества строк с указанной высотой строки (т. е. суммы высот основного текста и транскрипции).

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:layout-grid-lines">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

15.2.25 Цвет линий сеточной разметки

Атрибут `style:layout-grid-color` определяет цвет линий сеточной разметки.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:layout-grid-color">
            <ref name="color"/>
        </attribute>
    </optional>
</define>
```

15.2.26 Положение транскрипции сеточной разметки

Атрибут `style:layout-grid-ruby-below` определяет, будет ли транскрипция отображена над или под основным текстом.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:layout-grid-ruby-below">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.2.27 Печать сеточной разметки

Атрибут `style:layout-grid-ruby-print` определяет, будут ли печататься линии разметки сетки строк.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:layout-grid-print">
            <ref name="boolean"/>
```

```
        </attribute>
    </optional>
</define>
```

15.2.28 Отображение разметки сетки

Атрибут `style:layout-grid-ruby-print` определяет, будут ли отображаться линии разметки сетки строк.

```
<define name="style-page-layout-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:layout-grid-display">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.3 Свойства форматирования верхнего и нижнего колонтитулов

Свойства, описанные в этом разделе, могут содержаться в элементах стилей верхнего и нижнего колонтитулов на странице (см. раздел 14.3) Они содержатся в элементе `<style:header-footer-properties>`.

Используются следующие атрибуты:

- фиксированная и минимальная высота (см. раздел 15.27);
- левые и правые поля (см. раздел 15.5.17);
- нижнее (только для верхних колонтитулов) и верхнее (только для нижних колонтитулов) поля (см. раздел 15.5.20);
- обрамление (см. разделы 15.5.25 и 15.5.26);
- тени (см. раздел 15.5.28);
- фоны (см. разделы 15.5.23 и 15.5.24);
- динамический интервал.

```
<define name="style-header-footer-properties">
    <element name="style:header-footer-properties">
        <ref name="style-header-footer-properties-content"/>
    </element>
</define>

<define name="style-header-footer-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-header-footer-properties-content-strict">
        <ref name="style-header-footer-properties-attlist"/>
        <ref name="style-header-footer-properties-elements"/>
</define>
```

15.3.1 Фиксированная и минимальная высота

Атрибуты свойств `svg:height` и `fo:min-height` определяют фиксированную или минимальную высоту для нижнего или верхнего колонтитулов.

```
<define name="style-header-footer-properties-attlist" combine="interleave">
    <optional>
        <attribute name="svg:height">
            <ref name="length"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:min-height">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.3.2 Поля

Атрибуты полей `fo:margin`, `fo:margin-top`, `fo:margin-bottom`, `fo:margin-left` и `fo:margin-right` определяют размеры полей для верхнего и нижнего колонтитулов. Для получения дополнительной информации по этим атрибутам см. разделы 15.5.17, 15.5.20 и 15.5.21. Значения в процентах не поддерживаются. Нижние поля поддерживаются только для верхних колонтитулов, верхние поля — только для нижних колонтитулов.

```
<define name="style-header-footer-properties-attlist" combine="interleave">
    <ref name="common-horizontal-margin-attlist"/>
    <ref name="common-vertical-margin-attlist"/>
    <ref name="common-margin-attlist"/>
</define>
```

15.3.3 Обрамление

Атрибуты обрамления `fo:border`, `fo:border-top`, `fo:border-bottom`, `fo:border-left` и `fo:border-right` определяют свойства обрамления верхних и нижних колонтитулов. Для получения дополнительной информации по этим атрибутам см. раздел 15.5.25.

```
<define name="style-header-footer-properties-attlist" combine="interleave">
    <ref name="common-border-attlist"/>
</define>
```

15.3.4 Толщина линии обрамления

Если страница содержит обрамления, то атрибуты толщины линии обрамления `style:border-line-width`, `style:border-line-width-top`, `style:border-line-width-bottom`, `style:border-line-width-left` и `style:border-line-width-right` определяют свойства линий обрамления для верхних и нижних колонтитулов. Для получения дополнительной информации по этим атрибутам см. раздел 15.5.26.

```
<define name="style-header-footer-properties-attlist" combine="interleave">
    <ref name="common-border-line-width-attlist"/>
</define>
```

15.3.5 Заполнение

Атрибуты заполнения `fo:padding`, `fo:padding-top`, `fo:padding-bottom`, `fo:padding-left` и `fo:padding-right` определяют свойства заполнения для верхних и нижних колонтитулов. Для получения дополнительной информации по этим атрибутам см. раздел 15.5.27.

```
<define name="style-header-footer-properties-attlist" combine="interleave">
    <ref name="common-padding-attlist"/>
</define>
```

15.3.6 Фон

Атрибут фона `fo:background-color` и элемент фона `<style:background-image>` определяют свойства фона для верхнего и нижнего колонтитулов. Для получения дополнительной информации по этому атрибуту и элементу см. разделы 15.5.23 и 15.5.24.

```
<define name="style-header-footer-properties-attlist" combine="interleave">
    <ref name="common-background-color-attlist"/>
</define>
<define name="style-header-footer-properties-elements" combine="interleave">
    <ref name="style-background-image"/>
</define>
```

15.3.7 Тень

Атрибут тени style:shadow определяет тень для верхних и нижних колонтитулов. Для получения дополнительной информации по этому атрибуту см. раздел 15.5.28.

```
<define name="style-header-footer-properties-attlist" combine="interleave">
    <ref name="common-shadow-attlist"/>
</define>
```

15.3.8 Динамический интервал

Свойство style:dynamic-spacing определяет, будут ли колонтитулы увеличиваться за счет расстояния между основным содержанием страницы и колонтитулами до того, как высота основного содержания страницы начнет уменьшаться. Если значение атрибута равно true, то колонтитулы будут увеличиваться за счет расстояния между колонтитулами и основным содержанием страницы.

```
<define name="style-header-footer-attlist" combine="interleave">
    <optional>
        <attribute name="style:dynamic-spacing">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.4 Свойства форматирования текста

Свойства, описанные в этом разделе, могут содержаться в пределах стилей текста (см. раздел 14.8.1), а также в пределах других стилей, как стили абзаца (см. раздел 14.8.2) или стили ячейки (см. раздел 14.12.4) Они содержатся в элементе <style:text-properties>.

```
<define name="style-text-properties">
    <element name="style:text-properties">
        <ref name="style-text-properties-content"/>
    </element>
</define>

<define name="style-text-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-text-properties-content-strict">
    <ref name="style-text-properties-attlist"/>
    <ref name="style-text-properties-elements"/>
</define>
```

```
<define name="style-text-properties-elements">
    <empty/>
</define>
```

15.4.1 Вид шрифта

Используйте свойство `fo:font-variant` для включения или выключения опции отображения текста прописными буквами (см. § 7.8.8 [XSL] для подробного ознакомления).

Для некоторых реализаций свойства `fo:font-variant` и `fo:text-transform` являются взаимоисключающими. Если оба свойства используются одновременно, то результат будет неопределенным, за исключением случаев, когда значение `fo:text-transform` равно `none` и значение `fo:font-variant` равно `normal`.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:font-variant">
            <ref name="fontVariant"/>
        </attribute>
    </optional>
</define>

<define name="fontVariant">
    <choice>
        <value>normal</value>
        <value>small-caps</value>
    </choice>
</define>
```

15.4.2 Преобразования текста

Используйте свойство `fo:text-transform` для описания преобразований текста к верхнему регистру, нижнему регистру или прописным буквам (см. § 7.16.6 [XSL] для подробного ознакомления).

Для некоторых реализаций свойства `fo:font-variant` и `fo:text-transform` являются взаимоисключающими. Если оба свойства заданы и используются одновременно, то результат будет неопределенным, за исключением случаев, когда значение `fo:text-transform` равно `none` и значение `fo:font-variant` равно `normal`.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:text-transform">
            <choice>
                <value>none</value>
```

```
13661                <value>lowercase</value>
13662                <value>uppercase</value>
13663                <value>capitalize</value>
13664            </choice>
13665        </attribute>
13666    </optional>
13667 </define>
```

15.4.3 Цвет

Используйте свойство `fo:color` для определения цвета переднего плана текста (см. § 7.17.1 [XSL] для подробного ознакомления).

```
13668 <define name="style-text-properties-attlist" combine="interleave">
13669     <optional>
13670             <attribute name="fo:color">
13671             <ref name="color"/>
13672         </attribute>
13673     </optional>
13674 </define>
```

15.4.4 Цвет шрифта окна

Используйте свойство `style:use-window-font-color`, чтобы определить, должен ли цвет переднего плана окна использоваться как цвет переднего плана для светлого фона и белый — для темного цвета фона.

```
13675 <define name="style-text-properties-attlist" combine="interleave">
13676     <optional>
13677         <attribute name="style:use-window-font-color">
13678             <ref name="boolean"/>
13679         </attribute>
13680     </optional>
13681 </define>
```

15.4.5 Структура текста

Используйте свойство `style:text-outline` чтобы определить, показывать ли структуру текста или текст непосредственно. Этот атрибут может иметь значения `true` или `false`.

```
13682 <define name="style-text-properties-attlist" combine="interleave">
13683     <optional>
13684         <attribute name="style:text-outline">
13685             <ref name="boolean"/>
13686         </attribute>
13687     </optional>
13688 </define>
```

15.4.6 Тип линии зачеркивания

Используйте свойство `style:text-line-through-type`, чтобы определить, зачеркнут ли текст, и если зачеркнут, то одиночной или двойной линией (см. раздел 15.4.28 для подробного ознакомления).

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-line-through-type">
            <ref name="lineType"/>
        </attribute>
    </optional>
</define>
```

15.4.7 Стиль линии зачеркивания

Используйте свойство `style:text-line-through-style`, чтобы определить, зачеркнут ли текст, и если зачеркнут, то как. Это свойство подобно свойству [CSS3Text] `text-line-style` за исключением того, что оно имеет дополнительное значение `long-dash` и не имеет значения `double`. Вместо этого атрибут `style:text:line-through-type` может использоваться для преобразования каждой линии в двойную (см. § 9.2 [CSS3Text] и раздел 15.4.29 для подробного ознакомления).

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-line-through-style">
            <ref name="lineStyle"/>
        </attribute>
    </optional>
</define>
```

15.4.8 Протяженность линии зачеркивания

Используйте свойство `style:text-line-through-width` для того, чтобы установить протяженность линии зачеркивания. Данное свойство очень похоже на свойство `[CSS3Text] text-line-through-width` за исключением того, что оно имеет дополнительное свойство `bold`. `bold` устанавливает протяженность линии, которая рассчитывается на основании размеров шрифтов, подобно `auto` ширине, однако шире, чем `auto` ширина (см. § 9.3 [CSS3Text] и раздел 15.4.30).

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-line-through-width">
            <ref name="lineWidth"/>
        </attribute>
```

```
    </optional>
</define>
```

15.4.9 Цвет зачеркивания

Используйте свойство `style:text-line-through-color` для определения цвета, используемого при зачеркивании текста. Значением этого свойства может быть либо `font-color`, либо `color`. Если значение равно `font-color`, тогда для зачеркивания используется текущий цвет текста.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-line-through-color">
            <choice>
                <value>font-color</value>
                <ref name="color"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.4.10 Текст зачеркивания

Атрибут `style:text-line-through-text` интерпретируется только в том случае, если значение атрибута `style:text-line-through-style` отлично от пустого. Если значение атрибута не пустое, то оно используется для зачеркивания вместо линии, однако, только в том случае, если зачеркивание поддерживается текстом. Если нет, то атрибут игнорируется, это означает, что только `style:text-line-through-style` будет интерпретирован. Если приложение поддерживает только зачеркивание одним символом, а `text-line-through-text` более одного символа, тогда для зачеркивания будет использоваться только первый символ `line-through-text`. Если приложения поддерживают зачеркивание только строго определенными символами (например "x" или "/"), то приложение должно использовать один из этих символов в том случае, когда `text-line-through-text` устанавливает неподдерживаемые символы. Иными словами, зачеркивание текстом имеет более высокий приоритет, чем зачеркивание линией, даже если текст зачеркивания не предназначен для использования приложением.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
```

```
13722         <attribute name="style:text-line-through-text">
13723             <ref name="string"/>
13724         </attribute>
13725     </optional>
13726 </define>
```

15.4.11 Стиль текста зачеркивания

Атрибут `style:text-line-through-text-style` определяет стиль текста, который применяется к символам зачеркивания. Он не применяется к линии зачеркивания. Если атрибут имеется в автоматическом стиле, то он может относиться либо к автоматическому стилю текста, либо к общему стилю. Если атрибут имеется в общем стиле, он может относиться только к нему.

```
13727 <define name="style-text-properties-attlist" combine="interleave">
13728     <optional>
13729         <attribute name="style:text-line-through-text-style">
13730             <ref name="styleNameRef"/>
13731         </attribute>
13732     </optional>
13733 </define>
```

15.4.12 Позиция текста

Используйте свойство форматирования `style:text-position` для того, чтобы определить находится ли текст ниже или выше базовой линии, а также чтобы определить относительную высоту шрифта для данного текста.

Этот атрибут может иметь одно или два значения.

Первое значение должно присутствовать и устанавливает вертикальную позицию текста в процентах от текущей высоты шрифта, либо может принимать одно из двух значений — `sub` или `super`. Отрицательное значение в процентах, а также значение `sub` помещают текст ниже базовой линии. Положительное значение в процентах, а также значение `super` помещают текст выше базовой линии. Если указаны значения `sub` или `super`, приложение может выбирать подходящее положение текста.

Второе значение необязательно и указывает высоту шрифта в процентах относительно текущей высоты шрифта. Если это значение не указано, используется подходящая высота шрифта. Хотя это значение и может менять высоту шрифта отображаемого текста, оно не меняет высоту шрифта, используемого для дополнительных вычислений.

```
13734 <define name="style-text-properties-attlist" combine="interleave">
```

```
13735     <optional>
13736         <attribute name="style:text-position">
13737             <list>
13738                 <choice>
13739                     <ref name="percent"/>
13740                     <value>super</value>
13741                     <value>sub</value>
13742                 </choice>
13743                 <optional>
13744                     <ref name="percent"/>
13745                 </optional>
13746             </list>
13747         </attribute>
13748     </optional>
13749 </define>
```

15.4.13 Название шрифта

Используйте свойства `style:font-name`, `style:font-name-asian` и `style:font-name-complex`, чтобы определить шрифт текста.

Значения этих атрибутов формируют имя шрифта, определяемого элементом `<style:font-face>` внутри элемента `<office:font-face-decls>`.

Атрибут `style:font-name-asian` применяется для [UNICODE]-символов восточно-азиатской системы письменности.

Атрибут `style:font-name-complex` применяется для [UNICODE]-символов сложных систем письменности.

Атрибут `style:font-name` применяется для остальных [UNICODE]-символов.

```
13750 <define name="style-text-properties-attlist" combine="interleave">
13751     <optional>
13752         <attribute name="style:font-name">
13753             <ref name="string"/>
13754         </attribute>
13755     </optional>
13756     <optional>
13757         <attribute name="style:font-name-asian">
13758             <ref name="string"/>
13759         </attribute>
13760     </optional>
13761     <optional>
13762         <attribute name="style:font-name-complex">
13763             <ref name="string"/>
13764         </attribute>
13765     </optional>
13766 </define>
```

15.4.14 Семейство шрифтов

Используйте свойства `fo:font-family`, `style:font-family-asian` и `style:font-family-complex`, чтобы определить семейство шрифтов для текста.

Эти атрибуты могут использоваться вместо атрибутов названия шрифта, чтобы отдельно установить свойства шрифта. Однако целесообразно использовать атрибуты `style:font-name`. Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности, см. раздел 15.4.13, а также см. § 7.8.2 [XSL].

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:font-family">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:font-family-asian">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:font-family-complex">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

15.4.15 Группа семейства шрифтов

Используйте свойства `style:font-family-generic`, `style:font-family-generic-asian` и `style:font-family-generic-complex`, чтобы определить группу семейства шрифтов.

Эти свойства игнорируются, если нет соответствующего свойства `fo:font-family`, привязанного к тому же элементу свойства.

Несмотря на то, что рекомендуется использовать атрибуты названия шрифтов (см. раздел 15.4.13), эти свойства могут использоваться для установки свойств шрифта вместо них.

Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности, см. раздел 15.4.13.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:font-family-generic">
            <ref name="fontFamilyGeneric"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:font-family-generic-asian">
            <ref name="fontFamilyGeneric"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:font-family-generic-complex">
            <ref name="fontFamilyGeneric"/>
        </attribute>
    </optional>
</define>

<define name="fontFamilyGeneric">
    <choice>
        <value>roman</value>
        <value>swiss</value>
        <value>modern</value>
        <value>decorative</value>
        <value>script</value>
        <value>system</value>
    </choice>
</define>
```

15.4.16 Стиль шрифта

Используйте свойства `style:font-style-name`, `style:font-style-name-asian` и `style:font-style-name-complex`, чтобы определить имя стиля шрифта.

Эти свойства игнорируются, если нет соответствующего свойства `fo:font-family`, привязанного к тому же элементу свойства.

Несмотря на то, что рекомендуется использовать атрибуты названия шрифтов (см. 15.4.13), эти свойства могут использоваться для установки свойств шрифта вместо них.

Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности, см. раздел 15.4.13.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:font-style-name">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:font-style-name-asian">
            <ref name="string"/>
```

```
13821         </attribute>
13822     </optional>
13823     <optional>
13824         <attribute name="style:font-style-name-complex">
13825             <ref name="string"/>
13826         </attribute>
13827     </optional>
13828 </define>
```

15.4.17 Шаг шрифта

Используйте свойства `style:font-pitch`, `style:font-pitch` и `style:font-pitch-complex`,чтобы определить, применяется ли постоянная или переменная ширина шрифта

Эти свойства игнорируются, если нет соответствующего свойства `fo:font-family`, привязанного к тому же элементу свойства.

Несмотря на то, что рекомендуется использовать атрибуты названия шрифтов (см. раздел 15.4.13), эти свойства могут использоваться для установки свойств шрифта вместо них.

Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности, см. раздел 15.4.13.

```
13829 <define name="style-text-properties-attlist" combine="interleave">
13830     <optional>
13831         <attribute name="style:font-pitch">
13832             <ref name="fontPitch"/>
13833         </attribute>
13834     </optional>
13835     <optional>
13836         <attribute name="style:font-pitch-asian">
13837             <ref name="fontPitch"/>
13838         </attribute>
13839     </optional>
13840     <optional>
13841         <attribute name="style:font-pitch-complex">
13842             <ref name="fontPitch"/>
13843         </attribute>
13844     </optional>
13845 </define>
13846
13847 <define name="fontPitch">
13848     <choice>
13849         <value>fixed</value>
13850         <value>variable</value>
13851     </choice>
13852 </define>
```

15.4.18 Набор символов шрифта

Используйте свойства `style:font-charset`, `style:font-charset-asian` и `style:font-charset-complex`, чтобы определить набор символов шрифта.

Значением этого атрибута может быть `x-symbol` либо кодировка символов в нотации, описанной в § 4.3.3 [XML1.0]. Если значение равно `x-symbol`, все символы, отображаемые с использованием данного шрифта должны находиться в интервале [UNICODE] символов от 0xf000 до 0xf0ff.

Эти свойства игнорируются, если нет соответствующего свойства `fo:font-family`, привязанного к тому же элементу свойства.

Несмотря на то, что рекомендуется использовать атрибуты названия шрифтов (см. раздел 15.4.13), эти свойства могут использоваться для установки свойств шрифта вместо них.

Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности, см. раздел 15.4.13.

```
13853 <define name="style-text-properties-attlist" combine="interleave">
13854     <optional>
13855         <attribute name="style:font-charset">
13856             <ref name="textEncoding"/>
13857         </attribute>
13858     </optional>
13859 </define>
13860
13861 <define name="textEncoding">
13862     <data type="string">
13863         <param name="pattern">[A-Za-z][A-Za-z0-9._\-]*</param>
13864     </data>
13865 </define>
```

15.4.19 Размер шрифта

Используйте свойства `fo:font-size`, `style:font-size-asian` и `style:font-size-complex`, чтобы определить размер шрифта.

Значение этого свойства — абсолютная величина или значение в процентах (см. § 8.8.4 [XSL]). В отличие от XSL, процентные значения могут использоваться только в обычных стилях и зависят от высоты шрифта родительского стиля в большей степени, чем от значения высоты шрифта, установленного соседним атрибутом. Абсолютные значения высоты шрифта, такие как `medium`, `large`, `x-`

large и так далее, и относительные значения высоты шрифта, такие как smaller и larger, не поддерживаются.

Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности, см. раздел 15.4.13.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:font-size">
            <choice>
                <ref name="positiveLength"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:font-size-asian">
            <choice>
                <ref name="positiveLength"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:font-size-complex">
            <choice>
                <ref name="positiveLength"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.4.20 Масштаб шрифта

Используйте свойства style:font-size-rel, style:font-size-rel-asian и style:font-size-rel-complex, чтобы определить масштаб шрифта.

Эти свойства определяют изменение масштаба шрифта в некоторых единицах, например +1pt, -3pt. Это не может использоваться с автоматическими стилями. Изменения размеров зависят от размеров шрифта, установленных в родительском стиле для данного стиля.

Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности, см. раздел 15.4.13.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:font-size-rel">
            <ref name="length"/>
        </attribute>
```

```
13897     </optional>
13898     <optional>
13899         <attribute name="style:font-size-rel-asian">
13900             <ref name="length"/>
13901         </attribute>
13902     </optional>
13903     <optional>
13904         <attribute name="style:font-size-rel-complex">
13905             <ref name="length"/>
13906         </attribute>
13907     </optional>
13908 </define>
```

15.4.21 Тип скрипта

Свойство `style:script-type` может использоваться для указания атрибутов, связанных с типом скрипта (например, `fo:font-family`, `style:font-family-asian`, `style:font-family-complex`), активных в настоящее время для некоторого текста. Атрибут должен вычисляться приложениями, которые не поддерживают типы скриптов, чтобы установить свойства, обусловленные типами скриптов. Приложение, поддерживающее типы скриптов, также может интерпретировать атрибут и переопределять тип, используемый для некоторых символов, но не обязательно.

Использование данного свойства упрощает преобразования из/в [CSS2]/[XSL] и другие форматы, не поддерживающие атрибутов, связанных с типами скриптов. Данное свойство может использоваться также для применения типов скриптов для [UNICODE] символов, там где приложение может выбрать различные типы скриптов.

Значениями данного свойства могут быть `latin`, `asian`, `complex` и `ignore`. Значение `ignore` может быть использовано только со стилями по умолчанию. Если оно установлено, атрибуты, связанные с типами скриптов, применяются ко всем типам скриптов. Это может, например, означать, что `fo:font-family` будет применен ко всем типам скриптов, также как `style:font-family-asian` или `style:font-family-complex`. Это упрощает сохранение документов из приложений, не поддерживающих тип скрипта.

```
13909 <define name="style-text-properties-attlist" combine="interleave">
13910     <optional>
13911         <attribute name="style:script-type">
13912             <choice>
13913                 <value>latin</value>
13914                 <value>asian</value>
13915                 <value>complex</value>
```

```
                <value>ignore</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.4.22 Межбуквенный интервал

Используйте свойство `fo:letter-spacing`, чтобы определить количество межбуквенных пробелов. Значением этого свойства может быть либо `normal`, либо конкретное значение (см. § 7.16.2 в [XSL]).

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:letter-spacing">
            <choice>
                <ref name="length"/>
                <value>normal</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.4.23 Язык

Используйте свойства `fo:language`, `fo:language-asian` и `fo:language-complex`, чтобы определить язык написания текста (см. § 7.9.2 в [XSL]).

Некоторые приложения игнорируют эти свойства в том случае, если они не устанавливаются вместе с соответствующим свойством `fo:country`.

Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности, см. раздел 15.4.13.

`fo:language`, `fo:language-asian` и `fo:language-complex`

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:language">
            <ref name="languageCode"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:language-asian">
            <ref name="languageCode"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:language-complex">
            <ref name="languageCode"/>
        </attribute>
    </optional>
</define>
```

15.4.24 Страна

Используйте свойства `fo:country`, `style:country-asian` и `style:country-complex`, чтобы определить страну текста (см. § 7.9.1 в [XSL]).

Некоторые приложения игнорируют данное свойство, если оно не задается вместе с соответствующим свойством `fo:language`.

Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности см. раздел 15.4.13.

```
13948 <define name="style-text-properties-attlist" combine="interleave">
13949     <optional>
13950         <attribute name="fo:country">
13951             <ref name="countryCode"/>
13952         </attribute>
13953     </optional>
13954     <optional>
13955         <attribute name="style:country-asian">
13956             <ref name="countryCode"/>
13957         </attribute>
13958     </optional>
13959     <optional>
13960         <attribute name="style:country-complex">
13961             <ref name="countryCode"/>
13962         </attribute>
13963     </optional>
13964 </define>
```

15.4.25 Стиль шрифта

Используйте свойства `fo:font-style`, `style:font-style-asian` и `style:font-style-complex`, чтобы определить, какой шрифт применяется — курсив или обычный (см. § 7.8.7 в [XSL]).

Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности, см. раздел 15.4.13.

```
13965 <define name="style-text-properties-attlist" combine="interleave">
13966     <optional>
13967         <attribute name="fo:font-style">
13968             <ref name="fontStyle"/>
13969         </attribute>
13970     </optional>
13971     <optional>
13972         <attribute name="style:font-style-asian">
13973             <ref name="fontStyle"/>
13974         </attribute>
13975     </optional>
13976     <optional>
13977         <attribute name="style:font-style-complex">
```

```
            <ref name="fontStyle"/>
        </attribute>
    </optional>
</define>

<define name="fontStyle">
    <choice>
        <value>normal</value>
        <value>italic</value>
        <value>oblique</value>
    </choice>
</define>
```

15.4.26 Рельефный шрифт

Используйте свойство `style:font-relief`, чтобы определить, является ли текст выпуклым, высеченным или плоским.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:font-relief">
            <choice>
            <value>none</value>
                <value>embossed</value>
                <value>engraved</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.4.27 Тень шрифта

Используйте свойство `fo:text-shadow`, чтобы определить применение теневого эффекта (см. § 7.16.5 в [XSL]).

Некоторые приложения могут поддерживать ограниченное количество теневых эффектов, например только теневой стиль по умолчанию.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:text-shadow">
            <ref name="shadowType"/>
        </attribute>
    </optional>
</define>

<define name="shadowType">
    <choice>
        <value>none</value>
        <!-- The following string must match an XSL shadow decl -->
        <ref name="string"/>
    </choice>
</define>
```

15.4.28 Тип подчеркивания

Используйте свойство `style:text-underline-type`, чтобы определить подчеркивается ли текст, и если да, то одинарной или двойной чертой.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-underline-type">
            <ref name="lineType"/>
        </attribute>
    </optional>
</define>

<define name="lineType">
    <choice>
        <value>none</value>
        <value>single</value>
        <value>double</value>
    </choice>
</define>
```

15.4.29 Стиль подчеркивания

Используйте свойство `style:text-underline-style`, чтобы определить подчеркивается ли текст, и если да, то как. Значение данного свойства — стиль подчеркивания, например `single`, `dotted`, `dash`. Это свойство похоже на свойство `text-underline-style` (см. [CSS3Text]), за исключением того, что оно имеет дополнительное значение `long-dash` и не имеет значения `double`. Вместо этого имеется атрибут `style:text:underline-type`, который можно использовать для установления двойной линии подчеркивания (см. § 9.2 [CSS3Text]).

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-underline-style">
            <ref name="lineStyle"/>
        </attribute>
    </optional>
</define>

<define name="lineStyle">
    <choice>
        <value>none</value>
        <value>solid</value>
        <value>dotted</value>
        <value>dash</value>
        <value>long-dash</value>
        <value>dot-dash</value>
        <value>dot-dot-dash</value>
        <value>wave</value>
    </choice>
</define>
```

15.4.30 Протяженность подчеркивания

Используйте свойство `style:text-underline-width`, чтобы определить протяженность подчеркивания. Данное свойство очень похоже на свойство [CSS3Text] `text-underline-width` за исключением того, что оно имеет дополнительное значение `bold`. `bold` устанавливает протяженность линии, которая вычисляется на основании размеров шрифта, подобно `auto` протяженности, однако `bold` шире `auto` протяженности.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-underline-width">
            <ref name="lineWidth"/>
        </attribute>
    </optional>
</define>

<define name="lineWidth">
    <choice>
        <value>auto</value>
        <value>normal</value>
        <value>bold</value>
        <value>thin</value>
        <value>dash</value>
        <value>medium</value>
        <value>thick</value>
        <ref name="positiveInteger"/>
        <ref name="percent"/>
        <ref name="positiveLength"/>
    </choice>
</define>
```

15.4.31 Цвет подчеркивания

Используйте свойство `style:text-underline-color`, чтобы определить цвет подчеркивания. Значением данного свойства может быть `font-color` либо конкретное значение цвета. Если значение равно `font-color`, то цветом линий подчеркивания становится текущее значение цвета шрифта.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-underline-color">
            <choice>
                <value>font-color</value>
                <ref name="color"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.4.32 Насыщенность шрифта

Используйте свойства `fo:font-weight`, `style:font-weight-asian` и `style:font-weight-complex`, чтобы определить насыщенность шрифта (см. § 7.8.9 в [XSL]).

Поддерживаются точные числовые значения, относительные значения `lighter` или `bolder` не поддерживаются. Неподдерживаемые числовые значения округляются до ближайшего поддерживаемого значения.

Чтобы получить информацию о том, когда используется значение атрибута для восточно-азиатской системы письменности, а когда значение атрибута для сложных систем письменности, см. раздел 15.4.13.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:font-weight">
            <ref name="fontWeight"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:font-weight-asian">
            <ref name="fontWeight"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:font-weight-complex">
            <ref name="fontWeight"/>
        </attribute>
    </optional>
</define>

<define name="fontWeight">
    <choice>
        <value>normal</value>
        <value>bold</value>
        <value>100</value>
        <value>200</value>
        <value>300</value>
        <value>400</value>
        <value>500</value>
        <value>600</value>
        <value>700</value>
        <value>800</value>
        <value>900</value>
    </choice>
</define>
```

15.4.33 Режим подчеркивания слов

Используйте свойство `style:text-underline-mode`, чтобы определить, применяется ли подчеркивание для отдельных слов или для участков текста. Если применяется подчеркивание для участков текста, подчеркиваются слова и

пробелы между ними. Это свойство подобно свойству `text-underline-mode` из [CSS3Text] (см. § 9.5 в [CSS3Text]).

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-underline-mode">
            <ref name="lineMode"/>
        </attribute>
    </optional>
</define>

<define name="lineMode">
    <choice>
        <value>continuous</value>
        <value>skip-white-space</value>
    </choice>
</define>
```

15.4.34 Режим зачеркивания слов

Используйте свойство `style:text-line-through-mode`, чтобы определить, применяется ли зачеркивание для отдельных слов или для участков текста. Если применяется зачеркивание для участков текста, зачеркиваются слова и пробелы между ними. Это свойство подобно свойству `text-line-through-mode` из [CSS3Text] (см. § 9.5 [CSS3Text]).

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-line-through-mode">
            <ref name="lineMode"/>
        </attribute>
    </optional>
</define>
```

15.4.35 Кернинг букв

Используйте свойство, чтобы активировать или деактивировать кернинг между символами.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:letter-kerning">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.4.36 Мигание текста

Используйте свойство `style:text-blinking`, чтобы определить мигает ли текст.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-blinking">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.4.37 Цвет фона текста

Используйте свойство `fo:background-color`, чтобы определить цвет фона (см. § 7.7.2 [XSL]).

Значением этого свойства может быть `transparent` либо конкретное значение цвета.

```
<define name="style-text-properties-attlist" combine="interleave">
    <ref name="common-background-color-attlist"/>
</define>
```

15.4.38 Объединение текста

Используйте свойство `style:text-combine`, чтобы объединить символы так, чтобы они отображались в две строки.

Значением этого атрибута может быть `none`, `letters` или `lines`.

Если значение равно `lines`, все символы с таким значением атрибута, идущие непосредственно друг за другом отображаются в две строки примерно одинаковой длины. Чтобы этого добиться, между некоторыми двумя символами может иметься разделитель.

В том случае, если значением атрибута является `letters`, до пяти символов объединяется в две строки. Любые дополнительные символы отображаются как обычный текст.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-combine">
            <choice>
                <value>none</value>
                <value>letters</value>
                <value>lines</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.4.39 Начальный и конечный символы объединения текста

Используйте свойства `style:text-combine-start-char` и `style:text-combine-end-char`, чтобы определить начальный и конечный символы, которые отображаются перед и после участка текста, чье свойство `style:text-combine` принимает значение `lines`.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-combine-start-char">
            <ref name="character"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:text-combine-end-char">
            <ref name="character"/>
        </attribute>
    </optional>
</define>
```

15.4.40 Выделение текста

Используйте свойство `style:text-emphasize`, чтобы выделить текст в восточно-азиатских документах.

Значение данного атрибута состоит из двух разделенных пробелом значений.

Первое значение определяет стиль выделения и может принимать значения `none`, `accent`, `dot`, `circle` или `disc`.

Второе значение определяет положение выделения и может принимать значения `above` или `below`. Если первое значение равно `none`, то второе может быть опущено.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-emphasize">
            <choice>
                <value>none</value>
                <list>
                    <choice>
                        <value>none</value>
                        <value>accent</value>
                        <value>dot</value>
                        <value>circle</value>
                        <value>disc</value>
                    </choice>
                    <choice>
                        <value>above</value>
                        <value>below</value>
                    </choice>
                </list>
```

```
            </choice>
        </attribute>
    </optional>
</define>
```

15.4.41 Масштаб текста

Используйте свойство `style:text-scale`, чтобы увеличить или уменьшить масштаб текста, масштабируя шрифт.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-scale">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

15.4.42 Угол вращения текста

Свойство `style:text-rotation-angle` определяет угол, на который текст будет повернут. Значением этого атрибута могут быть `0`, `90`, или `270`. При значении угла, большем 359, используется остаток от деления этого значения на 360. Любое значение остатка, не равное 0, 90 или 270, заменяется ближайшим из них.

Если данный атрибут установлен для более чем одного символа, то будет повернут весь текст, содержащий эти символы.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-rotation-angle">
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

15.4.43 Масштабирование при вращении

Если текст повернут, свойство `style:text-rotation-scale` определяет, должен ли текст быть масштабирован, чтобы уместиться в текущую строку по высоте либо он должен остаться неизменным, но тогда меняется текущая высота строки.

```
<define name="style-text-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-rotation-scale">
            <choice>
                <value>fixed</value>
```

```
14218                <value>line-height</value>
14219            </choice>
14220        </attribute>
14221    </optional>
14222 </define>
```

15.4.44 Расстановка переносов

Используйте свойство `fo:hyphenate`, чтобы включить или выключить автоматическую расстановку переносов (см. § 7.9.4 в [XSL]).

Некоторые приложения могут не поддерживать свойства `fo:hyphenate`, `fo:hyphenation-keep`, `fo:hyphenation-remain-char-count`, `fo:hyphenation-push-char-count` и `fo:hyphenation-ladder-count`, независимые друг от друга в стиле. Целесообразным значением по умолчанию для свойства `fo:hyphenate` в этом случае является `false`.

```
14223 <define name="style-text-properties-attlist" combine="interleave">
14224     <optional>
14225         <attribute name="fo:hyphenate">
14226             <ref name="boolean"/>
14227         </attribute>
14228     </optional>
14229 </define>
```

15.4.45 Количество символов до переноса

Используйте свойство, чтобы установить количество символов, которое должно должно присутствовать до символа переноса (см. § 7.9.7 в [XSL]).

Некоторые приложения могут не поддерживать такие значения свойства, как `fo:hyphenate`, `fo:hyphenation-keep`, `fo:hyphenation-remain-char-count`, `fo:hyphenation-push-char-count` и `fo:hyphenation-ladder-count`, независимо друг от друга в стиле. Целесообразное значение по умолчанию для свойства `fo:hyphenation-remain-char-count` в таком случае будет 0.

```
14230 <define name="style-text-properties-attlist" combine="interleave">
14231     <optional>
14232         <attribute name="fo:hyphenation-remain-char-count">
14233             <ref name="positiveInteger"/>
14234         </attribute>
14235     </optional>
14236 </define>
```

15.4.46 Количество переносимых символов

Используйте свойство `fo:hyphenation-push-char-count`, чтобы указать, какое минимальное количество символов может быть перенесено на следующую строку (см. § 7.9.6 в [XSL]).

Некоторые приложения не поддерживают такие значения данного свойства, как `fo:hyphenate`, `fo:hyphenation-keep`, `fo:hyphenation-remain-char-count`, `fo:hyphenation-push-char-count` и `fo:hyphenation-ladder-count` независимо друг от друга в стиле. Целесообразное значение по умолчанию для параметра `fo:hyphenation-push-char-count` в этом случае будет 0.

```
14237 <define name="style-text-properties-attlist" combine="interleave">
14238     <optional>
14239         <attribute name="fo:hyphenation-push-char-count">
14240             <ref name="positiveInteger"/>
14241         </attribute>
14242     </optional>
14243 </define>
```

15.4.47 Скрытый или условный текст

Свойство `text:display` позволяет скрывать текст. Также это скрытие текста можно сделать зависимым от некоторого условия. Эти атрибуты и их значения такие же, как и для атрибута `text:display` для разделов текста (см. раздел 4.4). Значениями этого атрибута могут быть:

- `true` — текст будет отображен как обычно, это значение используется по умолчанию;
- `none` — текст будет скрытым;
- `condition` — определяет, будет ли текст скрытым или нет, в этом случае должен присутствовать атрибут `text:condition`, определяющий условие.

```
14244 <define name="style-text-properties-attlist" combine="interleave">
14245     <choice>
14246         <attribute name="text:display">
14247             <value>true</value>
14248         </attribute>
14249         <attribute name="text:display">
14250             <value>none</value>
14251         </attribute>
14252         <group>
14253             <attribute name="text:display">
14254                 <value>condition</value>
14255             </attribute>
14256             <attribute name="text:condition">
```

```
                <value>none</value>
            </attribute>
        </group>
        <empty/>
    </choice>
</define>
```

15.5 Свойства форматирования абзаца

Свойства описанные в этом разделе могут быть описаны для стилей абзаца (см. раздел 14.8.2), а также и для других стилей, таких как стили ячеек (см. раздел 14.12.4). Они включатся в элемент `<style:paragraph-properties>`.

```
<define name="style-paragraph-properties">
    <element name="style:paragraph-properties">
        <ref name="style-paragraph-properties-content"/>
    </element>
</define>

<define name="style-paragraph-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-paragraph-properties-content-strict">
    <ref name="style-paragraph-properties-attlist"/>
    <ref name="style-paragraph-properties-elements"/>
</define>
```

15.5.1 Фиксированная высота строки

Используйте свойство `fo:line-height` для определения фиксированной высоты строки, размер или процентное значение которой связано с самым большим символом в строке. Значение `normal` активизирует расчет высоты строки по умолчанию. Это свойство также используется для отключения эффектов свойств `style:line-height-at-least` и `style:line-spacing`. Значением этого свойства может быть размер, процент или значение `normal` (см. § 7.15.4 в [XSL] для подробного ознакомления).

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:line-height">
            <choice>
                <value>normal</value>
                <ref name="nonNegativeLength"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.2 Минимальная высота строки

Используйте свойство `style:line-height-at-least` для определения минимальной высоты строки. Это свойство может содержать значение размера. Значение `normal` использовать нельзя.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:line-height-at-least">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

15.5.3 Межстрочный интервал

Свойство `style:line-spacing` используется для определения фиксированного интервала между двумя строками. Значение `normal` не может быть использовано.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:line-spacing">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.5.4 Межстрочный интервал, независимый от шрифта

Это свойство `style:font-independent-line-spacing` определяет, зависит ли межстрочный интервал от шрифта. Если установлено значение `true`, то высота строки рассчитывается только от высоты шрифта и зависит от атрибутов `fo:font-size`, `style:font-size-asian` и `style:font-size-complex`, определяющих размер шрифта. Если значение равно `false`, то единицы текущего шрифта принимаются во внимание.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:font-independent-line-spacing">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.5.5 Выравнивание текста

Свойство `fo:text-align` используется для определения того, как выравнен текст в абзацах.

Это свойство может принимать значение `start`, `end`, `left`, `right`, `center` или `justify` (см. §7.15.9 [XSL] для детального ознакомления). Значения `inside` и `outside` не поддерживаются.

Если значение для свойств `fo:text-align-last` и `style:justify-single-word` не определено в текущем элементе, то значение этих свойств устанавливается в `start` и `false` соответственно.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <ref name="common-text-align"/>
</define>

<define name="common-text-align">
    <optional>
        <attribute name="fo:text-align">
            <choice>
                <value>start</value>
                <value>end</value>
                <value>left</value>
                <value>right</value>
                <value>center</value>
                <value>justify</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.6 Выравнивание текста в последней строке

Используйте свойство `fo:text-align-last` для определения, как выравнена последняя строка настраиваемого абзаца (см. § 7.15.9 [XSL] для детального ознакомления). Это свойство может принимать только следующие значения `start`, `center` или `justify`.

Это свойство игнорируется, если не сопровождается свойством `fo:text-align`.

Если значения для свойств `fo:text-align` и `style:justify-single-word` не определены, то они принимают значение `start` и `false` соответственно.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:text-align-last">
            <choice>
                <value>start</value>
                <value>center</value>
```

```
                <value>justify</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.7 Выравнивание отдельного слова

Если последняя строка абзаца выровнена по ширине, то свойство `style:justify-single-word` используется для определения того, должно ли единственное слово быть выровнено по ширине.

Если значение для свойств `fo:text-align` и `fo:text-align-last` не определены, то они устанавливаются в значение `start`. Это означает, что определение свойства `style:justify-single-word` без определения свойств `fo:text-align` и `fo:text-align-last` не даст никакого эффекта.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:justify-single-word">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.5.8 Не разрывать

Используйте свойство `fo:keep-together` для контролирования строк абзаца, которые должны быть вместе на странице или в колонке, если установлено значение `alway`), или при разрешенных разрывах в абзаце, если установлено значение `auto` (см. § 7.19.3 [XSL] для детального ознакомления).

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:keep-together">
            <choice>
                <value>auto</value>
                <value>always</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.9 Висячие строки в начале абзаца

Свойство `fo:widows` используется для определения минимального числа строк, допустимых вверху страницы, без отрыва от абзаца (см. § 7.19.7 [XSL] для детального ознакомления).

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:widows">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.5.10 Висячие строки в конце абзаца

Свойство `fo:orphans` используется для определения минимального числа строк, допустимых внизу страницы, без отрыва от абзаца (см. § 7.19.6 [XSL] для детального ознакомления).

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:orphans">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.5.11 Табулостопы

Элемент `<style:tab-stops>` табуляции используется для определения табулостопов.

Каждая позиция табулостопов представлена одним элементом `<style:tab-stop>`, включенным в элемент `<style:tab-stops>`.

```
<define name="style-paragraph-properties-elements" combine="interleave">
    <ref name="style-tab-stops"/>
</define>

<define name="style-tab-stops">
    <optional>
        <element name="style:tab-stops">
            <zeroOrMore>
                <ref name="style-tab-stop"/>
            </zeroOrMore>
        </element>
    </optional>
</define>

<define name="style-tab-stop">
    <element name="style:tab-stop">
        <ref name="style-tab-stop-attlist"/>
        <empty/>
    </element>
</define>
```

С элементами `<style:tab-stop>` могут быть связаны следующие атрибуты:

- позиция табулостопа;

- тип табулостопа;
- символ разделителя;
- тип пунктирной линии;
- стиль пунктирной линии;
- ширина пунктирной линии;
- цвет пунктирной линии;
- текстовая линия;
- стиль текстовой линии.

Позиция табулостопа

Атрибут `style:position` определяет позицию табулостопа.

Этот атрибут связан с элементом `<style:tab-stop>`, в качестве значения устанавливается длина.

```
<define name="style-tab-stop-attlist" combine="interleave">
    <attribute name="style:position">
        <ref name="nonNegativeLength"/>
    </attribute>
</define>
```

Тип табулостопа

Атрибут `style:type` определяет тип табулостопа.

Этот атрибут связан с элементом `<style:tab-stop>` и может принимать значения `left`, `center`, `right` или `char`.

```
<define name="style-tab-stop-attlist" combine="interleave">
    <choice>
        <optional>
            <attribute name="style:type" a:defaultValue="left">
                <choice>
                    <value>left</value>
                    <value>center</value>
                    <value>right</value>
                </choice>
            </attribute>
        </optional>
        <group>
            <attribute name="style:type">
                <value>char</value>
            </attribute>
            <ref name="style-tab-stop-char-attlist"/>
        </group>
    </choice>
</define>
```

Символ разделителя

Атрибут style:char определяет символ разделителя типа char для табулостопа.

Этот атрибут связан с элементом <style:tab-stop> и должен присутствовать, если установлено значение char атрибута style:type. Если значение атрибута style:type установлено не char, то оно игнорируется.

Значение этого атрибута должно быть символом [UNICODE].

```
<define name="style-tab-stop-char-attlist" combine="interleave">
    <attribute name="style:char">
        <ref name="character"/>
    </attribute>
</define>
```

Тип пунктирной линии

Атрибут style:leader-type определяет, отображается ли пунктирная линия, и если да, то используется ли одиночная или двойная линия (см. также раздел 15.4.28).

```
<define name="style-tab-stop-attlist" combine="interleave">
    <optional>
        <attribute name="style:leader-type">
            <ref name="lineType"/>
        </attribute>
    </optional>
</define>
```

Стиль пунктирной линии

Свойство style:leader-style используется для определения когда и каким образом изображается пунктирная линия. Стили линий, которые можно использовать рассмотрены в разделе 15.4.29.

```
<define name="style-tab-stop-attlist" combine="interleave">
    <optional>
        <attribute name="style:leader-style">
            <ref name="lineStyle"/>
        </attribute>
    </optional>
</define>
```

Ширина пунктирной линии

Свойство style:leader-width используется для определения ширины пунктирной линии. Для определения значений этого атрибута см. раздел 15.4.30.

```
<define name="style-tab-stop-attlist" combine="interleave">
    <optional>
        <attribute name="style:leader-width">
            <ref name="lineWidth"/>
        </attribute>
    </optional>
</define>
```

Цвет пунктирной линии

Свойство `style:leader-color` используется для определения цвета, установленного для пунктирной линии. Это свойство может принимать значение `font-color` или название цвета. Если значение установлено в `font-color`, то цвет текущего текста используется и для пунктирной линии.

```
<define name="style-tab-stop-attlist" combine="interleave">
    <optional>
        <attribute name="style:leader-color">
            <choice>
                <value>font-color</value>
                <ref name="color"/>
            </choice>
        </attribute>
    </optional>
</define>
```

Текстовая линия

Атрибут `style:leader-text` определяет текстовую линию, используемую в табуляции. Если значение атрибута не пустое, то значение строки атрибута используется как текстовая линия вместо линии, которая была определена, при условии, что приложение поддерживает текстовую линию. Если приложение не поддерживает текст, то этот атрибут игнорируется, это означает, что будет использован только `style:leader-style`. Если приложение поддерживает текст, состоящий только из отдельных символов, и текстовая линия состоит больше чем из одного символа, используется только первый символ. Если приложение поддерживает текстовую линию только с определенными символами (как «.» или «_»), то приложение должно использовать только один из этих символов, если текстовая линия определена неподдерживаемыми символами. Другими словами: текстовая линия имеет более высокий приоритет, чем пунктирная линия, даже если текстовая линия должна быть адаптирована для использования ее приложением.

Этот атрибут связан с элементом `<style:tab-stop>` и в качестве значения должен быть установлен один символ [UNICODE].

```
<define name="style-tab-stop-attlist" combine="interleave">
    <optional>
        <attribute name="style:leader-text" a:defaultValue=" ">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Стиль текстовой линии

Атрибут `style:leader-text-style` определяет стиль текста, применяемого для текстовой линии. Он не применяется для основного текста строк. Если атрибут использует автоматический стиль, то он может определить автоматически стиль для каждого текста или только общий стиль. Если атрибут используется в общем стиле, то он может быть связан только с общим стилем.

```
<define name="style-tab-stop-attlist" combine="interleave">
    <optional>
        <attribute name="style:leader-text-style">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

15.5.12 Дистанция табулостопов

Атрибут `style:tab-stop-distance` определяет расстояние между началом и концом табулостопа. По умолчанию табулостоп автоматически повторяется после указанного расстояния. Табулостопы по умолчанию вычисляются, только если они определены стилем по умолчанию (см. раздел 14.2).

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:tab-stop-distance">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

15.5.13 Удержание переноса

Свойство `fo:hyphenation-keep` используется для включения или отключения расстановки переносов в последнем слове страницы (см. § 7.15.1 в [XSL] для детального ознакомления).

Некоторые приложения могут не поддерживать установку свойств fo:hyphenate, fo:hyphenation-keep, fo:hyphenation-remain-char-count, fo:hyphenation-push-char-count и fo:hyphenation-ladder-count в стиле независимо друг от друга. В этом случае для fo:hyphenation-keep подходящим значением является auto.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:hyphenation-keep">
            <choice>
                <value>auto</value>
                <value>page</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.14 Максимальное число последовательных переносов

Свойство fo:hyphenation-ladder-count используется для определения максимального числа следующих друг за другом строк, которые могут включать слова с переносом (см. § 7.15.2 [XSL] для детального ознакомления).

Некоторые приложения могут не поддерживать установку свойств fo:hyphenate, fo:hyphenation-keep, fo:hyphenation-remain-char-count, fo:hyphenation-push-char-count и fo:hyphenation-ladder-count в стиле независимо друг от друга. В этом случае для fo:hyphenation-push-char-count целесообразным значением является no-limit.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:hyphenation-ladder-count">
            <choice>
                <value>no-limit</value>
                <ref name="positiveInteger"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.15 Буквица

Элемент <style:drop-cap> используется для определения, отображается ли первый символ или больше символов в абзаце в прописном шрифте. Этот элемент может включать элемент <style:paragraph-properties>.

```
<define name="style-paragraph-properties-elements" combine="interleave">
    <ref name="style-drop-cap"/>
```

```
14492 </define>
14493
14494 <define name="style-drop-cap">
14495     <optional>
14496         <element name="style:drop-cap">
14497             <ref name="style-drop-cap-attlist"/>
14498             <empty/>
14499         </element>
14500     </optional>
14501 </define>
```

Элемент `<style:drop-cap>` может быть связан со следующими атрибутами:

- длина;
- строки;
- интервал;
- стиль текста.

Длина

Атрибут `style:length` определяет число символов, которые нужно сделать прописными.

Значение этого атрибута может быть числом или `word`, которое указывает, что первое слово должно быть сделано прописным.

```
14502 <define name="style-drop-cap-attlist" combine="interleave">
14503     <optional>
14504         <attribute name="style:length" a:defaultValue="1">
14505             <choice>
14506                 <value>word</value>
14507                 <ref name="positiveInteger"/>
14508             </choice>
14509         </attribute>
14510     </optional>
14511 </define>
```

Строки

Атрибут `style:lines` определяет число строк, окружающих буквицу.

```
14512 <define name="style-drop-cap-attlist" combine="interleave">
14513     <optional>
14514         <attribute name="style:lines" a:defaultValue="1">
14515             <ref name="positiveInteger"/>
14516         </attribute>
14517     </optional>
14518 </define>
```

Интервал

Атрибут `style:distance` определяет размер между последним символом буквицы и первым из оставшихся символов в каждой строке. Значением этого атрибута является длина.

```
14519 <define name="style-drop-cap-attlist" combine="interleave">
14520     <optional>
14521         <attribute name="style:distance" a:defaultValue="0cm">
14522             <ref name="length"/>
14523         </attribute>
14524     </optional>
14525 </define>
```

Стиль текста

Атрибут `style:style-name` определяет стиль текста буквицы.

```
14526 <define name="style-drop-cap-attlist" combine="interleave">
14527     <optional>
14528         <attribute name="style:style-name">
14529             <ref name="styleNameRef"/>
14530         </attribute>
14531     </optional>
14532 </define>
```

15.5.16 Приводка

Свойство `style:register-true` определяет, совпадают ли линии обоих сторон на распечатанной странице, когда печать осуществлялась в режиме двусторонней печати. Это также гарантирует, что текст в колонках страницы или колонки текстового блока расположены так, что базовые линии текста начала одной колонки кажутся окончанием другой (см. также раздел 15.2.12).

```
14533 <define name="style-paragraph-properties-attlist" combine="interleave">
14534         <optional>
14535             <attribute name="style:register-true">
14536                 <ref name="boolean"/>
14537             </attribute>
14538         </optional>
14539     </define>
```

15.5.17 Левое и правое поля

Свойства `fo:margin-left` и `fo:margin-right` используются для определения левого и правого поля для абзаца (см. § 7.10.3 и § 7.10.4 [XSL] для детального ознакомления). Значение `auto` не поддерживается. Процентное

значение поддерживается, только для общих стилей. В этом случае они определяют края родительского стиля.

Для некоторых приложений эти два свойства должны быть использованы совместно и также вместе со свойством `fo:text-indent`. Если хотя бы одно из этих значений отсутствует, его значение устанавливается в 0 см.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <ref name="common-horizontal-margin-attlist"/>
</define>

<define name="common-horizontal-margin-attlist">
    <optional>
        <attribute name="fo:margin-left">
            <choice>
                <ref name="length"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:margin-right">
            <choice>
                <ref name="length"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.18 Отступ текста

Свойство `fo:text-indent` используется для определения положительного или отрицательного отступа первой строки абзаца (см. § 7.15.11 [XSL] для детального ознакомления). Процентное значение поддерживается только в общих стилях. Здесь они соответствуют относительным границам родительского стиля.

Для некоторых приложений свойство `fo:text-indent` должно использоваться вместе со свойствами `fo:margin-left` и `fo:margin-right`. Если любое из этих свойств пропущено, то его значение устанавливается в 0 см.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:text-indent">
            <choice>
                <ref name="length"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.19 Автоматический отступ текста

Свойство `style:auto-text-indent` используется для определения того, что значение отступа первой строки абзаца базируется на размере текущего шрифта.

Для некоторых приложений свойство `style:auto-text-indent` должно использоваться вместе со свойствами `fo:margin-left` и `fo:margin-right`. Если любое из этих свойств пропущено, то его значение устанавливается в 0 см.

Если значение этого свойства `true` и используется вместе с свойством `fo:text-indent`, тогда свойство `fo:text-indent` игнорируется.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:auto-text-indent">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.5.20 Верхнее и нижнее поля

Свойства `fo:margin-top` и `fo:margin-bottom` используются для определения верхнего и нижнего поля для абзаца (см. § 7.10.1 и § 7.10.2 [XSL] для детального ознакомления). Значение `auto` не поддерживается. Процентные значения поддерживаются только для общих стилей. Они ссылаются на соответствующие поля родительских стилей.

Для некоторых приложений эти два свойства должны использоваться вместе. Если любое из этих свойств пропущено, то его значение устанавливается в 0 см.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <ref name="common-vertical-margin-attlist"/>
</define>

<define name="common-vertical-margin-attlist">
    <optional>
        <attribute name="fo:margin-top">
            <choice>
                <ref name="nonNegativeLength"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:margin-bottom">
            <choice>
                <ref name="nonNegativeLength"/>
```

```
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.21 Поля

Свойство `fo:margin` используется для одновременного определения верхнего, нижнего, левого и правого поля для абзацев (см. § 7.29.4 [XSL] и разделы 15.5.17 и 15.5.20 для детального ознакомления).

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <ref name="common-margin-attlist"/>
</define>

<define name="common-margin-attlist">
    <optional>
        <attribute name="fo:margin">
            <choice>
                <ref name="nonNegativeLength"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.22 Разрыв до и разрыв после

Свойства `fo:break-before` и `fo:break-after` используются для размещения на странице или в колонке: разрыв до или разрыв после абзаца (см. § 7.19.1 и § 7.19.2 [XSL] детального ознакомления). Значения `odd-page` и `even-page` не поддерживаются.

Эти два свойства взаимоисключающие. Если они используются вместе, то результат не определен.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <ref name="common-break-attlist"/>
</define>

<define name="common-break-attlist">
    <optional>
        <attribute name="fo:break-before">
            <choice>
                <value>auto</value>
                <value>column</value>
                <value>page</value>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:break-after">
            <choice>
```

```
                <value>auto</value>
                <value>column</value>
                <value>page</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.23 Фоновый цвет абзаца

Свойство `fo:background-color` используется для определения фонового цвета абзаца (см. § 7.7.2 [XSL] для детального ознакомления).

Значение этого атрибута может быть `transparent` или цветом. Если значение `transparent`, то включается любое изображение, определенное в элементе `<style:background-image>`.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <ref name="common-background-color-attlist"/>
</define>

<define name="common-background-color-attlist">
    <optional>
        <attribute name="fo:background-color">
            <choice>
                <value>transparent</value>
                <ref name="color"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.24 Фоновое изображение абзаца

Элемент `<style:background-image>` используется для определения фонового изображения абзаца.

Фоновое изображение может храниться одним из следующих способов (см. также раздел 9.3.2):

- данные изображения хранятся во внешнем файле, используются атрибуты [XLink] для определения месторасположения рисунка;
- данные изображения включены в подэлемент `<office:binary-data>` в кодировке BASE64.

Если элемент `<style:background-image>` отсутствует и если цвет не определен элементом `fo:background-color` в свойствах тождественных элементов, то фоновый цвет устанавливается в значение `transparent`.

```
<define name="style-paragraph-properties-elements" combine="interleave">
```

```
14654     <ref name="style-background-image"/>
14655 </define>
14656
14657 <define name="style-background-image">
14658     <optional>
14659         <element name="style:background-image">
14660             <ref name="style-background-image-attlist"/>
14661             <choice>
14662                 <ref name="common-draw-data-attlist"/>
14663                 <ref name="office-binary-data"/>
14664                 <empty/>
14665             </choice>
14666         </element>
14667     </optional>
14668 </define>
```

Элемент `<style:background-image>` может быть связан со следующими атрибутами:

- повторение;
- позиция;
- фильтр;
- непрозрачность.

Повторение

Атрибут `style:repeat` определяет, повторяется ли фоновое изображение в абзаце.

Этот атрибут присоединен к элементу `<style:background-image>` и может иметь значения `no-repeat`, `repeat` или `stretch`.

```
14669 <define name="style-background-image-attlist" combine="interleave">
14670     <optional>
14671         <attribute name="style:repeat" a:defaultValue="repeat">
14672             <choice>
14673                 <value>no-repeat</value>
14674                 <value>repeat</value>
14675                 <value>stretch</value>
14676             </choice>
14677         </attribute>
14678     </optional>
14679 </define>
```

Позиция

Атрибут `style:position` определяет позицию фонового изображения в абзаце.

Этот атрибут присоединен к элементу `<style:background-image>` и может иметь разделенную пробелами комбинацию значений `top`, `center` или `bottom` по

вертикальной позиции и `left`, `center` или `right` для горизонтальной позиции. Вертикальные или горизонтальные позиции могут определяться в любом сочетании. Если определена одна позиция, то вторая принимает значение по умолчанию `center`.

```
14680 <define name="style-background-image-attlist" combine="interleave">
14681     <optional>
14682         <attribute name="style:position" a:defaultValue="center">
14683             <choice>
14684                 <value>left</value>
14685                 <value>center</value>
14686                 <value>right</value>
14687                 <value>top</value>
14688                 <value>bottom</value>
14689                 <list>
14690                     <ref name="horiBackPos"/>
14691                     <ref name="vertBackPos"/>
14692                 </list>
14693                 <list>
14694                     <ref name="vertBackPos"/>
14695                     <ref name="horiBackPos"/>
14696                 </list>
14697             </choice>
14698         </attribute>
14699     </optional>
14700 </define>
14701
14702 <define name="horiBackPos">
14703     <choice>
14704         <value>left</value>
14705         <value>center</value>
14706         <value>right</value>
14707     </choice>
14708 </define>
14709 <define name="vertBackPos">
14710     <choice>
14711         <value>top</value>
14712         <value>center</value>
14713         <value>bottom</value>
14714     </choice>
14715 </define>
```

Фильтр

Атрибут `style:filter-name` определяет специфические имена фильтров приложения, которые могут быть загружены для изображения в документе.

Этот атрибут присоединен к элементу `<style:background-image>`.

```
14716 <define name="style-background-image-attlist" combine="interleave">
14717     <optional>
14718         <attribute name="style:filter-name">
14719             <ref name="string"/>
14720         </attribute>
14721     </optional>
14722 </define>
```

Непрозрачность

Атрибут draw:opacity определяет непрозрачность фонового изображения. Значение устанавливается в процентах, где 0 % — прозрачное, 100 % — полностью непрозрачное.

```
<define name="style-background-image-attlist" combine="interleave">
    <optional>
        <attribute name="draw:opacity">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

15.5.25 Обрамление

Свойства обрамления fo:border, fo:border-top, fo:border-bottom, fo:border-left и fo:border-right используются для определения свойств обрамления абзаца (см. § 7.29.3 - § 7.29.7 [XSL] для детального ознакомления).

Свойство fo:border применяется ко всем четырем сторонам абзаца, в то время как другие свойства применимы только к определенным сторонам.

Для некоторых приложений все четыре обрамления должны быть установлены либо использованием свойства fo:border, либо применением всех остальных четырех свойств обрамления к элементу свойств. В последнем случае, если одно или больше свойств пропущены, то их значение устанавливается в none.

Также может быть наложено ограничение относительно поддерживаемых стилей обрамления. В дополнение к этому, некоторые приложения могут добавлять заполнение по умолчанию для сторон с обрамлением.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <ref name="common-border-attlist"/>
</define>

<define name="common-border-attlist">
    <optional>
        <attribute name="fo:border">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:border-top">
            <ref name="string"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:border-bottom">
            <ref name="string"/>
```

```
14748         </attribute>
14749     </optional>
14750     <optional>
14751         <attribute name="fo:border-left">
14752             <ref name="string"/>
14753         </attribute>
14754     </optional>
14755     <optional>
14756         <attribute name="fo:border-right">
14757             <ref name="string"/>
14758         </attribute>
14759     </optional>
14760 </define>
```

15.5.26 Толщина линии обрамления

Если стиль линии для обрамления `double`, то используются свойства обрамления `style:border-line-width`, `style:border-line-width-top`, `style:border-line-width-bottom`, `style:border-line-width-left` и `style:border-line-width-right` для индивидуального определения толщины внутренних и внешних линий и расстояния между ними.

`style:border-line-width` определяет толщину линии для всех четырех сторон, в то время как другие атрибуты определят толщину линии только для одной из сторон.

Значение атрибутов может быть списком трех разделенных пробелом значений:

- первое значение определяет толщину внутренней линии;
- второе значение определяет расстояние между двумя линиями;
- третье значение определяет толщину внешней линии.

Результат объединения обрамления без определения рамки со стилем `double` для некоторых рамок не определен.

```
14761 <define name="style-paragraph-properties-attlist" combine="interleave">
14762     <ref name="common-border-line-width-attlist"/>
14763 </define>
14764
14765 <define name="common-border-line-width-attlist">
14766     <optional>
14767         <attribute name="style:border-line-width">
14768             <ref name="borderWidths"/>
14769         </attribute>
14770     </optional>
14771     <optional>
14772         <attribute name="style:border-line-width-top">
14773             <ref name="borderWidths"/>
14774         </attribute>
14775     </optional>
14776     <optional>
```

```
        <attribute name="style:border-line-width-bottom">
            <ref name="borderWidths"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:border-line-width-left">
            <ref name="borderWidths"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="style:border-line-width-right">
            <ref name="borderWidths"/>
        </attribute>
    </optional>
</define>

<define name="borderWidths">
    <list>
        <ref name="positiveLength"/>
        <ref name="positiveLength"/>
        <ref name="positiveLength"/>
    </list>
</define>
```

15.5.27 Заполнение

Свойства заполнения `fo:padding`, `fo:padding-top`, `fo:padding-bottom`, `fo:padding-left` и `fo:padding-right` используются для определения промежутка вокруг абзаца (см. § 7.29.15 и §7.7.35- § 7.7.38 [XSL] для детального ознакомления).

Для некоторых приложений значение этих параметров может быть не нулевым, только если для этих сторон есть обрамление и оно определено некоторыми свойствами элемента. Если свойства элемента включают специальное заполнение для одной, но не для всех сторон, некоторые приложения могут также связывать нулевое или установленное по умолчанию значение заполнения для этих сторон, зависящее от того, есть ли обрамление на той стороне или нет. Также может быть и другое ограничение относительно обрамления и заполнения.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <ref name="common-padding-attlist"/>
</define>

<define name="common-padding-attlist">
    <optional>
        <attribute name="fo:padding">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:padding-top">
```

```
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:padding-bottom">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:padding-left">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:padding-right">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

15.5.28 Тень

Свойство `style:shadow` используется для определения эффекта тени для абзаца.

Допустимыми значениями для этого атрибута являются значения для свойства `fo:text-shadow` (см. раздел 15.4.27 для получения дополнительной информации).

Некоторые приложения могут поддерживать ограниченное число эффектов тени, например, когда позиция смещения тени одинакова по горизонтали и вертикали.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <ref name="common-shadow-attlist"/>
</define>

<define name="common-shadow-attlist">
    <optional>
        <attribute name="style:shadow">
            <ref name="shadowType"/>
        </attribute>
    </optional>
</define>
```

15.5.29 Вместе со следующим

Свойство `fo:keep-with-next` используется для определения возможности связать текущий абзац со следующим на странице или в колонке после добавления разрыва (см. § 7.9.14 [XSL] для детального ознакомления). Поддерживаются только значения `auto` и `always`.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <ref name="common-keep-with-next-attlist"/>
</define>

<define name="common-keep-with-next-attlist">
    <optional>
        <attribute name="fo:keep-with-next">
            <choice>
                <value>auto</value>
                <value>always</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.30 Нумерация строк

Атрибут `text:number-lines` определяет, контролируется ли подсчет строк.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:number-lines" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.5.31 Начальное значение нумерации строк

Свойство `text:line-number` определяет начальное значение для нумерации строк. Атрибут обрабатывается только в том случае, если в том же самом элементе задан атрибут `text:number-lines` со значением `true`.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:line-number">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.5.32 Автоматическое разделение текста

Свойство `style:text-autospace` используется для определения, добавлять ли промежуток между текстами восточно-азиатским и западным, а также текстом со сложной системой письменности.

Возможные значения `none` и `ideograph-alpha`.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-autospace">
            <choice>
                <value>none</value>
```

```
                <value>ideograph-alpha</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.33 Перенос знаков препинания на новую строку

Свойство `style:punctuation-wrap` используется для определения, действительно ли знак препинания, если он присутствует, может быть подвешен, то есть может ли разместиться на поле, если строка полностью заполнена. Это обычная настройка в восточно-азиатской типографии.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:punctuation-wrap">
            <choice>
                <value>simple</value>
                <value>hanging</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.34 Разрыв строки

Свойство `style:line-break` используется для установки правил разрыва строки для текста. Если установлено значение `strict`, то разрыв строк между установленными пользователем и настраиваемыми приложением символами запрещается. Если установлено значение `normal`, то разрывы строки могут встречаться между любыми символами.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:line-break">
            <choice>
                <value>normal</value>
                <value>strict</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.35 Вертикальное выравнивание

Свойство `style:vertical-align` определяет вертикальную позицию символа. По умолчанию символы выравнены по их базовой строке, которая используется по умолчанию для большинства европейских языков. Это также

выравнивание, которое используется в этой спецификации. Иначе вертикальное выравнивание символов может быть следующим:

- `bottom` — по нижнему краю строки;
- `top` — по верхнему краю строки;
- `middle` — по центру строки;
- `auto` — автоматически, которое устанавливает вертикальное выравнивание в зависимости от ориентации строки. Текст, повернутый на 0 или 90 градусов, выравнивается от базовой линии, в то время как текст, повернутый на 270 градусов, выравнивается от центра строки.

Следующая иллюстрация демонстрирует эффект вертикального выравнивания, когда свойство принимает значение `baseline`, `top`, `bottom`, и `center` соответственно.

```
14900 <define name="style-paragraph-properties-attlist" combine="interleave">
14901     <optional>
14902         <attribute name="style:vertical-align" a:defaultValue="auto">
14903             <choice>
14904                 <value>top</value>
14905                 <value>middle</value>
14906                 <value>bottom</value>
14907                 <value>auto</value>
14908             </choice>
14909         </attribute>
14910     </optional>
14911 </define>
```

15.5.36 Режим записи

Атрибут `style:writing-mode` определяет режим записи в абзаце. Этот атрибут подобен атрибуту `writing-mode`, определенному в § 7.27.7 [XSL], за исключением добавленного значения `page`. Это значение определяет, что режим записи перенят от страницы, которая включает абзац.

```
14912 <define name="style-paragraph-properties-attlist" combine="interleave">
14913     <ref name="common-writing-mode-attlist"/>
14914 </define>
14915
14916 <define name="common-writing-mode-attlist">
14917     <optional>
14918         <attribute name="style:writing-mode">
14919             <choice>
14920                 <value>lr-tb</value>
14921                 <value>rl-tb</value>
14922                 <value>tb-rl</value>
14923                 <value>tb-lr</value>
14924                 <value>lr</value>
```

```
                <value>rl</value>
                <value>tb</value>
                <value>page</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.5.37 Режим автоматической записи

Если атрибут `style:writing-mode-automatic` определен для абзаца и если его значение `true`, тогда приложению разрешено повторно пересчитывать значение режима записи абзаца всякий раз, когда изменяется его содержимое. Фактически значение режима записи должно быть включено в атрибут `style:writing-mode` так, чтобы приложения, которые не поддерживают автоматический пересчет значения режима записи или используют различные алгоритмы, всегда имели актуальные значения.

Определяя дополнительно атрибут `fo:text-align='start'`, выравнивание текста одновременно может быть адаптировано к режиму записи.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:writing-mode-automatic">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.5.38 Привязка к разметке

Атрибут `style:snap-to-layout-grid` определяет, должны ли в абзаце учитываться настройки сетки разметки страницы (см. раздел 15.2.21).

```
<define name="style-paragraph-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:snap-to-layout-grid">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.5.39 Номер страницы

Если стиль абзаца определяет стиль страницы, который должен быть применен, начиная с начала абзаца, атрибут `style:page-number` определяет номер страницы, который должен использоваться для новой страницы.

```
<define name="style-paragraph-properties-attlist" combine="interleave">
```

```
14947     <ref name="common-page-number-attlist"/>
14948 </define>
14949
14950 <define name="common-page-number-attlist">
14951     <optional>
14952         <attribute name="style:page-number">
14953             <ref name="positiveInteger"/>
14954         </attribute>
14955     </optional>
14956 </define>
```

15.5.40 Прозрачность фона

```
14957 <define name="style-paragraph-properties-attlist" combine="interleave">
14958     <optional>
14959         <attribute name="style:background-transparency">
14960             <ref name="percent"/>
14961         </attribute>
14962     </optional>
14963 </define>
```

15.6 Свойства форматирования транскрипции

Свойства, описанные в этом разделе, могут быть использованы в стилях транскрипций (см. раздел 14.8.4 для детального ознакомления). Они включаются в элемент `<style:ruby-properties>`.

```
14964 <define name="style-ruby-properties">
14965     <element name="style:ruby-properties">
14966         <ref name="style-ruby-properties-content"/>
14967     </element>
14968 </define>
14969
14970 <define name="style-ruby-properties-content">
14971     <ref name="style-properties-content"/>
14972 </define>
14973
14974 <define name="style-ruby-properties-content-strict">
14975     <ref name="style-ruby-properties-attlist"/>
14976     <ref name="style-ruby-properties-elements"/>
14977 </define>
14978
14979 <define name="style-ruby-properties-elements">
14980     <empty/>
14981 </define>
```

15.6.1 Позиция транскрипции

Это свойство определяет позицию текста транскрипции относительно базового текста.

```
14982 <define name="style-ruby-properties-attlist" combine="interleave">
14983     <optional>
14984         <attribute name="style:ruby-position">
14985             <choice>
14986                 <value>above</value>
```

```
                <value>below</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.6.2 Выравнивание транскрипции

Это свойство определяет выравнивание текста транскрипции относительно базового текста.

```
<define name="style-ruby-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:ruby-align">
            <choice>
                <value>left</value>
                <value>center</value>
                <value>right</value>
                <value>distribute-letter</value>
                <value>distribute-space</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.7 Свойства форматирования разделов

Свойства, описанные в этом разделе, могут использоваться со стилем раздела (см. раздел 14.8.3 для детального ознакомления). Они включаются в элемент `<style:section-properties>`.

```
<define name="style-section-properties">
    <element name="style:section-properties">
        <ref name="style-section-properties-content"/>
    </element>
</define>

<define name="style-section-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-section-properties-content-strict">
    <ref name="style-section-properties-attlist"/>
    <ref name="style-section-properties-elements"/>
</define>
```

15.7.1 Фон раздела

Атрибут фона `fo:background-color` и элемент фона `<style:background-image>` определяют свойства фона в разделе. Для получения более детальной информации об этом атрибуте и элементе см. разделы 15.5.23 и 15.5.24.

```
15019 <define name="style-section-properties-attlist" combine="interleave">
15020     <ref name="common-background-color-attlist"/>
15021 </define>
15022 <define name="style-section-properties-elements" combine="interleave">
15023     <ref name="style-background-image"/>
15024 </define>
```

15.7.2 Поля

Атрибуты полей `fo:margin-left` и `fo:margin-right` определяют размер полей раздела. Для получения более детальной информации об этих атрибутах см. разделы 15.5.17 . Процентное значение не поддерживается.

```
15025 <define name="style-section-properties-attlist" combine="interleave">
15026     <ref name="common-horizontal-margin-attlist"/>
15027 </define>
```

15.7.3 Колонки

Элемент `<style:columns>` включает элементы `<style:column>`, которые определяют каждую колонку индивидуально (см. раздел 15.7.4). Если эти элементы отсутствуют, то для всех колонок устанавливается одна и та же ширина.

Элемент `<style:columns>` может включать элемент `<style:column-sep>`, который описывает линию разделителя колонок. Для получения подробной информации об этом элементе см. раздел 15.7.5.

```
15028 <define name="style-section-properties-elements" combine="interleave">
15029     <ref name="style-columns"/>
15030 </define>
15031
15032 <define name="style-columns">
15033     <optional>
15034         <element name="style:columns">
15035             <ref name="style-columns-attlist"/>
15036             <optional>
15037                 <ref name="style-column-sep"/>
15038             </optional>
15039             <zeroOrMore>
15040                 <ref name="style-column"/>
15041             </zeroOrMore>
15042         </element>
15043     </optional>
15044 </define>
```

С элементом `<style:columns>` могут быть связаны следующие атрибуты:

- число колонок;
- интервал между колонками.

Число колонок

Атрибут `fo:columns-count` определяет число колонок в разделе.

```
15045 <define name="style-columns-attlist" combine="interleave">
15046     <attribute name="fo:column-count">
15047         <ref name="positiveInteger"/>
15048     </attribute>
15049 </define>
```

Примечание — Этот атрибут называется так же, как свойств0 [XSL], но присоединен к другому элементу.

Интервал между колонками

Если элемент `<style:columns>` не содержит индивидуальных элементов `<style:column>`, то интервал между колонками может быть определен атрибутом `fo:column-gap`. Если индивидуальные элементы колонок определены, то этот атрибут игнорируется.

```
15050 <define name="style-columns-attlist" combine="interleave">
15051     <optional>
15052         <attribute name="fo:column-gap">
15053             <ref name="length"/>
15054         </attribute>
15055     </optional>
15056 </define>
```

Примечание — Этот атрибут называется так же, как свойство [XSL], но присоединен к другому элементу.

15.7.4 Спецификация колонок

Элемент `<style:column>` может быть включен в элемент `<style:columns>`, определяя детализацию индивидуальной колонки. Этот элемент включен в элемент `<styles:columns>`. Здесь может быть либо элемент, не являющийся колонкой, либо такое же количество элементов колонок, как определено в атрибуте `fo:column-count`.

```
15057 <define name="style-column">
15058     <element name="style:column">
15059         <ref name="style-column-attlist"/>
15060     </element>
15061 </define>
```

Примечание — В [XSL] невозможно определить колонки индивидуально.

С элементом `<style:column>` могут быть связаны следующие атрибуты:

- ширина колонки;

- левый, правый, верхний, и нижний интервалы колонки.

Ширина колонки

Атрибут `style:rel-width` используется для определения ширины колонки. Ширина колонки определяется числовым значением вместо длины. Для определения абсолютной ширины колонки доступное для области колонки место распределяется пропорционально среди всех этих колонок.

Ширина колонки определяется не в процентах от длины, но предпочтительнее в терминах относительных весов, где число следует за символом « * ». Общее пространство, доступное для вставки таблицы, распределяется среди включенных колонок соответственно относительным весам. Например, если три колонки связаны с относительными величинами 1, 2 и 3, тогда первая колонка занимает 1/6 выделенной ширины, вторая 1/3 и последняя 1/2 выделенного пространства. Для достижения такой картины все заданные относительные величины должны быть просуммированы (шесть в примере), затем каждая колонка получает место пропорционально ее весу, отнесенному к сумме относительных весов (3/6 = 1/2 для последней колонки в примере).

```
15062 <define name="style-column-attlist" combine="interleave">
15063     <attribute name="style:rel-width">
15064         <ref name="relativeLength"/>
15065     </attribute>
15066 </define>
```

Левый, правый, верхний и нижний интервал колонки

Для каждой колонки может быть определен левый, правый, верхний и нижний интервал. Правое пространство колонки вместе с левым пространством следующей колонки соответствует интервалу между колонками. Если пространство колонки включает разделительную линию между колонками, пространство, занятое линией, включается в левое и правое пространство и поэтому не добавляется к ним.

```
15067 <define name="style-column-attlist" combine="interleave">
15068     <optional>
15069         <attribute name="fo:start-indent" a:defaultValue="0cm">
15070             <ref name="length"/>
15071         </attribute>
15072     </optional>
15073 </define>
15074 <define name="style-column-attlist" combine="interleave">
15075     <optional>
```

```
15076        <attribute name="fo:end-indent" a:defaultValue="0cm">
15077            <ref name="length"/>
15078        </attribute>
15079    </optional>
15080 </define>
15081 <define name="style-column-attlist" combine="interleave">
15082    <optional>
15083        <attribute name="fo:space-before" a:defaultValue="0cm">
15084            <ref name="length"/>
15085        </attribute>
15086    </optional>
15087 </define>
15088 <define name="style-column-attlist" combine="interleave">
15089    <optional>
15090        <attribute name="fo:space-after" a:defaultValue="0cm">
15091            <ref name="length"/>
15092        </attribute>
15093    </optional>
15094 </define>
```

15.7.5 Разделитель колонок

Элемент `<style:column-sep>` определяет разделительную линию, используемую между колонками. Этот элемент может быть включен в элемент `<style:columns>`, определяющий тип разделительной линии, используемой между колонками.

```
15095 <define name="style-column-sep">
15096    <element name="style:column-sep">
15097        <ref name="style-column-sep-attlist"/>
15098    </element>
15099 </define>
```

Примечание — [XSL] не поддерживает разделители колонок.

С элементом `<style:column-sep>` могут быть связаны следующие атрибуты:

- стиль линии;
- ширина линии;
- высота линии;
- вертикальное выравнивание линии;
- цвет линии.

Стиль линии

Атрибут `style:style` используется для определения стиля разделительной линии колонки.

```
15100 <define name="style-column-sep-attlist" combine="interleave">
15101    <optional>
```

```
15102         <attribute name="style:style" a:defaultValue="solid">
15103             <choice>
15104                 <value>none</value>
15105                 <value>solid</value>
15106                 <value>dotted</value>
15107                 <value>dashed</value>
15108                 <value>dot-dashed</value>
15109             </choice>
15110         </attribute>
15111     </optional>
15112 </define>
```

Ширина линии

Атрибут `style:width` используется для определения ширины разделительной линии колонки.

```
15113 <define name="style-column-sep-attlist" combine="interleave">
15114     <attribute name="style:width">
15115         <ref name="length"/>
15116     </attribute>
15117 </define>
```

Высота линии

Атрибут `style:height` используется для определения высоты разделительной линии колонки. Значение этого атрибута измеряется в процентах относительно высоты области колонки.

```
15118 <define name="style-column-sep-attlist" combine="interleave">
15119     <optional>
15120         <attribute name="style:height" a:defaultValue="100%">
15121             <ref name="percent"/>
15122         </attribute>
15123     </optional>
15124 </define>
```

Вертикальное выравнивание линии

Атрибут `style:vertical-align` используется для определения вертикального выравнивания линии, которое определяется от 100 % высоты области колонки. Значение этого атрибута может быть `top`, `middle` или `bottom`.

```
15125 <define name="style-column-sep-attlist" combine="interleave">
15126     <optional>
15127         <attribute name="style:vertical-align" a:defaultValue="top">
15128             <choice>
15129                 <value>top</value>
15130                 <value>middle</value>
15131                 <value>bottom</value>
15132             </choice>
15133         </attribute>
15134     </optional>
15135 </define>
```

Цвет линии

Атрибут `style:color` используется для определения цвета разделительной линии колонки.

```
15136 <define name="style-column-sep-attlist" combine="interleave">
15137     <optional>
15138         <attribute name="style:color" a:defaultValue="#000000">
15139             <ref name="color"/>
15140         </attribute>
15141     </optional>
15142 </define>
```

15.7.6 Защита

Раздел, помеченный атрибутом `style:protect`, не может быть изменен. Пользовательский интерфейс не должен допускать ручного внесения пользователем любых изменений. Атрибут `style:protect` должен по умолчанию устанавливаться для связанных разделов или индексов. Удаление меток защиты этих разделов доступны пользователю, но при изменении связей или индексов они изменяются.

```
15143 <define name="style-section-properties-attlist" combine="interleave">
15144     <optional>
15145         <attribute name="style:protect" a:defaultValue="false">
15146             <ref name="boolean"/>
15147         </attribute>
15148     </optional>
15149 </define>
```

15.7.7 Несбалансированные колонки текста

Атрибут `text:dont-balance-text-columns` определяет, должно ли содержимое текстовой колонки равномерно распределяться среди всех текстовых колонок.

```
15150 <define name="style-section-properties-attlist" combine="interleave">
15151     <optional>
15152         <attribute name="text:dont-balance-text-columns">
15153             <ref name="boolean"/>
15154         </attribute>
15155     </optional>
15156 </define>
```

15.7.8 Режим записи

Атрибут `style:writing-mode` определяет режим записи, который должен использоваться для раздела (см. раздел 15.5.36 для детального ознакомления).

```
<define name="style-section-properties-attlist" combine="interleave">
    <ref name="common-writing-mode-attlist"/>
</define>
```

15.7.9 Настройка примечаний

Стиль раздела может включать настройку примечаний (см. раздел 14.9.2). В этом случае есть выбор, примечания соответствующего типа отображаются в конце колонок раздела или в конце самого раздела вместо конца колонки страницы или в конце документа.

```
<define name="style-section-properties-elements" combine="interleave">
    <zeroOrMore>
        <ref name="text-notes-configuration"/>
    </zeroOrMore>
</define>
```

15.8 Свойства форматирования таблиц

Свойства, описанные в этом разделе, могут быть включены в стили таблицы (см. раздел 14.12.1) Они включаются в элемент `<style:table-properties>`.

```
<define name="style-table-properties">
    <element name="style:table-properties">
        <ref name="style-table-properties-content"/>
    </element>
</define>

<define name="style-table-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-table-properties-content-strict">
    <ref name="style-table-properties-attlist"/>
    <ref name="style-table-properties-elements"/>
</define>
```

15.8.1 Ширина таблицы

Каждая таблица должна иметь фиксированную ширину. Это определяется с помощью атрибута `style:width`.

Ширина таблицы также может быть определена относительно ширины области, на которой она размещается. В этом случае ширина определяется в процентах с использованием атрибута `style:rel-width`. Пользовательские агенты, которые поддерживают относительную ширину таблицы, могут задавать ее таким образом, но это не обязательно.

Причины, по которым каждая таблица должна иметь заданную и относительную ширину, перечислены ниже.

- Определение ширины таблицы в процентах обычно широко применяется для web-браузеров и других приложений, где процентное значение рассчитывается относительно ширины окна. Но могут возникать проблемы, если процентное значение рассчитывается относительно фиксированной ширины бумаги.
- Относительная ширина также может быть причиной проблем для таких приложений, как приложения для работы с электронными таблицами, где нет требований по подгонке таблицы к странице.

Однако, если приложение поддерживает относительную ширину, то относительно просто запрограммировать приложение на вычисление фиксированной ширины таблицы на основании процентного значения.

```
15179 <define name="style-table-properties-attlist" combine="interleave">
15180     <optional>
15181         <attribute name="style:width">
15182             <ref name="positiveLength"/>
15183         </attribute>
15184     </optional>
15185     <optional>
15186         <attribute name="style:rel-width">
15187             <ref name="percent"/>
15188         </attribute>
15189     </optional>
15190 </define>
```

15.8.2 Выравнивание таблицы

Свойство выравнивание таблицы `table:align` определяет горизонтальное выравнивание таблицы.

Значение для свойства выравнивание таблицы может быть:

- `left` — выравнивание таблицы по левому краю;
- `center` — выравнивание таблицы по центру;
- `right` — выравнивание таблицы по правому краю;
- `margins` — таблица заполняет все пространство между левым и правым полем.

Пользовательские агенты, не поддерживающие значение `margins`, могут рассматривать это значение как `left`.

```
15191 <define name="style-table-properties-attlist" combine="interleave">
15192     <optional>
```

```
15193         <attribute name="table:align">
15194             <choice>
15195                 <value>left</value>
15196                 <value>center</value>
15197                 <value>right</value>
15198                 <value>margins</value>
15199             </choice>
15200         </attribute>
15201     </optional>
15202 </define>
```

15.8.3 Левое и правое поля таблицы

Свойства `fo:margin-left` и `fo:margin-right` определяют дистанцию до таблицы от левой и правой границ. Для детального ознакомления со свойствами левой и правой границ см. раздел 15.5.17. Приложения могут обрабатывать поля таблицы, но это не обязательно.

Таблицы, выравненные по левому полю или центру, игнорируют правые поля, и таблицы, выравненные по правому полю или центру, игнорируют левые поля.

```
15203 <define name="style-table-properties-attlist" combine="interleave">
15204     <ref name="common-horizontal-margin-attlist"/>
15205 </define>
```

15.8.4 Верхнее и нижнее поля таблицы

Свойства `fo:margin-top` и `fo:margin-bottom` определяют расстояние до таблицы сверху и снизу. Для получения более полной информации о свойствах верхних и нижних полей см. раздел 15.5.20.

```
15206 <define name="style-table-properties-attlist" combine="interleave">
15207     <ref name="common-vertical-margin-attlist"/>
15208 </define>
```

15.8.5 Поля таблицы

Свойство `fo:margin` определяет расстояние слева, справа, сверху и снизу в таблице. Для получения более полной информации об этом свойстве см. раздел 15.5.21.

```
15209 <define name="style-table-properties-attlist" combine="interleave">
15210     <ref name="common-margin-attlist"/>
15211 </define>
```

15.8.6 Номер страницы

Если в стиле таблицы задан стиль страницы, который должен быть применен с начала таблицы, то атрибут `style:page-number` определяет номер страницы, который должен быть использован для первой страницы таблицы (см. также раздел 15.5.39).

```
<define name="style-table-properties-attlist" combine="interleave">
    <ref name="common-page-number-attlist"/>
</define>
```

15.8.7 Разрыв до и разрыв после

Свойства `fo:break-before` и `fo:break-after` вставляют в страницу или колонку разрыв до или после таблицы. Для получения более детальной информации об этом свойстве см. раздел 15.5.22.

```
<define name="style-table-properties-attlist" combine="interleave">
    <ref name="common-break-attlist"/>
</define>
```

15.8.8 Фон таблицы и фоновое изображение

Атрибут фона `fo:background-color` и элемент фона `<style:background-image>` определяет свойства фона таблицы. Для получения детальной информации об этом атрибуте и свойстве см. разделы 15.5.23 и 15.5.24.

```
<define name="style-table-properties-attlist" combine="interleave">
    <ref name="common-background-color-attlist"/>
</define>
<define name="style-table-properties-elements" combine="interleave">
    <ref name="style-background-image"/>
</define>
```

15.8.9 Тень таблицы

Стиль `style:shadow` определяет, какой визуальный эффект тени применяется к таблице. Для получения полной информации по этому свойству см. раздел 15.5.28

```
<define name="style-table-properties-attlist" combine="interleave">
    <ref name="common-shadow-attlist"/>
</define>
```

15.8.10 Вместе со следующим

Свойство `fo:keep-with-next` определяет, что таблица, размещенная в абзаце, остается вместе с ним. Для получения полной информации по этому свойству см. раздел 15.5.29.

```
15227 <define name="style-table-properties-attlist" combine="interleave">
15228     <ref name="common-keep-with-next-attlist"/>
15229 </define>
```

15.8.11 Возможность разрыва между строками

Свойство `style:may-break-between-rows` определяет, может ли применяться разрыв страницы внутри таблицы.

```
15230 <define name="style-table-properties-attlist" combine="interleave">
15231     <optional>
15232         <attribute name="style:may-break-between-rows">
15233             <ref name="boolean"/>
15234         </attribute>
15235     </optional>
15236 </define>
```

15.8.12 Свойство модели обрамления

Свойство `table:border-model` определяет, какая модель обрамления используется, когда создается таблица с обрамлением. Существует два типа модели обрамления.

- **Сокращенная модель обрамления**

  Когда две смежные ячейки имеют различные границы, более широкое обрамление отображается как обрамление между ячейками. Каждая ячейка получает половину ширины обрамления.

- **Модель разделенного обрамления**

  Обрамление отображается внутри ячейки, которая определяет обрамление.

Обе модели обрамления, сокращенная и разделенная, подобны моделям обрамления [XSL] и [CSS2]. Они различаются отношением ширины обрамления к ширине строки и столбца.

В OpenDocument высота строки или ширина столбца включает любое пространство, необходимое для отображения обрамления или полей. Это означает, что в то время как ширина и высота области содержимого меньше, чем

ширина столбца и высота строки, суммарная ширина всех столбцов равна полной ширине таблицы.

В XSL и CSS2 ширина столбца или высота строки определяет область содержимого ячейки. Это означает, что сумма ширины всех столбцов меньше ширины всей таблицы.

```
15237 <define name="style-table-properties-attlist" combine="interleave">
15238     <optional>
15239         <attribute name="table:border-model">
15240             <choice>
15241                 <value>collapsing</value>
15242                 <value>separating</value>
15243             </choice>
15244         </attribute>
15245     </optional>
15246 </define>
```

15.8.13 Режим записи

Атрибут `style:writing-mode` определяет режим записи, который должен быть использован для таблицы (см. раздел 15.5.36 для детального ознакомления).

```
15247 <define name="style-table-properties-attlist" combine="interleave">
15248     <ref name="common-writing-mode-attlist"/>
15249 </define>
```

15.8.14 Отображение

Атрибут `table:display` определяет, отображается ли таблица.

```
15250 <define name="style-table-properties-attlist" combine="interleave">
15251     <optional>
15252         <attribute name="table:display">
15253             <ref name="boolean"/>
15254         </attribute>
15255     </optional>
15256 </define>
```

15.9 Свойства форматирования столбцов

Свойства, описанные в этом разделе, могут быть включены в стили столбцов таблицы (см. раздел 14.12.2) Также они могут быть включены в элемент `<style:table-column-properties>`.

```
15257 <define name="style-table-column-properties">
15258     <element name="style:table-column-properties">
15259         <ref name="style-table-column-properties-content"/>
15260     </element>
15261 </define>
15262
15263 <define name="style-table-column-properties-content">
15264     <ref name="style-properties-content"/>
```

```
15265 </define>
15266
15267 <define name="style-table-column-properties-content-strict">
15268     <ref name="style-table-column-properties-attlist"/>
15269     <ref name="style-table-column-properties-elements"/>
15270 </define>
15271
15272 <define name="style-table-column-properties-elements">
15273     <empty/>
15274 </define>
```

15.9.1 Ширина столбца

Каждый столбец таблицы должен иметь фиксированную ширину. Эта ширина определяется атрибутом `style:column-width`.

Ширина столбца может также определяться относительно ширины других столбцов. Приложения, которые поддерживают относительную ширину столбца, могут определить ее таким образом, но это не обязательно.

Относительная ширина определяется свойством `style:rel-column-width`, которое принимает числовое значение, дополненное символом « * ». Если r_c — относительное значение ширины столбца, r_s — сумма относительных значений ширины всех столбцов, и w_s — абсолютная ширина, доступная для всех этих столбцов, то абсолютное значение ширины столбца w_c рассчитывается как $w_c=r_c w_s/r_s$.

```
15275 <define name="style-table-column-properties-attlist" combine="interleave">
15276     <optional>
15277         <attribute name="style:column-width">
15278             <ref name="positiveLength"/>
15279         </attribute>
15280     </optional>
15281     <optional>
15282         <attribute name="style:rel-column-width">
15283             <ref name="relativeLength"/>
15284         </attribute>
15285     </optional>
15286 </define>
```

15.9.2 Оптимальная ширина столбца таблицы

Атрибут `style:use-optimal-column-width` определяет, должна ли ширина столбца быть автоматически пересчитана, если изменилось содержимое столбца.

```
15287 <define name="style-table-column-properties-attlist" combine="interleave">
15288     <optional>
15289         <attribute name="style:use-optimal-column-width">
15290             <ref name="boolean"/>
15291         </attribute>
```

```
    </optional>
</define>
```

15.9.3 Разрыв до и разрыв после

Свойства `fo:break-before` и `fo:break-after` добавляют в страницу или в колонку разрыв до или после столбца таблицы. Для получения более полной информации по этим свойствам см. раздел 15.5.22.

```
<define name="style-table-column-properties-attlist" combine="interleave">
    <ref name="common-break-attlist"/>
</define>
```

15.10 Свойства форматирования строк таблиц

Свойства, рассмотренные в этом разделе, могут быть включены в стили столбцов таблицы (см. раздел 14.12.3). Они включаются в элемент `<style:table-column-properties>`.

```
<define name="style-table-row-properties">
    <element name="style:table-row-properties">
        <ref name="style-table-row-properties-content"/>
    </element>
</define>

<define name="style-table-row-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-table-row-properties-content-strict">
    <ref name="style-table-row-properties-attlist"/>
    <ref name="style-table-row-properties-elements"/>
</define>
```

15.10.1 Высота строки

Свойства `style:row-height` и `style:min-row-height` определяют высоту строки таблицы. По умолчанию, высота строки зависит от высоты наивысшего элемента в строке.

Свойство `style:row-height` определяет фиксированную высоту строки, в то время как свойство `style:min-row-height` определяет фиксированную минимальную высоту.

```
<define name="style-table-row-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:row-height">
            <ref name="positiveLength"/>
        </attribute>
    </optional>
```

```
15317     <optional>
15318         <attribute name="style:min-row-height">
15319             <ref name="nonNegativeLength"/>
15320         </attribute>
15321     </optional>
15322 </define>
```

15.10.2 Оптимальная высота строки таблицы

Атрибут `style:use-optimal-row-height` определяет, должна ли высота строки быть пересчитана автоматически, если изменилось содержимое строки.

```
15323 <define name="style-table-row-properties-attlist" combine="interleave">
15324     <optional>
15325         <attribute name="style:use-optimal-row-height">
15326             <ref name="boolean"/>
15327         </attribute>
15328     </optional>
15329 </define>
```

15.10.3 Фон строки

Атрибут фона `fo:background-color` и элемент фона `<style:background-image>` определяют свойства фона таблицы. Для получения более полной информации об этом атрибуте и элементе см. разделы 15.5.23 и 15.5.24.

```
15330 <define name="style-table-row-properties-attlist" combine="interleave">
15331     <ref name="common-background-color-attlist"/>
15332 </define>
15333 <define name="style-table-row-properties-elements" combine="interleave">
15334     <ref name="style-background-image"/>
15335 </define>
```

15.10.4 Разрыв до и разрыв после

Свойства `fo:break-before` и `fo:break-after` вставляют в страницу или колонку разрыв до или после столбца таблицы. Для получения более полной информации по этим свойствам см. раздел 15.5.22.

```
15336 <define name="style-table-row-properties-attlist" combine="interleave">
15337     <ref name="common-break-attlist"/>
15338 </define>
```

15.10.5 Не разрывать

Свойство `fo:keep-together` используется для определения, должно ли содержимое ячейки таблицы быть неразрывным на некоторой странице или в колонке (если установлено значение `always`) или разрешаются разрывы ячейки

(если установлено значение `auto`) (см. § 7.19.3 [XSL] для детального ознакомления).

```
<define name="style-table-row-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:keep-together">
            <choice>
                <value>auto</value>
                <value>always</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.11 Свойства форматирования ячеек таблиц

Свойства, описанные в данном разделе, могут содержаться в стилях ячеек таблицы (см. раздел 14.12.4). Они содержатся в элементе `<style:table-column-properties>`.

```
<define name="style-table-cell-properties">
    <element name="style:table-cell-properties">
        <ref name="style-table-cell-properties-content"/>
    </element>
</define>

<define name="style-table-cell-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-table-cell-properties-content-strict">
    <ref name="style-table-cell-properties-attlist"/>
    <ref name="style-table-cell-properties-elements"/>
</define>
```

15.11.1 Выравнивание по вертикали

Свойство выравнивания по вертикали `style:vertical-align` используется для определения выравнивания по вертикали текста в ячейке таблицы.

Варианты выравнивания по вертикали следующие:

- `top` — выравнивание текста по верхнему краю ячейки;
- `middle` — выравнивание текста по середине ячейки;
- `bottom` — выравнивание текста по нижнему краю ячейки;
- `automatic` — выравнивание текста определяется приложением.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:vertical-align">
            <choice>
```

```
                <value>top</value>
                <value>middle</value>
                <value>bottom</value>
                <value>automatic</value>
            </choice>
            </attribute>
    </optional>
</define>
```

15.11.2 Источник выравнивания текста

Свойство `style:text-align-source` определяет источник свойства выравнивания текста. Если значение данного атрибута равно `fix`, используется значение свойства `fo:text-align`. Если значение равно `value-type`, выравнивание текста зависит от типа значения ячейки.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:text-align-source">
            <choice>
                <value>fix</value>
                <value>value-type</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.11.3 Направление

Свойство `style:direction` определяет направление символов в ячейке. Наиболее распространенным является направление слева направо (`ltr`). Другое направление — сверху вниз (`ttb`), когда символы в ячейке представляются в виде стопки, но не поворачиваются.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <ref name="common-style-direction-attlist"/>
</define>

<define name="common-style-direction-attlist">
    <optional>
        <attribute name="style:direction">
            <choice>
                <value>ltr</value>
                <value>ttb</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.11.4 Ориентация глифа по вертикали

Свойство `style:glyph-orientation-vertical` определяет ориентацию глифа по вертикали. Свойство определяет угол или автоматический режим. Для запрещения этой настройки может использоваться только значение угла 0.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:glyph-orientation-vertical">
            <choice>
                <value>auto</value>
                <value>0</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.11.5 Тень ячейки

Свойство `style:shadow` определяет, что к ячейке применяется визуальный эффект тени. Полное описание этого свойства приведено в разделе 15.5.28.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <ref name="common-shadow-attlist"/>
</define>
```

15.11.6 Фон ячейки

Атрибут фона `fo:background-color` и элемент фона `<style:background-image>` определяют свойства фона ячейки таблицы. Детальная информация об этих атрибуте и элементе приведена в разделах 15.5.23 и 15.5.24.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <ref name="common-background-color-attlist"/>
</define>
<define name="style-table-cell-properties-elements" combine="interleave">
    <ref name="style-background-image"/>
</define>
```

15.11.7 Обрамление ячейки

Атрибуты обрамления `fo:border`, `fo:border-top`, `fo:border-bottom`, `fo:border-left` и `fo:border-right` определяют свойства обрамления ячейки таблицы. Детальная информация об этих атрибутах приведена в разделе 15.5.25.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <ref name="common-border-attlist"/>
</define>
```

15.11.8 Диагональные линии

Ячейки электронных таблиц также могут иметь диагональные линии, которые рассматриваются приложениями как обрамления.

`style:diagonal-tl-br` определяет стиль «border», используемый для диагонали «верхний левый угол — правый нижний угол» (см. раздел 15.5.25 для получения детальной информации). В случае двойной линии `style:diagonal-bl-tr-widths` позволяет определить ширину внутренних и внешних линий и расстояние между ними (см. раздел 15.5.26 для получения детальной информации).

`style:diagonal-bl-tr` и `style:diagonal-tl-br-widths` определяют те же свойства для диагонали «нижний левый угол — верхний правый угол».

```
15421 <define name="style-table-cell-properties-attlist" combine="interleave">
15422     <optional>
15423         <attribute name="style:diagonal-tl-br">
15424             <ref name="string"/>
15425         </attribute>
15426     </optional>
15427     <optional>
15428         <attribute name="style:diagonal-tl-br-widths">
15429             <ref name="borderWidths"/>
15430         </attribute>
15431     </optional>
15432     <optional>
15433         <attribute name="style:diagonal-bl-tr">
15434             <ref name="string"/>
15435         </attribute>
15436     </optional>
15437     <optional>
15438         <attribute name="style:diagonal-bl-tr-widths">
15439             <ref name="borderWidths"/>
15440         </attribute>
15441     </optional>
15442 </define>
```

15.11.9 Толщина линии обрамления

Атрибуты ширины линии обрамления `style:border-line-width`, `style:border-line-width-top`, `style:border-line-width-bottom`, `style:border-line-width-left` и `style:border-line-width-right` определяют свойства линий обрамления страницы. Детальная информация об этих атрибутах приведена в разделе 15.5.26.

```
15443 <define name="style-table-cell-properties-attlist" combine="interleave">
15444     <ref name="common-border-line-width-attlist"/>
15445 </define>
```

15.11.10 Отступы от содержимого

Атрибуты отступов от содержимого `fo:padding`, `fo:padding-top`, `fo:padding-bottom`, `fo:padding-left` и `fo:padding-right` определяют свойства отступов от содержимого ячейки таблицы. Детальная информация об этих атрибутах приведена в разделе 15.5.27.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <ref name="common-padding-attlist"/>
</define>
```

15.11.11 Перенос по словам

Свойство `fo:wrap-option` определяет, переносится ли текст в ячейке таблицы по словам. Детальная информация приведена в § 7.5.13 [XSL]. Если перенос по словам запрещен, приложение определяет, будет ли текст видимым или скрытым. Если текст приложением скрывается, то должен обеспечиваться механизм прокрутки для доступа к тексту. Это подобно установке свойства `fo:overflow` в значение `auto`. См. также § 7.20.2 [XSL].

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:wrap-option">
            <choice>
                <value>no-wrap</value>
                <value>wrap</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.11.12 Угол поворота

Свойство `style:rotation-angle` определяет угол поворота содержимого ячейки в градусах.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <ref name="common-rotation-angle-attlist"/>
</define>

<define name="common-rotation-angle-attlist">
    <optional>
        <attribute name="style:rotation-angle">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.11.13 Выравнивание при повороте

Свойство `style:rotation-align` определяет, как край текста будет выравнен в ячейке после поворота. Имеется четыре варианта выравнивания: `none`, `bottom`, `top` или `center`.

Выравнивание	*Текст...*	*Обрамление и фон...*
Нет	Повернут	Неизменны
Нижний край ячейки	Повернут и может частично перекрываться с другими ячейками, если текст длиннее ячейки	Располагаются параллельно тексту, посредством чего верхний и нижний края изображаются в исходной позиции ячейки
Верхний край ячейки		
Центр ячейки		

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:rotation-align">
            <choice>
                <value>none</value>
                <value>bottom</value>
                <value>top</value>
                <value>center</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.11.14 Защита ячейки

Свойство `style:cell-protect` определяет, каким образом защищена ячейка.

Атрибут рассматривается только в том случае, если текущая таблица защищена (см. раздел 8.1.1). Атрибут может принимать значения `none`, `hidden-and-protected` или разделенный пробелами список, содержащий значения `protected` или `formula-hidden`.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:cell-protect">
            <choice>
                <value>none</value>
                <value>hidden-and-protected</value>
                <list>
                    <oneOrMore>
                        <choice>
                            <value>protected</value>
                            <value>formula-hidden</value>
                        </choice>
                    </oneOrMore>
                </list>
```

```
15496            </choice>
15497        </attribute>
15498    </optional>
15499 </define>
```

15.11.15 Печатать содержимое

Свойство `style:print-content` определяет, выводится ли на печать содержимое ячейки.

```
15500 <define name="style-table-cell-properties-attlist" combine="interleave">
15501     <optional>
15502         <attribute name="style:print-content">
15503             <ref name="boolean"/>
15504         </attribute>
15505     </optional>
15506 </define>
```

15.11.16 Десятичные разряды

Атрибут `style:decimal-places` определяет максимальное количество отображаемых десятичных разрядов, если числа отформатированы стилем данных, не имеющим собственных настроек количества десятичных разрядов см. также раздел 14.7.9).

Свойство рассматривается, только если оно содержится в стиле по умолчанию (см. раздел 14.2).

```
15507 <define name="style-table-cell-properties-attlist" combine="interleave">
15508     <optional>
15509         <attribute name="style:decimal-places">
15510             <ref name="nonNegativeInteger"/>
15511         </attribute>
15512     </optional>
15513 </define>
```

15.11.17 Повторять содержимое

Свойство `style:repeat-content` определяет, отображается ли содержимое ячейки столько раз, сколько остается в ячейке незанятого места в направлении вывода. Отображаются только полные вхождения повторяемого текста. Свойство не изменяет содержимое ячейки, содержащее разрыв строки. Это свойство используется, например, для заполнения ячейки таблицы символами «–» или «x», чтобы не могли быть введены другие данные.

```
15514 <define name="style-table-cell-properties-attlist" combine="interleave">
15515     <optional>
15516         <attribute name="style:repeat-content">
15517             <ref name="boolean"/>
```

```
        </attribute>
    </optional>
</define>
```

15.11.18 Подгонять по размеру

Свойство `style:shrink-to-fit` определяет, подгоняется ли, если необходимо, содержимое ячейки по ее размеру. Подгонка означает, что уменьшается размер шрифта так, что весь текст помещается в ячейке. Данное свойство не приводит к изменениям в ячейках, содержимое которых уже помещается в ячейке.

```
<define name="style-table-cell-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:shrink-to-fit">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.12 Свойства стилей уровней списков

Свойства, описанные в данном разделе, могут содержаться в различных элементах стилей уровней списков (см. раздел 14.10). Они содержатся в элементе `<style:list-level-properties>`.

```
<define name="style-list-level-properties">
    <element name="style:list-level-properties">
        <ref name="style-list-level-properties-content"/>
    </element>
</define>

<define name="style-list-level-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-list-level-properties-content-strict">
    <ref name="style-list-level-properties-attlist"/>
    <ref name="style-list-level-properties-elements"/>
</define>

<define name="style-list-level-properties-elements">
    <empty/>
</define>
```

Выравнивание маркера

Атрибут `fo:text-align` определяет выравнивание маркера (номера) по горизонтали в поле, определенном атрибутом `text:min-label-width` (см. также раздел 15.5.5).

```
<define name="style-list-level-properties-attlist" combine="interleave">
    <ref name="common-text-align"/>
</define>
```

Начальный отступ

Атрибут `text:space-before` определяет пространство, включаемое до номера для всех абзацев данного уровня. Если абзац имеет отступ слева больше 0, фактическая позиция блока маркера списка — ширина левого поля, увеличенная на значение начального отступа.

Этот атрибут может быть связан с элементом набора пунктов, содержащимся в элементе `<text:list-level-style-*>`.

Значение атрибута является абсолютным. Это означает, что когда вычисляется позиция маркера, рассматривается только значение начального отступа текущего уровня. Значение начального отступа нижних уровней не влияет на позицию маркера.

```
<define name="style-list-level-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:space-before">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

Минимальная ширина маркера

Атрибут `text:min-label-width` определяет минимальную ширину маркера.

Этот атрибут может быть связан с элементом набора пунктов, содержащимся в элементе `<text:list-level-style-*>`.

Маркер может быть выравнен по горизонтали с заданной шириной с использованием свойства `fo:text-align`. См. более детальную информацию о приведенном выше атрибуте выравнивание маркера.

```
<define name="style-list-level-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:min-label-width">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

Минимальное расстояние от маркера

Атрибут `text:min-label-distance` определяет минимальное расстояние между номером и текстом пункта списка.

Этот атрибут может быть связан с элементом набора пунктов, содержащимся в элементах `<text:list-level-style-*>`.

```
<define name="style-list-level-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:min-label-distance">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

Название шрифта

Атрибут `style:font-name` определяет название шрифта, используемого для отображения символа маркера списка (см. раздел 15.4.13).

```
<define name="style-list-level-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:font-name">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

Размер изображения

Размер изображения определяется следующими атрибутами:

```
<define name="style-list-level-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:width">
            <ref name="positiveLength"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="fo:height">
            <ref name="positiveLength"/>
        </attribute>
    </optional>
</define>
```

Выравнивание по вертикали

Выравнивание изображения по вертикали определяется свойствами `style:vertical-pos` и `style:vertical-rel`. Детальная информация приведена в разделах 15.27.11 и 15.27.12.

```
<define name="style-list-level-properties-attlist" combine="interleave">
```

```
15590     <ref name="common-vertical-rel-attlist"/>
15591     <ref name="common-vertical-pos-attlist"/>
15592 </define>
```

15.13 Свойства контурной линии

Для определения характеристик линий графических объектов используются следующие свойства контурной линии, доступные для графических объектов, содержащихся во всех видах приложений:

- стиль;
- штрих;
- толщина;
- цвет;
- начальный маркер;
- конечный маркер;
- ширина начального маркера;
- ширина конечного маркера;
- центрировать начальный маркер;
- центрировать конечный маркер;
- непрозрачность;
- соединение.

Свойства, описанные в данном разделе, могут содержатся в элементах стилей `<style:style>` семейства `graphic` или `presentation`. Они содержатся в элементе `<style:graphic-properties>`.

15.13.1 Стиль контурной линии

Атрибут `draw:stroke` определяет стиль контурной линии текущего объекта. Значение `none` означает, что контурная линия не изображается, а значение `solid` означает, что изображается сплошная контурная линия. Если значение равно `dash`, контурная линия изображается в зависимости от значения свойства `draw:stroke-dash`.

```
15593 <define name="style-graphic-properties-attlist" combine="interleave">
15594     <optional>
15595         <attribute name="draw:stroke">
15596             <choice>
15597                 <value>none</value>
15598                 <value>dash</value>
```

```
                <value>solid</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.13.2 Штрих

Атрибут `draw:stroke-dash` определяет стиль штриха, используемый для контурной линии. Для получения информации о стилях штрихов см. раздел 14.14.7.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:stroke-dash">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

15.13.3 Составные штрихи

Атрибут `draw:stroke-dash-names` определяет список стилей штрихов, который используется в дополнение к штриху, определенному атрибутом `draw:stroke-dash`. Для получения информации об атрибуте `draw:stroke-dash` см. раздел 15.13.2, о стилях штрихов — раздел 14.14.7.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:stroke-dash-names">
            <ref name="styleNameRefs"/>
        </attribute>
    </optional>
</define>
```

15.13.4 Толщина

Атрибут `svg:stroke-width` определяет толщину контурной линии текущего объекта.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="svg:stroke-width">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.13.5 Цвет

Атрибут `svg:stroke-color` определяет цвет контурной линии текущего объекта.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="svg:stroke-color">
            <ref name="color"/>
        </attribute>
    </optional>
</define>
```

15.13.6 Начальный маркер

Атрибут `draw:marker-start` определяет начальный маркер линии, являющийся траекторией и присоединяемый к началу контурной линии. Информация о маркерах приведена в разделе 14.14.6.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:marker-start">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

15.13.7 Конечный маркер

Атрибут `draw:marker-end` определяет конечный маркер линии, являющийся траекторией и присоединяемый к концу контурной линии. Информация о маркерах приведена в разделе 14.14.6.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:marker-end">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

15.13.8 Ширина начального маркера

Атрибут `draw:marker-start-width` определяет ширину маркера начала контурной линии.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:marker-start-width">
            <ref name="length"/>
        </attribute>
    </optional>
```

```
</define>
```

15.13.9 Ширина конечного маркера

Атрибут `draw:marker-end-width` определяет ширину маркера конца контурной линии.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:marker-end-width">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.13.10 Центрировать начальный маркер

Атрибут `draw:marker-start-center` определяет, центрируется ли начальный маркер по началу контурной линии.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:marker-start-center">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.13.11 Центрировать конечный маркер

Атрибут `draw:marker-end-center` определяет, центрируется ли конечный маркер по концу контурной линии.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:marker-end-center">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.13.12 Непрозрачность

Атрибут `svg:stroke-opacity` определяет непрозрачность контурной линии. Атрибут может принимать численные значения от 0 (полностью прозрачный) до 1 (полностью непрозрачный) или значения в процентах.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="svg:stroke-opacity">
            <choice>
                <data type="double">
```

```
                    <param name="minInclusive">0</param>
                    <param name="maxInclusive">1</param>
                </data>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.13.13 Соединение линий

Атрибут `draw:stroke-linejoin` определяет фигуру на углах траектории или другой графической фигуры, образуемой контурными линиями. Атрибут принимает такие же значения, как и атрибут `stroke-linejoin` [SVG], за исключением того, что дополнительно к значениям, поддерживаемым SVG, он может принимать значение `middle`, которое означает, что между соединениями используется среднее значение.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:stroke-linejoin">
            <choice>
                <value>miter</value>
                <value>round</value>
                <value>bevel</value>
                <value>middle</value>
                <value>none</value>
                <value>inherit</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.14 Свойства заполнения

Приведенные ниже свойства заполнения используются для определения характеристик заполнения графического объекта. Они доступны для графических объектов, содержащихся во всех видах приложений;

- стиль;
- цвет;
- градиент;
- количество шагов градиента;
- штриховка;
- сплошная штриховка;
- растровое изображение;
- непрозрачность;

- правило заполнения.

15.14.1 Стиль заполнения

Атрибут draw:fill определяет стиль заполнения графического объекта. Незамкнутый графический объект, такой как траектория без closepath на конце, не будет заполнен. Операция заполнения автоматически не замыкает все открытые подтраектории перед рисованием заполнения, соединяя последнюю точку подтраектории с первой точкой. Атрибут имеет следующие значения:

- none — графический объект не заполняется;
- solid — графический объект заполняется цветом, определенным атрибутом draw:fill-color;
- bitmap — графический объект заполняется растровым изображением, определенным атрибутом draw:fill-image-name;
- gradient — графический объект заполняется градиентом, определенным атрибутом draw:fill-gradient-name;
- hatch — графический объект заполняется штриховкой, определенной атрибутом draw:fill-hatch-name.

```
15701 <define name="style-graphic-fill-properties-attlist" combine="interleave">
15702     <optional>
15703         <attribute name="draw:fill">
15704             <choice>
15705                 <value>none</value>
15706                 <value>solid</value>
15707                 <value>bitmap</value>
15708                 <value>gradient</value>
15709                 <value>hatch</value>
15710             </choice>
15711         </attribute>
15712     </optional>
15713 </define>
```

15.14.2 Цвет

Атрибут draw:fill-color определяет цвет заполнения графического объекта. Он используется, только если атрибут draw:fill имеет значение solid.

```
15714 <define name="style-graphic-fill-properties-attlist" combine="interleave">
15715     <optional>
15716         <attribute name="draw:fill-color">
15717             <ref name="color"/>
15718         </attribute>
15719     </optional>
```

```
</define>
```

15.14.3 Вспомогательный цвет заполнения

Атрибут `draw:secondary-fill-color` определяет вспомогательный цвет заполнения. Он может быть использован как цвет заполнения для выдавливания.

```
<define name="style-graphic-fill-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:secondary-fill-color">
            <ref name="color"/>
        </attribute>
    </optional>
</define>
```

15.14.4 Градиент

Атрибут `draw:fill-gradient-name` определяет стиль градиента, который используется для заполнения графического объекта. Он используется, только если атрибут `draw:fill` имеет значение `gradient`. Информация о градиентах приведена в разделах 14.14.1 и 14.14.2.

```
<define name="style-graphic-fill-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:fill-gradient-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

15.14.5 Количество шагов градиента

Если для заполнения применяется градиент, то атрибут `draw:gradient-step-count` позволяет установить фиксированное число шагов цветовой интерполяции градиента. По умолчанию количество шагов вычисляется автоматически на основании размера и разрешения заполняемой области.

Количество шагов, меньшее 3, недопустимо, так как интерполяция в таком случае невозможна. Значения, большие 256, могут не поддерживаться или приводить к потере производительности.

```
<define name="style-graphic-fill-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:gradient-step-count">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.14.6 Штриховка

Атрибут `draw:fill-hatch-name` определяет стиль штриховки, используемый для заполнения. Он используется, только если атрибут `draw:fill` имеет значение `hatch`. Информация о штриховках приведена в разделе 14.14.3.

```
15742 <define name="style-graphic-fill-properties-attlist" combine="interleave">
15743     <optional>
15744         <attribute name="draw:fill-hatch-name">
15745             <ref name="styleNameRef"/>
15746         </attribute>
15747     </optional>
15748 </define>
```

15.14.7 Сплошная штриховка

Атрибут `draw:fill-hatch-solid` определяет, является ли фон штриховки сплошным или прозрачным.

```
15749 <define name="style-graphic-fill-properties-attlist" combine="interleave">
15750     <optional>
15751         <attribute name="draw:fill-hatch-solid">
15752             <ref name="boolean"/>
15753         </attribute>
15754     </optional>
15755 </define>
```

15.14.8 Заполняющее изображение

Атрибут `draw:fill-image-name` определяет изображение, используемое для заполнения. Он используется, только если атрибут `draw:fill` имеет значение `bitmap`. Информация о заполняющих изображения приведена в разделе 14.14.4.

```
15756 <define name="style-graphic-fill-properties-attlist" combine="interleave">
15757     <optional>
15758         <attribute name="draw:fill-image-name">
15759             <ref name="styleNameRef"/>
15760         </attribute>
15761     </optional>
15762 </define>
```

15.14.9 Стиль визуализации заполняющего изображения

Если для заполнения используется растровое изображение, то оно может быть отрисовано без изменений, с растягиванием по заданным размерам области заполнения или размножено по всей области. Атрибут `style:repeat` определяет, каким образом будет обработано растровое изображение.

Атрибут может принимать значения `no-repeat`, `repeat` или `stretch`.

```
<define name="style-graphic-fill-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:repeat">
            <choice>
                <value>no-repeat</value>
                <value>repeat</value>
                <value>stretch</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.14.10 Размер заполняющего изображения

Если для заполнения используется изображение, необязательные атрибуты `draw:fill-image-width` и `draw:fill-image-height` могут быть использованы для изменения логического размера данных исходного изображения. Эти атрибуты игнорируются, если атрибут `style:repeat` имеет значение `stretch`.

```
<define name="style-graphic-fill-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:fill-image-width">
            <choice>
                <ref name="length"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="draw:fill-image-height">
            <choice>
                <ref name="length"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.14.11 Опорная точка плитки заполняющего изображения

Если для заполнения используется изображение, атрибуты `draw:fill-image-ref-point`, `draw:fill-image-ref-point-x` и `draw:fill-image-ref-point-y` определяют опорную точку изображения. Атрибут `draw:fill-image-ref-point` определяет позицию выравнивания изображения в заполняемой области, атрибуты `draw:fill-image-ref-point-x` и `draw:fill-image-ref-point-y` определяют горизонтальное и вертикальное смещение в процентах относительно ширины и высоты изображения. Если выравнивание и смещение

определены одновременно, то изображение сначала выравнивается, а затем смещается.

Эти атрибуты рассматриваются, только если текущее значение атрибута `style:repeat` равно `repeat`.

```
<define name="style-graphic-fill-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:fill-image-ref-point-x">
            <ref name="percent"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="draw:fill-image-ref-point-y">
            <ref name="percent"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="draw:fill-image-ref-point">
            <choice>
                <value>top-left</value>
                <value>top</value>
                <value>top-right</value>
                <value>left</value>
                <value>center</value>
                <value>right</value>
                <value>bottom-left</value>
                <value>bottom</value>
                <value>bottom-right</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.14.12 Смещение плитки заполняющего изображения

Если для заполнения используется изображение, то атрибут `draw:tile-repeat-offset` определяет смещение каждой плитки относительно предыдущей плитки. Этот атрибут рассматривается, только если текущее значение атрибута `style:repeat` равно `repeat`. Значение этого атрибута — процентное, представляющее смещение повтора плитки относительно ее длины или ширины, следующее за одним из слов `horizontal` или `vertical`.

```
<define name="style-graphic-fill-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:tile-repeat-offset"/>
    </optional>
</define>
```

Пример — Смещение плитки:

```
<style:graphic-properties draw:tile-repeat-offset="50% horizontal"/>
```

15.14.13 Непрозрачность и линейная непрозрачность

Заполняемая область графического объекта может быть полностью, линейно или градиентно непрозрачной. Полная и линейная непрозрачность выбираются с использованием атрибута `draw:opacity`, в то время как градиентная непрозрачность выбирается с использованием атрибута `draw:opacity-name`.

Атрибут `draw:opacity` запрещает какие-либо другие эффекты непрозрачности или множество линейных непрозрачностей заполняемой области графического объекта.

```
<define name="style-graphic-fill-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:opacity">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

15.14.14 Градиентная непрозрачность

Атрибут `draw:opacity-name` определяет непрозрачный градиент, характеризующий непрозрачность области заполнения графического объекта. Когда применяется непрозрачный градиент, непрозрачность интерполируется, как определено в базовом стиле непрозрачного градиента. Этот стиль заполнения визуализируется независимо от других стилей заполнения, таких как градиент, изображение и штриховка. Для получения информации о непрозрачных градиентах см. раздел 14.14.5.

Значение данного атрибута перекрывает атрибут `draw:opacity`.

```
<define name="style-graphic-fill-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:opacity-name">
            <ref name="styleNameRef"/>
        </attribute>
    </optional>
</define>
```

15.14.15 Правило заполнения

Атрибут `svg:fill-rule` определяет алгоритм, который будет использован для определения того, какие части полотна включены внутрь фигуры. Для получения более подробной информации см. § 11.3 [SVG].

```
<define name="style-graphic-fill-properties-attlist" combine="interleave">
    <optional>
```

```
15840        <attribute name="svg:fill-rule">
15841            <choice>
15842                <value>nonzero</value>
15843                <value>evenodd</value>
15844            </choice>
15845        </attribute>
15846    </optional>
15847 </define>
```

15.14.16 Цвет символов

Атрибут `draw:symbol-color` определяет цвет, используемый для отображения символов, содержащихся в графическом объекте. Это могут быть, например, стрелки, отображаемые в элементе управления.

```
15848 <define name="style-graphic-properties-attlist" combine="interleave">
15849     <optional>
15850         <attribute name="draw:symbol-color">
15851             <ref name="color"/>
15852         </attribute>
15853     </optional>
15854 </define>
```

15.15 Свойства анимации текста

Графические объекты, содержащие текст и текстовые блоки, могут иметь дополнительные свойства анимации текста. Эти свойства всегда приводят к анимации всего текста или текстового фрейма. Анимацию текста определяют следующие атрибуты:

- анимация;
- направление анимации;
- запуск анимации внутри;
- остановка анимации внутри;
- повтор анимации;
- задержка анимации;
- шаги анимации.

Эти свойства доступны графическим объектам, содержащимся во всех видах приложений.

15.15.1 Анимация

Атрибут `text:animation` определяет используемый для текста тип анимации.

Атрибут может принимать следующие значения:

- `none` — запрещает анимацию текста;
- `scroll` — прокручивает текст от одной стороны к другой;
- `alternate` — прокручивает текст от одной стороны к другой и обратно;
- `slide` — прокручивает текст от одной стороны до исходной позиции и останавливает в ней.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:animation">
            <choice>
                <value>none</value>
                <value>scroll</value>
                <value>alternate</value>
                <value>slide</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.15.2 Направление анимации

Атрибут `text:animation-direction` определяет направление прокрутки анимированного текста.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:animation-direction">
            <choice>
                <value>left</value>
                <value>right</value>
                <value>up</value>
                <value>down</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.15.3 Запуск анимации внутри

Если атрибут `text:animation-start-inside` принимает значение `true`, то текст начинает движение внутри фигуры. Если `false`, то анимация текста начинается за пределами прямоугольника, обрамляющего фигуру.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:animation-start-inside">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.15.4 Остановка анимации внутри

Если атрибут `text:animation-stop-inside` принимает значение `true`, то текст останавливается внутри фигуры. Если `false`, то анимация текста завершается за пределами прямоугольника, обрамляющего фигуру.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:animation-stop-inside">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.15.5 Повтор анимации

Атрибут `text:animation-repeat` определяет количество повторений анимации. Если значение атрибута равно `0`, анимация повторяется бесконечно.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:animation-repeat">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.15.6 Задержка анимации

Атрибут `text:animation-delay` определяет величину задержки, после которой начинается анимация. Значение данного атрибута должно соответствовать формату периода времени, описанному в § 3.2.6 [xmlschema-2].

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:animation-delay">
            <ref name="duration"/>
        </attribute>
    </optional>
</define>
```

15.15.7 Шаги анимации

Атрибут `text:animation-steps` определяет расстояние, на которое текст перемещается на каждом шаге прокрутки.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="text:animation-steps">
            <ref name="length"/>
        </attribute>
    </optional>
```

```
15913 </define>
```

15.16 Свойства текста и выравнивания текста

Графические объекты, содержащие текст и текстовые блоки, могут иметь дополнительные свойства, определяющие выравнивание текста в графическом объекте. Эти свойства доступны графическим объектам, содержащимся во всех видах приложений.

15.16.1 Автоматически увеличивать ширину и высоту

Атрибуты `draw:auto-grow-width` и `draw:auto-grow-height` определяют, увеличиваются ли автоматически ширина и высота графического объекта при добавлении текста в графическом объекте. Эти атрибуты обычно рассматриваются только для текстовых блоков.

```
15914 <define name="style-graphic-properties-attlist" combine="interleave">
15915     <optional>
15916         <attribute name="draw:auto-grow-width">
15917             <ref name="boolean"/>
15918         </attribute>
15919     </optional>
15920     <optional>
15921         <attribute name="draw:auto-grow-height">
15922             <ref name="boolean"/>
15923         </attribute>
15924     </optional>
15925 </define>
```

15.16.2 Подгонять по размеру

Атрибут `draw:fit-to-size` определяет, растягивается ли текстовое содержимое графического объекта до заполнения всего объекта. Если атрибут принимает значение `true`, текстовое содержимое растягивается.

```
15926 <define name="style-graphic-properties-attlist" combine="interleave">
15927     <optional>
15928         <attribute name="draw:fit-to-size">
15929             <ref name="boolean"/>
15930         </attribute>
15931     </optional>
15932 </define>
```

15.16.3 Подгонять по контуру

Атрибут `draw:fit-to-contour` определяет, растягивается ли текстовое содержимое графического объекта до заполнения контура объекта. Если атрибут принимает значение `true`, текстовое содержимое растягивается.

```
15933 <define name="style-graphic-properties-attlist" combine="interleave">
15934     <optional>
15935         <attribute name="draw:fit-to-contour">
15936             <ref name="boolean"/>
15937         </attribute>
15938     </optional>
15939 </define>
```

15.16.4 Вертикальное выравнивание области текста

Атрибут `draw:textarea-vertical-align` определяет вертикальное выравнивание области текста внутри фигуры.

```
15940 <define name="style-graphic-properties-attlist" combine="interleave">
15941     <optional>
15942         <attribute name="draw:textarea-vertical-align">
15943             <choice>
15944                 <value>top</value>
15945                 <value>middle</value>
15946                 <value>bottom</value>
15947                 <value>justify</value>
15948             </choice>
15949         </attribute>
15950     </optional>
15951 </define>
```

15.16.5 Выравнивание области текста по горизонтали

Атрибут `draw:textarea-horizontal-align` определяет горизонтальное выравнивание области текста внутри фигуры.

```
15952 <define name="style-graphic-properties-attlist" combine="interleave">
15953     <optional>
15954         <attribute name="draw:textarea-horizontal-align">
15955             <choice>
15956                 <value>left</value>
15957                 <value>center</value>
15958                 <value>right</value>
15959                 <value>justify</value>
15960             </choice>
15961         </attribute>
15962     </optional>
15963 </define>
```

15.16.6 Перенос слов

Атрибут `fo:wrap-option` определяет, переносится ли текст в фигуре по словам.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:wrap-option">
            <choice>
                <value>no-wrap</value>
                <value>wrap</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.16.7 Стили списков

Элемент `<text:list-style>`, как описано в разделе 14.10, определяет стиль списка, применяемый к абзацам, содержащимся в текстовых блоках. Хотя стиль списка имеет имя, он не отображается в пользовательском интерфейсе, даже если содержащий его графический стиль является общим.

Включение элемента стиля списка в графический стиль имеет ту же семантику, что и добавление атрибута `style:list-style-name` (см. раздел 14.1) в стиль, который ссылается на стиль списка, объявленный за пределами графического стиля. Включение элемента стиля списка требуется в тех случаях, когда общий графический стиль должен быть связан с автоматическим стилем списка.

На стили списков, содержащиеся в графическом стиле, могут ссылаться другие графические стили с использованием атрибута `style:list-style-name`.

```
<define name="style-graphic-properties-elements" combine="interleave">
    <optional>
        <ref name="text-list-style"/>
    </optional>
</define>
```

15.17 Свойства цвета

Графические объекты, отображающие растровые изображения, могут иметь дополнительные свойства, которые регулируют параметры цвета растрового изображения. Эти свойства доступны графическим объектам, содержащимся во всех видах приложений.

15.17.1 Цветовой режим

Атрибут `draw:color-mode` влияет на вывод цветов исходных растровых изображений.

```
15979 <define name="style-graphic-properties-attlist" combine="interleave">
15980     <optional>
15981         <attribute name="draw:color-mode">
15982             <choice>
15983                 <value>greyscale</value>
15984                 <value>mono</value>
15985                 <value>watermark</value>
15986                 <value>standard</value>
15987             </choice>
15988         </attribute>
15989     </optional>
15990 </define>
```

15.17.2 Инверсия цвета

Атрибут `draw:color-inversion` определяет, инвертируются ли цвета в графическом изображении.

```
15991 <define name="style-graphic-properties-attlist" combine="interleave">
15992     <optional>
15993         <attribute name="draw:color-inversion">
15994             <ref name="boolean"/>
15995         </attribute>
15996     </optional>
15997 </define>
```

15.17.3 Регулировка яркости

Атрибут `draw:luminance` определяет процентное значение со знаком, которое влияет на выходную яркость растрового изображения.

```
15998 <define name="style-graphic-properties-attlist" combine="interleave">
15999     <optional>
16000         <attribute name="draw:luminance">
16001             <ref name="percent"/>
16002         </attribute>
16003     </optional>
16004 </define>
```

15.17.4 Регулировка контрастности

Атрибут `draw:contrast` определяет процентное значение со знаком, которое влияет на выходную контрастность растрового изображения.

```
16005 <define name="style-graphic-properties-attlist" combine="interleave">
16006     <optional>
16007         <attribute name="draw:contrast">
16008             <ref name="percent"/>
16009         </attribute>
```

```
16010     </optional>
16011 </define>
```

15.17.5 Регулировка гаммы

Атрибут `draw:gamma` определяет значение, влияющее на выходную гамму растрового изображения.

```
16012 <define name="style-graphic-properties-attlist" combine="interleave">
16013     <optional>
16014         <attribute name="draw:gamma">
16015             <ref name="percent"/>
16016         </attribute>
16017     </optional>
16018 </define>
```

15.17.6 Регулировка красного

Атрибут `draw:red` определяет процентное значение со знаком, которое влияет на выходную интенсивность красного цвета растрового изображения.

```
16019 <define name="style-graphic-properties-attlist" combine="interleave">
16020     <optional>
16021         <attribute name="draw:red">
16022             <ref name="percent"/>
16023         </attribute>
16024     </optional>
16025 </define>
```

15.17.7 Регулировка зеленого

Атрибут `draw:green` определяет процентное значение со знаком, которое влияет на выходную интенсивность зеленого цвета растрового изображения.

```
16026 <define name="style-graphic-properties-attlist" combine="interleave">
16027     <optional>
16028         <attribute name="draw:green">
16029             <ref name="percent"/>
16030         </attribute>
16031     </optional>
16032 </define>
```

15.17.8 Регулировка синего

Атрибут `draw:blue` определяет процентное значение со знаком, которое влияет на выходную интенсивность синего цвета растрового изображения.

```
16033 <define name="style-graphic-properties-attlist" combine="interleave">
16034     <optional>
16035         <attribute name="draw:blue">
16036             <ref name="percent"/>
16037         </attribute>
16038     </optional>
```

```
</define>
```

15.17.9 Регулировка непрозрачности

Атрибут `draw:image-opacity` регулирует непрозрачность изображения. Значения могут принимать значения от 0 % до 100 % (см. также раздел 15.14.13).

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:image-opacity">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

15.18 Свойства тени

Большинство графических объектов могут иметь тень. Следующие атрибуты определяют, каким образом отображается тень. Эти свойства доступны графическим объектам, содержащимся во всех видах приложений.

15.18.1 Тень

Атрибут `draw:shadow` разрешает или запрещает отображение тени.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:shadow">
            <choice>
                <value>visible</value>
                <value>hidden</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.18.2 Смещение

Атрибуты `draw:shadow-offset-x` и `draw:shadow-offset-y` используются для отображения тени. Копия фигуры отображается позади нее в одном цвете тени (определяется `draw:shadow-color`). Атрибуты смещения определяют смещение между левым верхним краем фигуры и верхним левым краем тени.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:shadow-offset-x">
            <ref name="length"/>
        </attribute>
    </optional>
    <optional>
```

```
16064         <attribute name="draw:shadow-offset-y">
16065             <ref name="length"/>
16066         </attribute>
16067     </optional>
16068 </define>
```

15.18.3 Цвет

Атрибут `draw:shadow-color` определяет цвет, в котором отображается тень.

```
16069 <define name="style-graphic-properties-attlist" combine="interleave">
16070     <optional>
16071         <attribute name="draw:shadow-color">
16072             <ref name="color"/>
16073         </attribute>
16074     </optional>
16075 </define>
```

15.18.4 Непрозрачность

Атрибут `draw:shadow-opacity` определяет непрозрачность отображения тени. Значение атрибута выражается в процентах.

```
16076 <define name="style-graphic-properties-attlist" combine="interleave">
16077     <optional>
16078         <attribute name="draw:shadow-opacity">
16079             <ref name="percent"/>
16080         </attribute>
16081     </optional>
16082 </define>
```

15.19 Свойства соединительных элементов

Свойства, описанные в данном разделе, являются специфичными для соединительных графических объектов. Эти свойства доступны соединительным графическим объектам, содержащимся во всех видах приложений.

15.19.1 Отступ начала линии

Для стандартных соединительных элементов атрибуты `draw:start-line-spacing-horizontal` и `draw:start-line-spacing-vertical` увеличивают длину выходящей из начальной фигуры линии. Для соединительных элементов линий эти атрибуты определяют абсолютную длину выходящей из начальной фигуры линии. Для других типов соединительных элементов атрибуты игнорируются.

```
16083 <define name="style-graphic-properties-attlist" combine="interleave">
```

```
16084     <optional>
16085         <attribute name="draw:start-line-spacing-horizontal">
16086             <ref name="distance"/>
16087         </attribute>
16088     </optional>
16089     <optional>
16090         <attribute name="draw:start-line-spacing-vertical">
16091             <ref name="distance"/>
16092         </attribute>
16093     </optional>
16094 </define>
```

15.19.2 Отступ конца линии

Для стандартных соединительных элементов атрибуты `draw:end-line-spacing-horizontal` и `draw:end-line-spacing-vertical` увеличивают длину выходящей из конечной фигуры линии. Для соединительных элементов линий эти атрибуты определяют абсолютную длину выходящей из конечной фигуры линии. Для других типов соединительных элементов атрибуты игнорируются.

```
16095 <define name="style-graphic-properties-attlist" combine="interleave">
16096     <optional>
16097         <attribute name="draw:end-line-spacing-horizontal">
16098             <ref name="distance"/>
16099         </attribute>
16100     </optional>
16101     <optional>
16102         <attribute name="draw:end-line-spacing-vertical">
16103             <ref name="distance"/>
16104         </attribute>
16105     </optional>
16106 </define>
```

15.20 Свойства измерений

Свойства, описанные в данном разделе, являются специфическими для графических размерных объектов. Эти свойства доступны графическим размерным объектам, содержащимся во всех видах приложений.

15.20.1 Расстояние до линии

Атрибут `draw:line-distance` определяет расстояние от опорных точек до размерной линии.

```
16107 <define name="style-graphic-properties-attlist" combine="interleave">
16108     <optional>
16109         <attribute name="draw:line-distance">
16110             <ref name="distance"/>
16111         </attribute>
```

```
    </optional>
</define>
```

15.20.2 Выступ направляющих

Направляющие — это две линии, идущие из опорных точек до размерной линии. Атрибут `draw:guide-overhang` определяет длину, на которую прорисовываются направляющие после того, как они пересекли размерную линию.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:guide-overhang">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.20.3 Расстояние до направляющих

Атрибут `draw:guide-distance` определяет расстояние между опорными точками и начальными точками направляющих. Это расстояние не использует атрибуты `draw:start-guide` и `draw:end-guide`, то есть расстояние, определенное в `draw:guide-distance`, равно фактически прорисованному расстоянию, только если `draw:start-guide` и `draw:end-guide` оба равны 0.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:guide-distance">
            <ref name="distance"/>
        </attribute>
    </optional>
</define>
```

15.20.4 Начальная направляющая

Атрибут `draw:start-guide` определяет длину, которая добавляется к расстоянию от первой опорной точки до размерной линии. Направляющая продолжается на эту длину от конца, указывающего в направлении опорных точек.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:start-guide">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.20.5 Конечная направляющая

Атрибут `draw:end-guide` определяет длину, которая добавляется к расстоянию от второй опорной точки до размерной линии. Направляющая продолжается на эту длину от конца, указывающего в направлении опорных точек.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:end-guide">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.20.6 Расположение

Атрибут `draw:placing` определяет, отображается ли размерная линия под или над ребром, определенном опорными точками. Атрибут может принимать значения `below` или `above`.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:placing">
            <choice>
                <value>below</value>
                <value>above</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.20.7 Параллельность

Атрибут `draw:parallel` определяет, как будет отображаться текст измерения — параллельно или перпендикулярно к размерной линии.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:parallel">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.20.8 Выравнивание текста

Атрибуты `draw:measure-align` и `draw:measure-vertical-align` определяют выравнивание текста относительно размерной линии по горизонтали и по вертикали. Если эти атрибуты принимают значение `automatic`, приложение выбирает наилучшую позицию.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:measure-align">
            <choice>
                <value>automatic</value>
                <value>left-outside</value>
                <value>inside</value>
                <value>right-outside</value>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="draw:measure-vertical-align">
            <choice>
                <value>automatic</value>
                <value>above</value>
                <value>below</value>
                <value>center</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.20.9 Единица измерения

Атрибут `draw:unit` определяет единицу измерения, используемую в текстовом представлении размерной фигуры.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:unit">
            <choice>
                <value>automatic</value>
                <value>mm</value>
                <value>cm</value>
                <value>m</value>
                <value>km</value>
                <value>pt</value>
                <value>pc</value>
                <value>inch</value>
                <value>ft</value>
                <value>mi</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.20.10 Отображать единицу измерения

Атрибут `draw:show-unit` включает отображение единицы измерения в текстовом представлении размерной фигуры.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:show-unit">
            <ref name="boolean"/>
        </attribute>
    </optional>
```

```
</define>
```

15.20.11 Десятичные разряды

Атрибут `draw:decimal-places` определяет количество десятичных разрядов, используемых для размерного текста.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:decimal-places">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.21 Свойства примечаний

Следующие атрибуты могут быть использованы в стилях для фигур примечаний. Эти свойства доступны объектам примечаний, содержащимся во всех видах приложений:

- тип;
- тип угла;
- угол;
- отступ;
- направление выхода;
- точка выхода;
- длина линии;
- подгонять длину линии.

15.21.1 Тип

Атрибут `draw:caption-type` определяет геометрию линии примечания. Он может принимать следующие значения:

- `straight-line` — прямая перпендикулярная линия, рисуемая к точке примечания;
- `angled-line` — прямая линия, рисуемая к точке примечания;
- `Angled-connector-line` — прямая перпендикулярная линия, следующая за прямой линией, рисуемая к точке примечания.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:caption-type">
```

```
            <choice>
                <value>straight-line</value>
                <value>angled-line</value>
                <value>angled-connector-line</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.21.2 Тип угла

Атрибут `draw:caption-angle-type` определяет, является ли угол выхода линии фиксированным или изменяемым. Если атрибут принимает значение `free`, приложение может выбирать лучший из возможных углов.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:caption-angle-type">
            <choice>
                <value>fixed</value>
                <value>free</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.21.3 Угол

Атрибут `draw:caption-angle` определяет угол выхода линии примечания. Он рассматривается, только если атрибут `draw:caption-angle-type` имеет значение `fixed`.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:caption-angle">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.21.4 Отступ

Атрибут `draw:caption-gap` определяет расстояние между областью текста примечания и началом линии.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:caption-gap">
            <ref name="distance"/>
        </attribute>
    </optional>
</define>
```

15.21.5 Направление выхода

Атрибут `draw:caption-escape-direction` определяет направление выхода линии примечания. Если он установлен в значение `auto`, приложение может выбрать лучшее из возможных направлений.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:caption-escape-direction">
            <choice>
                <value>horizontal</value>
                <value>vertical</value>
                <value>auto</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.21.6 Точка выхода

Атрибут `draw:caption-escape` определяет точку выхода линии примечания, отсчитываемую от верхнего левого угла области текста. Значение может быть абсолютной длиной или выражено в процентах.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:caption-escape">
            <choice>
                <ref name="length"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.21.7 Длина линии

Атрибут `draw:caption-line-length` определяет длину первой линии примечания (т. е. той, которая начинается от области текста примечания). Атрибут рассматривается только в том случае, если атрибут `draw:caption-fit-line-length` имеет значение `false`.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:caption-line-length">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.21.8 Подгонять длину линии

Если атрибут `draw:caption-fit-line-length` имеет значение `true`, приложение определяет лучшую из возможных длин линии примечания.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:caption-fit-line-length">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.22 Свойства трехмерной геометрии

Свойства трехмерной геометрии, описанные в данном разделе, применимы к графическим трехмерным объектам. Эти свойства доступны графическим трехмерным объектам, содержащимся во всех видах приложений.

15.22.1 Горизонтальные сегменты

Если геометрия трехмерного объекта генерируется во время выполнения, атрибут `dr3d:horizontal-segments` используется для определения количества горизонтальных сегментов, применяемых для генерации геометрии. Обычно приложения поддерживают значения от 2 до 256.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:horizontal-segments">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.22.2 Вертикальные сегменты

Если геометрия трехмерного объекта генерируется во время выполнения, атрибут `dr3d:vertical-segments` используется для определения количества вертикальных сегментов, применяемых для генерации геометрии. Обычно приложения поддерживают значения от 2 до 256.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:vertical-segments">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.22.3 Скругление граней

Если геометрия трехмерного объекта генерируется во время выполнения, атрибут `dr3d:edge-rounding` применяется для определения размера области на гранях геометрии, которые используются для скругления граней.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:edge-rounding">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

15.22.4 Режим скругления граней

Атрибут `dr3d:edge-rounding-mode` определяет, каким образом генерируются скругленные грани. Атрибут может принимать значения `correct` или `attractive`. Если атрибут принимает значение `correct`, используется точный математический метод, если — `attractive`, используется метод, который сохраняет визуальное отображение используемого текста.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:edge-rounding-mode">
            <choice>
                <value>correct</value>
                <value>attractive</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.22.5 Перспектива

Атрибут `dr3d:back-scale` определяет пропорции перспективы для объектов вращения и выдавливания.

Например, при величине перспективы 50 %, задний план объекта выдавливания имеет размер, в два раза меньший, чем передний план.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:back-scale">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

15.22.6 Глубина

Атрибут `dr3d:depth` определяет глубину выдавливания для объектов выдавливания.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:depth">
            <ref name="length"/>
        </attribute>
    </optional>
</define>
```

15.22.7 Невидимая поверхность

Атрибут `dr3d:backface-culling` разрешает или запрещает отображение невидимой поверхности.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:backface-culling">
            <choice>
                <value>enabled</value>
                <value>disabled</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.22.8 Конечный угол

Атрибут `dr3d:end-angle` определяет угол поворота для трехмерных объектов вращения. Если он имеет значение по умолчанию (360°), то объект вращения замыкается и поворачивается полностью. При меньших значениях возможно определение открытого объекта вращения (сегмента). Тогда видимые стороны замыкаются и получаются вычислением атрибутов `dr3d:back-scale` и `dr3d:edge-rounding`. При больших значениях возможно создание объектов вращения с более чем одним поворотом. Это приведет к видимому эффекту, только если используется, например, `dr3d:back-scale`.

Например, при конечном угле 270° объект вращения будет открыт на 90°.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:end-angle">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.22.9 Закрытая фронтальная поверхность

Свойство `dr3d:close-front` определяет, генерируется ли фронтальная поверхность. Например, если эллипс подвергается выдавливанию, и данный атрибут установлен, эллипс будет иметь открытую фронтальную поверхность. Атрибут может использоваться для объектов выдавливания и вращения.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:close-front">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.22.10 Закрытая задняя поверхность

Свойство `dr3d:close-back` определяет, генерируется ли задняя поверхность. Например, если эллипс подвергается выдавливанию, и данный атрибут установлен, эллипс будет иметь открытую заднюю поверхность. Атрибут может использоваться для объектов выдавливания и вращения.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:close-back">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.23 Свойства трехмерного освещения

Свойства трехмерного освещения, описанные в данном разделе, применимы к графическим трехмерным объектам. Эти свойства доступны графическим трехмерным объектам, содержащимся во всех видах приложений.

15.23.1 Режим

Атрибут `dr3d:lighting-mode` определяет алгоритм освещения, используемый для визуализации соответствующего трехмерного объекта.

Атрибут может принимать значения `standard` или `double-sided`. Если принимается значение `double-sided`, обратная сторона объекта также освещается.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
```

```
16361         <attribute name="dr3d:lighting-mode">
16362             <choice>
16363                 <value>standard</value>
16364                 <value>double-sided</value>
16365             </choice>
16366         </attribute>
16367     </optional>
16368 </define>
```

15.23.2 Тип нормали

Атрибут `dr3d:normals-kind` определяет, как настраивается нормаль для генерации освещения. Он может принимать следующие значения:

- `object` — не вырабатывает стандартных нормалей и оставляет нормали, специфические для объекта без изменения;
- `flat` — привязывает одну нормаль на плоскость;
- `sphere` — привязывает нормали для поведения объекта как сферы.

```
16369 <define name="style-graphic-properties-attlist" combine="interleave">
16370     <optional>
16371         <attribute name="dr3d:normals-kind">
16372             <choice>
16373                 <value>object</value>
16374                 <value>flat</value>
16375                 <value>sphere</value>
16376             </choice>
16377         </attribute>
16378     </optional>
16379 </define>
```

15.23.3 Направление нормалей

Атрибут `dr3d:normals-direction` используется для инвертирования настроек сгенерированных нормалей.

```
16380 <define name="style-graphic-properties-attlist" combine="interleave">
16381     <optional>
16382         <attribute name="dr3d:normals-direction">
16383             <choice>
16384                 <value>normal</value>
16385                 <value>inverse</value>
16386             </choice>
16387         </attribute>
16388     </optional>
16389 </define>
```

15.24 Свойства трехмерного текстурирования

Свойства трехмерных текстур, описанные в данном разделе, применимы к графическим трехмерным объектам. Эти свойства доступны графическим трехмерным объектам, содержащимся во всех видах приложений.

15.24.1 Режим генерации

Атрибуты `dr3d:texture-generation-mode-x` и `dr3d:texture-generation-mode-y` определяют как генерируются координаты текстуры. Они могут принимать следующие значения:

- `object` — это значение определяет, что используется стандартный метод проецирования;
- `parallel` — это значение определяет плоскую параллельную проекцию в определенной степени свободы (X или Y);
- `sphere` — это значение привязывает проекцию к окружности в X и/или Y направлении.

```
16390 <define name="style-graphic-properties-attlist" combine="interleave">
16391     <optional>
16392         <attribute name="dr3d:texture-generation-mode-x">
16393             <choice>
16394                 <value>object</value>
16395                 <value>parallel</value>
16396                 <value>sphere</value>
16397             </choice>
16398         </attribute>
16399     </optional>
16400     <optional>
16401         <attribute name="dr3d:texture-generation-mode-y">
16402             <choice>
16403                 <value>object</value>
16404                 <value>parallel</value>
16405                 <value>sphere</value>
16406             </choice>
16407         </attribute>
16408     </optional>
16409 </define>
```

15.24.2 Тип

Атрибут `dr3d:texture-kind` используется для определения того, что изменяет текстура: яркость, интенсивность или цвет фигуры.

```
16410 <define name="style-graphic-properties-attlist" combine="interleave">
16411     <optional>
16412         <attribute name="dr3d:texture-kind">
16413             <choice>
16414                 <value>luminance</value>
16415                 <value>intesity</value>
16416                 <value>color</value>
16417             </choice>
16418         </attribute>
16419     </optional>
16420 </define>
```

15.24.3 Фильтр

Атрибут `dr3d:texture-filter` используется для разрешения или запрещения фильтрации текстуры.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:texture-filter">
            <choice>
                <value>enabled</value>
                <value>disabled</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.24.4 Режим

Атрибут `dr3d:texture-mode` используется для определения того, каким образом модулируется текстура.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:texture-mode">
            <choice>
                <value>replace</value>
                <value>modulate</value>
                <value>blend</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.25 Свойства трехмерных материалов

Свойства трехмерных материалов, описанные в данном разделе, применимы к графическим трехмерным объектам. Эти свойства доступны графическим трехмерным объектам, содержащимся во всех видах приложений.

15.25.1 Цвета

Атрибуты `dr3d:ambient-color`, `dr3d:emissive-color`, `dr3d:specular-color` и `dr3d:diffuse-color` определяют четыре цвета, задающие материал.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:ambient-color">
            <ref name="color"/>
        </attribute>
    </optional>
```

```
    <optional>
        <attribute name="dr3d:emissive-color">
            <ref name="color"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="dr3d:specular-color">
            <ref name="color"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="dr3d:diffuse-color">
            <ref name="color"/>
        </attribute>
    </optional>
</define>
```

15.25.2 Глянцевость

Атрибут `dr3d:shininess` определяет глянец материала.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:shininess">
            <ref name="percent"/>
        </attribute>
    </optional>
</define>
```

15.26 Свойства трехмерных теней

Свойства трехмерных теней, описанные в данном разделе, применимы к графическим трехмерным объектам. Эти свойства доступны графическим трехмерным объектам, содержащимся во всех видах приложений.

15.26.1 Тень

Атрибут `dr3d:shadow` разрешает или запрещает трехмерную тень для трехмерных объектов.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="dr3d:shadow">
            <choice>
                <value>visible</value>
                <value>hidden</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.27 Свойства форматирования фреймов

Свойства, описанные в данном разделе, применяются для графических фреймов (см. раздел 9.3). Они могут быть использованы в графических стилях (см. раздел 14.13.1) и содержатся в элементе `<style:graphic-properties>`.

15.27.1 Ширина фрейма

Есть три типа определения ширины фрейма: фиксированная ширина, минимальная ширина и относительная ширина. Фиксированная ширина определяется атрибутом `svg:width`, минимальная ширина — атрибутом `fo:min-width`, и относительная ширина — атрибутом `style:rel-width`. Значение этих атрибутов такое, как описано в разделе 9.3, за исключением того, что атрибуты определяют ширину по умолчанию только для вновь созданного фрейма. Атрибут `style:rel-width` рассматривается только для графических стилей, примененных к текстовым блокам.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <ref name="common-draw-rel-size-attlist"/>
    <optional>
        <attribute name="fo:min-width">
            <choice>
                <ref name="length"/>
                <ref name="percent"/>
            </choice>
        </attribute>
    </optional>
</define>
```

15.27.2 Высота фрейма

Есть три типа определения высоты фрейма: фиксированная высота, минимальная высота и относительная высота. Фиксированная высота определяется атрибутом `svg:height`, минимальная высота — атрибутом `fo:min-height`, и относительная высота — атрибутом `style:rel-height`. Значение этих атрибутов такое, как описано в разделе 9.3, за исключением того, что атрибуты определяют высоту по умолчанию только для вновь созданного фрейма. Атрибут `style:rel-height` рассматривается только для графических стилей, примененных к текстовым блокам (см. также раздел 15.27.1).

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:min-height">
```

```
16495            <choice>
16496                <ref name="length"/>
16497                <ref name="percent"/>
16498            </choice>
16499        </attribute>
16500    </optional>
16501 </define>
```

15.27.3 Максимальные ширина и высота

Текстовые блоки могут увеличиваться в размере при добавлении содержимого. Атрибуты `fo:max-width` и `fo:max-height` определяют максимальные ширину и высоту фрейма. Когда достигается максимальное значение, фрейм прекращает увеличиваться в размере. Атрибут может принимать значения в единицах длины или процентах. Если текстовый блок привязан к ячейке таблицы, значение в процентах относится к окружающему табличному блоку. Если данный текстовый блок привязан к другому текстовому блоку, то значение в процентах относится к окружающему текстовому блоку. В других случаях значение в процентах относится к высоте страницы или окна.

```
16502 <define name="style-graphic-properties-attlist" combine="interleave">
16503     <optional>
16504         <attribute name="fo:max-height">
16505             <choice>
16506                 <ref name="length"/>
16507                 <ref name="percent"/>
16508             </choice>
16509         </attribute>
16510     </optional>
16511     <optional>
16512         <attribute name="fo:max-width">
16513             <choice>
16514                 <ref name="length"/>
16515                 <ref name="percent"/>
16516             </choice>
16517         </attribute>
16518     </optional>
16519 </define>
```

15.27.4 Левое и правое поля

Свойства `fo:margin-left` и `fo:margin-right` определяют левое и правое поля, устанавливаемые вокруг фрейма. Детальная информация об этих атрибутах приведена в разделе 15.5.17. Значения в процентах не поддерживаются.

```
16520 <define name="style-graphic-properties-attlist" combine="interleave">
16521     <ref name="common-horizontal-margin-attlist"/>
16522 </define>
```

15.27.5 Верхнее и нижнее поля

Свойства `fo:margin-top` и `fo:margin-bottom` определяют верхнее и нижнее поля, устанавливаемые вокруг фрейма. Детальная информация об этих атрибутах приведена в разделе 15.5.20. Значения в процентах не поддерживаются.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <ref name="common-vertical-margin-attlist"/>
</define>
```

15.27.6 Поля

Свойство `fo:margin` определяет поля для всех четырех сторон фрейма. Полное разъяснение данных свойств приведено в разделе 15.5.21.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <ref name="common-margin-attlist"/>
</define>
```

15.27.7 Печатать содержимое

Свойство `style:print-content` определяет, выводится ли содержимое фрейма на печать.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:print-content">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.27.8 Защита

Свойство `style:protect` определяет, являются ли защищенными содержимое, размер или положение фрейма. Значение этого свойства может принимать значения `none` или разделенный пробелами список значений `content`, `position` или `size`.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:protect">
            <choice>
                <value>none</value>
                <list>
                    <oneOrMore>
                        <choice>
                            <value>content</value>
                            <value>position</value>
```

```
                            <value>size</value>
                        </choice>
                    </oneOrMore>
                </list>
            </choice>
        </attribute>
    </optional>
</define>
```

15.27.9 Горизонтальное положение

В текстовом документе свойство `style:horizontal-pos` определяет горизонтальное выравнивание фрейма относительно определенной области.

Свойство может принимать следующие значения: `from-left`, `left`, `center`, `right`, `from-inside`, `inside` или `outside`. Область, которой соответствует положение, определяется свойством `style:horizontal-rel`. Значения `from-inside`, `inside` и `outside` соответствуют значениям `from-left`, `left` и `right` на странице, которая имеет четный номер, и обратные значения на страницах, имеющих нечетный номер.

Если свойство имеет значение `from-left` или `from-inside`, горизонтальное положение фрейма определяется атрибутом `svg:x`, связанным с элементом фрейма. Иначе атрибут `svg:x` в текстовых документах игнорируется.

Также возможно использование атрибута `svg:x` в графическом стиле. Если такой случай имеет место, атрибут определяет положение по умолчанию для всех новых фреймов, созданных с использованием этого стиля.

Некоторые значения могут быть использованы только в совокупности с определенными типами привязок и отношений фреймов.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:horizontal-pos">
            <choice>
                <value>left</value>
                <value>center</value>
                <value>right</value>
                <value>from-left</value>
                <value>inside</value>
                <value>outside</value>
                <value>from-inside</value>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="svg:x">
            <ref name="coordinate"/>
        </attribute>
    </optional>
```

16573 `</define>`

В следующей таблице представлены возможные значения атрибутов `style:horizontal-pos` и `style:horizontal-rel`. Возможные значения этих атрибутов выравнивания перечислены в первом столбце слева, а соответствия «значение атрибута выравнивания/тип привязки» обозначены знаком X.

Значение `style:horizontal-pos`	***Значение*** `text:anchor-type`				
	page	frame	paragraph	char	as-char
any	X	X	X	X	

Значение `style:horizontal-rel`	***Значение*** `text:anchor-type`				
	page	frame	paragraph	char	as-char
page	X		X	X	
page-content	X		X	X	
page-start-margin	X		X	X	
page-end-margin	X		X	X	
frame		X			
frame-content		X			
frame-start-margin		X			
frame-end-margin		X			
paragraph			X	X	
paragraph-content			X	X	
paragraph-start-margin			X	X	
paragraph-end-margin			X	X	
char				X	

15.27.10 Горизонтальное отношение

Свойство `style:horizontal-rel` определяет область, к которой относится горизонтальное положение фрейма. Для получения информации о свойстве `style:horizontal-pos` см. раздел 15.27.9.

Свойство может принимать следующие значения: `page`, `page-content`, `page-start-margin`, `page-end-margin`, `frame`, `frame-content`, `frame-start-margin`, `frame-end-margin`, `paragraph`, `paragraph-content`, `paragraph-start-margin`, `paragraph-end-margin` или `char`.

Некоторые значения могут использоваться только с определенными типами привязки фрейма.

Значение `start-margin` определяет левое поле, за исключением случая, когда горизонтальное положение имеет значение `from-inside`, `inside` или `outside` и фрейм привязан к странице с четным номером. В этом случае оно определяет правое поле. Значение `end-margin` определяет поле, противоположное значению `start-margin`.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:horizontal-rel">
            <choice>
                    <value>page</value>
                <value>page-content</value>
                <value>page-start-margin</value>
                <value>page-end-margin</value>
                <value>frame</value>
                <value>frame-content</value>
                <value>frame-start-margin</value>
                <value>frame-end-margin</value>
                <value>paragraph</value>
                <value>paragraph-content</value>
                <value>paragraph-start-margin</value>
                <value>paragraph-end-margin</value>
                <value>char</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.27.11 Вертикальное положение

В текстовом документе свойство `style:vertical-pos` определяет вертикальное выравнивание фрейма относительно определенной области.

Свойство может принимать следующие значения: `from-top`, `top`, `middle`, `below` или `bottom`. Область, которой соответствует положение, определяется свойством `style:vertical-rel`. Значения `top`, `middle` и `bottom` определяют, что заданные края фрейма и соответствующая область выравниваются. Значение `below` определяет, что верхний край фрейма располагается над соответствующей областью.

Если свойство имеет значение `from-top`, вертикальное положение фрейма определяется атрибутом `svg:y`, связанным с элементом фрейма. Иначе атрибут `svg:y` в текстовых документах игнорируется.

Также возможно использование атрибута `svg:y` в графическом стиле. Если такой случай имеет место, атрибут определяет положение по умолчанию для всех новых фреймов, созданных с использованием этого стиля.

Некоторые значения могут быть использованы только в совокупности с определенными фреймом типами привязки и отношения.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <ref name="common-vertical-pos-attlist"/>
</define>

<define name="common-vertical-pos-attlist">
    <optional>
        <attribute name="style:vertical-pos">
            <choice>
                <value>top</value>
                <value>middle</value>
                <value>bottom</value>
                <value>from-top</value>
                <value>below</value>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="svg:y">
            <ref name="coordinate"/>
        </attribute>
    </optional>
</define>
```

В следующей таблице представлены возможные значения атрибутов `style:vertical-pos` и `style:vertical-rel`. Возможные значения этих атрибутов выравнивания перечислены в первом столбце слева, а соответствия «значение атрибута выравнивания/тип привязки» обозначены знаком X.

Значение `style:vertical-pos`	***Значение*** `text:anchor-type`				
	page	frame	paragraph	char	as-char
any	X	X	X	X	X

Значение `style:vertical-rel`	***Значение*** `text:anchor-type`				
	page	frame	paragraph	char	as-char
page	X				
page-content	X				
frame		X			
frame-content		X			
paragraph			X	X	
paragraph-content			X	X	

Значение `style:vertical-rel`	***Значение*** `text:anchor-type`				
	page	frame	paragraph	char	as-char
char				X	X
line					X
baseline					X
text					X

15.27.12 Вертикальное отношение

Свойство `style:vertical-rel` определяет область, к которой относится вертикальное положение фрейма. Для получения информации о свойстве `style:vertical-pos` см. раздел 15.27.11.

Свойство может принимать следующие значения: `page`, `page-content`, `frame`, `frame-content`, `paragraph`, `paragraph-content`, `line`, `baseline`, `text` или `char`.

Некоторые значения могут использоваться только с определенными типами привязки фрейма.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <ref name="common-vertical-rel-attlist"/>
</define>

<define name="common-vertical-rel-attlist">
    <optional>
        <attribute name="style:vertical-rel">
            <choice>
                <value>page</value>
                <value>page-content</value>
                <value>frame</value>
                <value>frame-content</value>
                <value>paragraph</value>
                <value>paragraph-content</value>
                <value>char</value>
                <value>line</value>
                <value>baseline</value>
                <value>text</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.27.13 Привязка фрейма

Атрибуты `text:anchor-type` и `text:anchor-page-number` определяют привязку по умолчанию для новых фреймов и графических объектов. Для получения детальной информации см. раздел 9.2.16.

```
<define name="style-graphic-properties-attlist" combine="interleave">
```

```
    <ref name="common-text-anchor-attlist"/>
</define>
```

15.27.14 Обрамление

Атрибуты обрамления `fo:border`, `fo:border-top`, `fo:border-bottom`, `fo:border-left` и `fo:border-right` определяют свойства обрамления фрейма. Детальная информация об этих атрибутах приведена в разделе 15.5.25.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <ref name="common-border-attlist"/>
</define>
```

15.27.15 Толщина линии обрамления

Если фрейм имеет обрамление, атрибуты `style:border-line-width`, `style:border-line-width-top`, `style:border-line-width-bottom`, `style:border-line-width-left` и `style:border-line-width-right` определяют свойства линий обрамления фрейма. Детальная информация об этих атрибутах приведена в разделе 15.5.26.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <ref name="common-border-line-width-attlist"/>
</define>
```

15.27.16 Отступы

Атрибуты отступов `fo:padding`, `fo:padding-top`, `fo:padding-bottom`, `fo:padding-left` и `fo:padding-right` определяют свойства отступов фрейма. Детальная информация об этих атрибутах приведена в разделе 15.5.27.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <ref name="common-padding-attlist"/>
</define>
```

15.27.17 Тень

Атрибут `style:shadow` определяет тень фрейма. Детальная информация об этом атрибуте приведена в разделе 15.5.28.

```
<define name="style-graphic-properties-attlist" combine="interleave">
```

```
    <ref name="common-shadow-attlist"/>
</define>
```

15.27.18 Фон

Атрибут фона `fo:background-color` и элемент фона `<style:background-image>` определяют свойства фона фрейма. Детальная информация об этих атрибуте и элементе приведена в разделах 15.5.23 и 15.5.24.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <ref name="common-background-color-attlist"/>
</define>
<define name="style-graphic-properties-elements" combine="interleave">
    <ref name="style-background-image"/>
</define>
```

15.27.19 Колонки

Элемент `<style:columns>` определяет, содержит ли текстовый блок колонки. Детальная информация об этом элементе приведена в разделе 15.7.3.

```
<define name="style-graphic-properties-elements" combine="interleave">
    <ref name="style-columns"/>
</define>
```

15.27.20 Возможность редактирования

В текстовых документах текстовый блок может быть редактируемым, даже если документ, в котором он содержится, является документом только для чтения. Свойство `style:editable` определяет, может ли редактироваться текстовый блок.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:editable">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.27.21 Обтекание

В текстовых документах свойство `style:wrap` определяет, каким образом текст располагается вокруг фрейма или графического объекта. Например, текст

может идти вокруг левой стороны фрейма, вокруг правой стороны фрейма или сквозь фрейм. Возможные значения:

- none — текст не обтекает графическую фигуру;
- left — текст может обтекать графическую фигуру с левой стороны;
- right — текст может обтекать графическую фигуру с правой стороны;
- parallel — текст может обтекать графическую фигуру с двух сторон;
- dynamic — текст может обтекать графическую фигуру с двух сторон, если обеспечивается достаточное для этого пространство;
- biggest — текст может обтекать ту сторону, где расстояние до левой или правой границы страницы или колонки наибольшее;
- run-through — текст идет сквозь графический объект.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:wrap">
            <choice>
                <value>none</value>
                <value>left</value>
                <value>right</value>
                <value>parallel</value>
                <value>dynamic</value>
                <value>run-through</value>
                <value>biggest</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.27.22 Порог динамического обтекания

Атрибут style:wrap-dynamic-threshold рассматривается только в том случае, если атрибут style:wrap имеет значение dynamic. Он определяет минимальное расстояние между границей страницы или колонки и объектом, для которого было разрешено обтекание.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:wrap-dynamic-treshold">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

15.27.23 Обтекание только абзацами

Если позицией привязки фрейма или графического объекта является абзац или символ, и режим обтекания, установленный свойством style:wrap, имеет

значение `left`, `right`, `parallel` или `dynamic`, то с использованием атрибута `style:number-wrapped-paragraphs` можно определить количество абзацев, которые будут обтекать фрейм.

Свойство распознается только фреймами или стилями, свойство `style:wrap` которых имеет значение `left`, `right`, `parallel` или `dynamic`.

Если установлено значение `no-limit`, количество абзацев, которые могут обтекать фрейм, не ограничивается.

```
16692 <define name="style-graphic-properties-attlist" combine="interleave">
16693     <optional>
16694         <attribute name="style:number-wrapped-paragraphs">
16695             <choice>
16696                 <value>no-limit</value>
16697                 <ref name="positiveInteger"/>
16698             </choice>
16699         </attribute>
16700     </optional>
16701 </define>
```

15.27.24 Контурное обтекание

В текстовых документах атрибут `style:wrap-contour` для некоторых типов фреймов определяет, что текст должен обтекать фигуру объекта в фрейме вместо обтекания самого фрейма. Это называется контурным обтеканием.

```
16702 <define name="style-graphic-properties-attlist" combine="interleave">
16703     <optional>
16704         <attribute name="style:wrap-contour">
16705             <ref name="boolean"/>
16706         </attribute>
16707     </optional>
16708 </define>
```

15.27.25 Режим контурного обтекания

Атрибут `style:wrap-contour-mode` используется для дальнейшего определения, каким образом текст должен обтекать контур.

Значение атрибута распространяется только на фреймы, графические фигуры или стили, которые уже имеют примененные атрибуты `style:wrap` и `style:wrap-contour`.

Атрибут может принимать значения `outside` или `full`. Если атрибут имеет значение `outside`, текст обтекает основную область слева и справа фигуры. Если атрибут имеет значение `full`, текст обтекает фигуру и заполняет все возможное пространство, внедряясь в фигуру.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:wrap-contour-mode">
            <choice>
                <value>full</value>
                <value>outside</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.27.26 Сквозное обтекание

Если атрибут `style:wrap` имеет значение `run-through`, далее может быть определено, каким образом будет отображаться текст: на заднем плане или на переднем. Обычно для задания прозрачности объекта для текста используется атрибут `style:run-through`.

Атрибут может принимать значение `foreground` или `background`. Если значение равно `foreground`, содержимое фрейма отображается перед текстом. Если значение равно `background`, содержимое фрейма отображается за текстом.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:run-through">
            <choice>
                <value>foreground</value>
                <value>background</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.27.27 Следование за текстом

Атрибут `style:flow-with-text` определяет поведение графического изображения, которое расположено на некотором расстоянии ниже привязки и не помещается на странице привязки. Если свойство имеет значение `true`, такой графический объект следует за текстом, то есть отражается на следующей странице. Если атрибут имеет значение `false`, такой графический объект отображается за пределами текстовой области страницы.

Пример —

Изображение расположено на 10 см ниже его привязки. Оно сопровождает текст только 8 см, пока не достигает разрыва страницы. При `style:flow-with-text='false'` изображение будет расположено на 2 см ниже текстовой области

(где-то в нижнем колонтитуле); при style:flow-with-text='true' оно будет расположено через 2 см в полосе текста на следующей странице.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:flow-with-text">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.27.28 Поведение при переполнении

Для текстовых блоков, содержащихся в текстовых документах, свойство style:overflow-behavior определяет поведение текстового блока в том случае, когда содержащийся текст не помещается в текстовом блоке. Если свойство имеет значение clip, не помещающийся в текстовом блоке текст не отображается. Если свойство имеет значение auto-create-new-frame, на следующей странице будет создан новый фрейм, имеющий такое же положение и размеры, как и исходный фрейм.

Если свойство style:overflow-behavior имеет значение auto-create-new-frame и текстовый блок имеет определенную минимальную ширину или высоту, то текстовый блок будет увеличиваться перед созданием нового фрейма до тех пор, пока не достигнет границы страницы.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:overflow-behavior">
            <choice>
                <value>clip</value>
                <value>auto-create-new-frame</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.27.29 Зеркальное отображение

Свойство style:mirror определяет, отображается ли зеркально изображение перед выводом на экран. Отображение может быть вертикальным или горизонтальным. Горизонтальное отображение может быть ограничено для изображений, расположенных и на четной и на нечетной страницах.

Атрибут может принимать значения none, vertical, horizontal, horizontal-on-odd, или horizontal-on-even. Значение vertical и

различные горизонтальные значения могут быть определены совместно с разделением пробелами.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="style:mirror">
            <choice>
                <value>none</value>
                <value>vertical</value>
                <ref name="horizontal-mirror"/>
                <list>
                    <value>vertical</value>
                    <ref name="horizontal-mirror"/>
                </list>
                <list>
                    <ref name="horizontal-mirror"/>
                    <value>vertical</value>
                </list>
            </choice>
        </attribute>
    </optional>
</define>

<define name="horizontal-mirror">
    <choice>
        <value>horizontal</value>
        <value>horizontal-on-odd</value>
        <value>horizontal-on-even</value>
    </choice>
</define>
```

15.27.30 Обрезка

Свойство `fo:clip` определяет, отображается ли:

- прямоугольная секция изображения;
- изображение целиком.

Детальная информация приведена в § 7.20.1 [XSL].

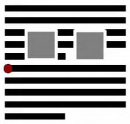
```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="fo:clip">
            <!-- The attribute value must match the one XSL's clip -->
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

15.27.31 Влияние обтекания на расположение

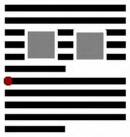
Этот атрибут детализирует влияние режима обтекания (см. атрибут `style:wrap`) на расположение фрейма. Он используется как предписание для алгоритма разметки, позволяющее алгоритму разметки принять решение о расположении фреймов в некоторых случаях, когда могут быть использованы

несколько корректных положений. Все три варианта описывают различные корректные интерпретации разметки, уже примененные в формате. Новое предписание позволяет устранить противоречия между этими ситуациями.

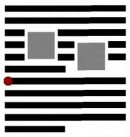
```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:wrap-influence-on-position"
                   a:defaultValue="iterative">
            <choice>
                <value>iterative</value>
                <value>once-concurrent</value>
                <value>once-successive</value>
            </choice>
        </attribute>
    </optional>
</define>
```



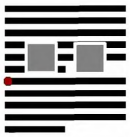
Ситуация, в которой этот атрибут имеет значение, — это случай, когда привязка, режим положения и обтекания фрейма таковы, что влияют друг на друга. Например, рассмотрим абзац текста с двумя изображениями, расположенными несколько выше привязки. Без обтекания изображения накроют текст и могут быть просто размещены на заданном расстоянии от привязки.



Если разрешено обтекание, скрытый позади изображений текст теперь должен проследовать вокруг них, приводя к тому, что первый абзац займет больше места, чем ранее. Это приведет к перемещению привязки далее вниз. Если это произойдет один раз одновременно для всех объектов, то будет достигнут конечный результат. Это соответствует варианту `once-concurrently`.



Если все происходит также, как описано выше, но изображения обрабатываются последовательно, они достигнут позиций, показанных справа. Это соответствует варианту `once-successive`.



Если размещение изображений осуществляется итеративно, данные действия могут выполняться много раз до достижения расположения (в некоторых случаях ценой неэффективной реализации), полностью соответствующего заданным свойствам разметки, т. е. положения, определенного заданным смещением от привязки.

15.28 Свойства форматирования плавающего фрейма

Атрибуты, описанные в этом разделе, могут применяться с графическими стилями, которые связаны с плавающими фреймами.

15.28.1 Отображение полосы прокрутки

Атрибут `draw:display-scrollbar` определяет, будет ли отображаться вертикальная и горизонтальная полосы прокрутки. Этот атрибут может относиться только к автоматическим стилям.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:frame-display-scrollbar">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.28.2 Отображение обрамления

Атрибут `draw:display-border` определяет, будет ли отображаться обрамление на плавающем фрейме. Этот атрибут может относится только к автоматическим стилям.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:frame-display-border">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.28.3 Поля

Атрибуты `draw:margin-horizontal` и `draw:margin-vertical` определяют горизонтальные и вертикальные поля между обрамлением и содержимым плавающего фрейма. Если атрибуты не заданы, используются поля по умолчанию. Эти атрибуты могут относиться только к автоматическим стилям. Значение этих атрибутов представляется длиной в пикселях.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:frame-margin-horizontal">
            <ref name="nonNegativePixelLength"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="draw:frame-margin-vertical">
```

```
            <ref name="nonNegativePixelLength"/>
        </attribute>
    </optional>
</define>

<define name="nonNegativePixelLength">
    <data type="string">
        <param name="pattern">([0-9]+(\.[0-9]*)?|\.[0-9]+)(px)</param>
    </data>
</define>
```

15.28.4 Свойства форматирования объекта

Атрибуты, описанные в данном разделе, могут применяться с графическими стилями, используемыми объектами.

15.28.5 Видимая область

Видимая область объекта — это прямоугольная область объекта, которая видна в данный момент. Атрибуты `draw:visible-area-left`, `draw:visible-area-top`, `draw:visible-area-width` и `draw:visible-area-height` определяют видимую область по умолчанию, которую может использовать объект.

Если видим весь объект, значение атрибутов `draw:visible-area-left` и `draw:visible-area-top` равно нулю, атрибуты `draw:visible-area-width` и `draw:visible-area-height` определяют размер объекта. Эти атрибуты могут относится только к автоматическим стилям.

Не все объекты поддерживают эти атрибуты. Некоторые объекты могут хранить и загружать их собственную видимую область.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:visible-area-left">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="draw:visible-area-top">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="draw:visible-area-width">
            <ref name="positiveLength"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="draw:visible-area-height">
            <ref name="positiveLength"/>
        </attribute>
    </optional>
```

```
</define>
```

15.28.6 Вид рисунка

Атрибут `draw:ole-draw-aspect` определяет вид рисунка, который используется при отображении внедряемых OLE-объектов (см. [OLE]). Вид рисунка определяет, отображается ли объект как обычный поддокумент или объект отображается, например, только как значок. В [OLE] API значение вида рисунка — целое число без знака, которое приложение передает объекту, когда он запрашивает отображение.

Атрибут `draw:ole-draw-aspect` принимает значение в виде неотрицательного целого числа, он имеет значение только для внедренного объекта [OLE] API. В данном случае это значение задает значение по умолчанию для методов, которые запрашивают параметр вида рисунка. Интерпретация этого целочисленного значения остается на усмотрение OLE-объектов и не является частью данной спецификации.

```
<define name="style-graphic-properties-attlist" combine="interleave">
    <optional>
        <attribute name="draw:ole-draw-aspect">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.29 Свойства форматирования диаграмм

Свойства, описанные в данном разделе, могут применяться ко всем диаграммам. Они используются в стилях диаграмм (см. раздел 14.16) и содержатся в элементе `<style:chart-properties>`.

```
<define name="style-chart-properties">
    <element name="style:chart-properties">
        <ref name="style-chart-properties-content"/>
    </element>
</define>

<define name="style-chart-properties-content">
    <ref name="style-properties-content"/>
</define>

<define name="style-chart-properties-content-strict">
    <ref name="style-chart-properties-attlist"/>
    <ref name="style-chart-properties-elements"/>
</define>

<define name="style-chart-properties-elements">
    <empty/>
```

```
</define>
```

15.29.1 Масштабирование текста

Свойство chart:scale-text используется для указания на то, что все текстовые объекты на диаграмме должны изменять масштаб в случае изменения размера диаграммы. Для включения масштабирования значение данного свойства должно быть установлено true.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:scale-text" a:defaultValue="true">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.30 Свойства подтипов диаграмм

Свойства, описанные в этом разделе, могут использоваться для настройки основного стиля диаграмм в элементе <chart:chart>. Они могут использоваться в стилях диаграмм (см. раздел 14.16), содержатся они в элементе <style:chart-properties>.

15.30.1 Трехмерные диаграммы

Свойство chart:three-dimensional определяет, отображается ли диаграмма как трехмерная сцена.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:three-dimensional">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.30.2 Глубина диаграммы

Свойство chart:deep совместимо только со свойством chart:three-dimensional. Оно определяет отображение рядов данных друг за другом, а не рядом.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:deep">
            <ref name="boolean"/>
```

```
16890         </attribute>
16891     </optional>
16892 </define>
```

15.30.3 Символ диаграммы

Для некоторых типов диаграмм разделы данных могут быть обозначены символами. Атрибут `chart:symbol-type` определяет, требуется ли использование символа, а также предопределяет тип символа, изображение или указывает, что приложение само свободно в выборе типа из набора предопределенных типов символа, например, выбрать по символу для каждого ряда в случайном порядке.

```
16893 <define name="style-chart-properties-attlist" combine="interleave">
16894     <choice>
16895         <attribute name="chart:symbol-type">
16896             <value>none</value>
16897         </attribute>
16898         <attribute name="chart:symbol-type">
16899             <value>automatic</value>
16900         </attribute>
16901         <group>
16902             <attribute name="chart:symbol-type">
16903                 <value>named-symbol</value>
16904             </attribute>
16905             <attribute name="chart:symbol-name">
16906                 <choice>
16907                     <value>square</value>
16908                     <value>diamond</value>
16909                     <value>arrow-down</value>
16910                     <value>arrow-up</value>
16911                     <value>arrow-right</value>
16912                     <value>arrow-left</value>
16913                     <value>bow-tie</value>
16914                     <value>hourglass</value>
16915                     <value>circle</value>
16916                     <value>star</value>
16917                     <value>x</value>
16918                     <value>plus</value>
16919                     <value>asterisk</value>
16920                     <value>horizontal-bar</value>
16921                     <value>vertical-bar</value>
16922                 </choice>
16923             </attribute>
16924         </group>
16925         <group>
16926             <attribute name="chart:symbol-type">
16927                 <value>image</value>
16928             </attribute>
16929             <element name="chart:symbol-image">
16930                 <attribute name="xlink:href">
16931                     <ref name="anyURI"/>
16932                 </attribute>
16933             </element>
16934         </group>
16935         <empty/>
16936     </choice>
```

```
</define>
```

15.30.4 Размер символа диаграммы

Высота и ширина каждого символа может быть установлена с использованием атрибутов `chart:symbol-width` и `chart:symbol-length`.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:symbol-width">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="chart:symbol-height">
            <ref name="nonNegativeLength"/>
        </attribute>
    </optional>
</define>
```

15.30.5 Свойства гистограммы

Свойства `chart:vertical` и `chart:connect-bars` используются только гистограммами. Если значение `chart:connect-bars` установлено `true`, пункты данных (верхушки столбиков) дополнительно соединяются линиями.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:vertical" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:connect-bars" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

Свойства гистограмм `chart:gap-width` и `chart:overlap` могут использоваться для определения относительного размера и дистанции между столбцами. Атрибут `chart:gap-width` содержит относительную ширину промежутка между столбцами смежных категорий. Атрибут `chart:overlap` определяет, сколько столбцов внутри одной категории накладываются. Оба атрибута представляются в виде целочисленных процентных значений.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:gap-width">
            <ref name="integer"/>
```

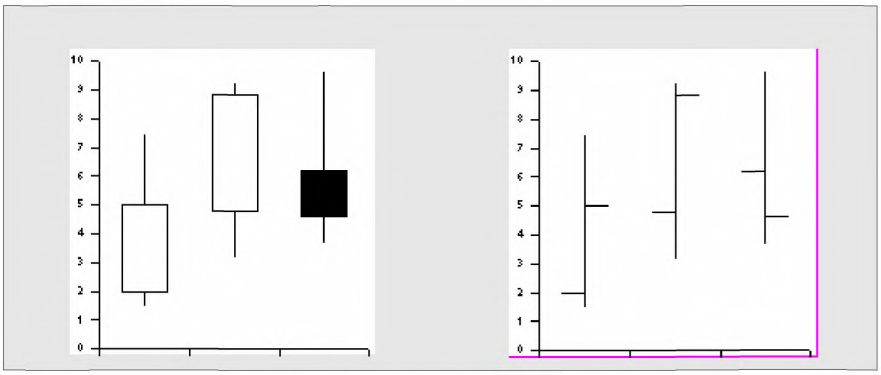
```
        </attribute>
    </optional>
    <optional>
        <attribute name="chart:overlap">
            <ref name="integer"/>
        </attribute>
    </optional>
</define>
```

15.30.6 Свойства биржевых диаграмм

Данные атрибуты применяются только для биржевых диаграмм.

Биржевые диаграммы отображают промежуток между максимальным и минимальным значениями в виде прямой линии. Курс открытия и закрытия может быть отображен либо с помощью горизонтальных линий слева и справа соответственно, либо закрашенными прямоугольниками, цвет которых зависит от того, больше ли курс открытия курса закрытия. Атрибут `chart:japanese-candle-stick` определяет один из двух видов отображения.

Пример — Биржевая диаграмма в виде японских свечей (слева), по умолчанию (справа):




```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:japanese-candle-stick"
                   a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.30.7 Свойства линейной диаграммы

Для линейного вида диаграммы атрибут `chart:interpolation` может принимать одно из следующих значений:

- none – прямая линия, не использовать сглаживание сплайнами;
- cubic-spline – кубические сплайны (chart:spline-resolution определяет количество промежуточных точек между точками данных);
- b-spline – B-сплайны (chart:spline-order определяет порядок полигонов, используемых при вычислениях, свойство chart:spline-resolution также используется при расчетах).

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:interpolation" a:defaultValue="none">
            <choice>
                <value>none</value>
                <value>cubic-spline</value>
                <value>b-spline</value>
            </choice>
        </attribute>
    </optional>
    <optional>
        <attribute name="chart:spline-order" a:defaultValue="2">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="chart:spline-resolution" a:defaultValue="20">
            <ref name="positiveInteger"/>
        </attribute>
    </optional>
</define>
```

15.30.8 Свойства круговой диаграммы

Атрибут chart:pie-offset используется только круговыми диаграммами. Он определяет расстояние от края круговой диаграммы (или круга диаграммы) до центра круга.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:pie-offset" a:defaultValue="0">
            <ref name="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

15.30.9 Линии

Свойство chart:lines определяет, показывать ли линии, соединяющие точки данных. Сглаживание линий зависит от свойства chart:splines.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:lines" a:defaultValue="false">
            <ref name="boolean"/>
```

```
        </attribute>
    </optional>
</define>
```

15.30.10 Сплошные гистограммы

Атрибут `chart:solid-type` определяет, как полосы диаграммы должны отображаться в трехмерном виде.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:solid-type" a:defaultValue="cuboid">
            <choice>
                <value>cuboid</value>
                <value>cylinder</value>
                <value>cone</value>
                <value>pyramid</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.30.11 Гистограммы с накоплением

Атрибут `chart:stacked` указывает на то, что размещать полосы диаграммы необходимо друг над другом, вместо расположения их друг за другом. Если значение `chart:percentage` установлено `true`, все накапливающиеся полосы будут масштабированы на всю высоту диаграммы, так что полосы отражают процент от их точки данных в общей полосе диаграммы.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:stacked" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="chart:percentage" a:defaultValue="false">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.31 Свойства осей диаграмм

Свойства, описанные в данном разделе, могут применяться к элементам осей диаграмм (см. раздел 10.8). Они могут использоваться в стилях диаграмм (см. раздел 14.16) и содержаться в элементе `<style:chart-properties>`.

15.31.1 Форматы связанных данных

Атрибут `chart:link-data-style-to-source` может быть использован только в документах с диаграммами, находящимися в документе, который снабжает диаграмму данными. Если значение атрибута — истина, то числовой формат, используемый для отображения осей, — формат, который документ выбирает, основываясь на диапазоне выбранных ячеек. Например, если диапазон ячейки содержит курсы, отформатированные в €, тогда этот формат также будет использован при отображении осей.

```
17043 <define name="style-chart-properties-attlist" combine="interleave">
17044     <optional>
17045         <attribute name="chart:link-data-style-to-source">
17046             <ref name="boolean"/>
17047         </attribute>
17048     </optional>
17049 </define>
```

15.31.2 Видимость

Для определения, будут ли видны оси, используется свойство стиля `chart:axis-visible`. В таком случае диаграмма с информацией о масштабировании может быть предоставлена без отображения осей.

```
17050 <define name="style-chart-properties-attlist" combine="interleave">
17051     <optional>
17052         <attribute name="chart:visible">
17053             <ref name="boolean"/>
17054         </attribute>
17055     </optional>
17056 </define>
```

15.31.3 Масштабирование

Если атрибут, задающий масштаб, пропущен, то устанавливается режим автоматического масштаба осей. Это означает, что не задано фиксированное значение, но оно может быть изменено программой отображения, если данные изменились. При этом значение атрибута `chart:axis-logarithmic` устанавливается `false`.

Необязательный атрибут `chart:axis-logarithmic` может быть использован для установления логарифмического масштаба по осям. По умолчанию используется пропорциональное масштабирование.

```
17057 <define name="style-chart-properties-attlist" combine="interleave">
17058     <optional>
```

```
17059        <attribute name="chart:logarithmic">
17060            <ref name="boolean"/>
17061        </attribute>
17062    </optional>
17063 </define>
```

Следующий набор необязательных атрибутов уточняет масштабирование по осям. Свойства имеют следующие применения:

`chart:minimum`, `chart:maximum` — задает минимальное и максимальное значения по осям;

`chart:origin` — задает начало оси диаграммы;

`chart:interval-major`, `chart:interval-minor-divisor` — задают основной и дополнительный интервалы делений на осях, первый определяет интервал подписи значений, дополнительный интервал вычисляется делением значения `chart:interval-major` на значение `chart:interval-minor-divisor`.

```
17064 <define name="style-chart-properties-attlist" combine="interleave">
17065     <optional>
17066         <attribute name="chart:maximum">
17067             <ref name="double"/>
17068         </attribute>
17069     </optional>
17070     <optional>
17071         <attribute name="chart:minimum">
17072             <ref name="double"/>
17073         </attribute>
17074     </optional>
17075     <optional>
17076         <attribute name="chart:origin">
17077             <ref name="double"/>
17078         </attribute>
17079     </optional>
17080     <optional>
17081         <attribute name="chart:interval-major">
17082             <ref name="double"/>
17083         </attribute>
17084     </optional>
17085     <optional>
17086         <attribute name="chart:interval-minor">
17087             <ref name="positiveInteger"/>
17088         </attribute>
17089     </optional>
17090 </define>
```

15.31.4 Отметки

Свойства отметок используются для задания присутствия данных отметок на осях. Основные отметки нарисованы с основным интервалом, который может быть задан атрибутом `chart:axis-interval-major`. Дополнительные отметки

определяются атрибутом chart:axis-interval-minor. Внутренние отметки рисуются правее на оси, отображенной с левой стороны графика, и левее на оси, отображаемой с правой стороны графика. Внешние отметки — с противоположной стороны. Если оба свойства определены, отметка пересекает ось.

```
17091 <define name="style-chart-properties-attlist" combine="interleave">
17092     <optional>
17093         <attribute name="chart:tick-marks-major-inner">
17094             <ref name="boolean"/>
17095         </attribute>
17096     </optional>
17097     <optional>
17098         <attribute name="chart:tick-marks-major-outer">
17099             <ref name="boolean"/>
17100         </attribute>
17101     </optional>
17102     <optional>
17103         <attribute name="chart:tick-marks-minor-inner">
17104             <ref name="boolean"/>
17105         </attribute>
17106     </optional>
17107     <optional>
17108         <attribute name="chart:tick-marks-minor-outer">
17109             <ref name="boolean"/>
17110         </attribute>
17111     </optional>
17112 </define>
```

15.31.5 Подписи

Следующий набор свойств описывает, как будут отображаться подписи осей. Свойство chart:display-label определяет, отображаются ли подписи. Если chart:text-overlap установлен true, подписи могут перекрываться. Атрибут text:line-break определяет, могут ли подписи разбиваться на несколько строк.

Свойство chart:label-arrangement позволяет расположить подписи либо side-by-side (т. е. все подписи начинаются на одной линии) либо ступенями (т. е. подписи распределены по двум линиям, где каждая подпись начинается на соответствующей строке). В случае ступенчатых подписей можно выбрать между четным и нечетным режимами ступеней, т. е. можно выбрать какие элементы будут находится на уровне, который используется при side-by-side расположении.

```
17113 <define name="style-chart-properties-attlist" combine="interleave">
17114     <optional>
17115         <attribute name="chart:display-label">
17116             <ref name="boolean"/>
17117         </attribute>
17118     </optional>
17119     <optional>
```

```
17120         <attribute name="chart:text-overlap">
17121             <ref name="boolean"/>
17122         </attribute>
17123     </optional>
17124     <optional>
17125         <attribute name="text:line-break">
17126             <ref name="boolean"/>
17127         </attribute>
17128     </optional>
17129     <optional>
17130         <attribute name="chart:label-arrangement"
17131                     a:defaultValue="side-by-side">
17132             <choice>
17133                 <value>side-by-side</value>
17134                 <value>stagger-even</value>
17135                 <value>stagger-odd</value>
17136             </choice>
17137         </attribute>
17138     </optional>
17139 </define>
```

15.32 Общие свойства диаграмм

Свойства, описанные в данном разделе, применимы ко всем типам объектов, представляющих данные, включая элементы `<chart:plot-area>`, `<chart:series>` и `<chart:data-point>`. Они могут использоваться при описании стилей диаграмм (см. раздел 14.16) и содержаться в элементе `<style:chart-properties>`.

Свойства применены в иерархическом виде. Если свойство задано в элементе `<chart:chart>`, то оно применяется для всех точек данных, содержащихся на диаграмме. Если такое же свойство установлено для элемента `<chart:series>`, то оно применяется только для точек данных, содержащихся в определенном ряду данных. Для установки форматирующего свойства только для одной точки данных, установите свойство в элементе `<chart:data-point>`.

15.32.1 Текст с накоплением по вертикали

Свойство `style:direction` определяет, используется ли метод вертикального отображения текста без поворота букв или нет. Он может быть использован для нескольких текстовых объектов.

Значением свойства может быть `ltr`, если текст направлен слева направо, или `ttb`, если текст с накоплением по вертикали, в таком случае он направлен сверху вниз. Он может быть использован для нескольких текстовых объектов (см. раздел 15.11.3).

```
<define name="style-chart-properties-attlist" combine="interleave">
    <ref name="common-style-direction-attlist"/>
</define>
```

15.32.2 Угол поворота

Свойство `style:rotation-angle` определяет значение угла поворота в градусах. Для получения информации об использовании данного свойства см. раздел 15.11.12.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <ref name="common-rotation-angle-attlist"/>
</define>
```

15.32.3 Подписи данных

Подписи данных могут применяться к рядам данных и к точкам данных, как и для диаграмм целиком. В последнем случае подписи отображаются для всех точек данных. Подписи данных могут состоять из трех следующих частей:

- значение, которое может быть отражено непосредственно или в процентах;
- подпись соответствующих рядов;
- символ легенды.

Значение

Атрибут `chart:data-label-number` определяет значение точки данных.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:data-label-number">
            <choice>
                <value>none</value>
                <value>value</value>
                <value>percentage</value>
            </choice>
        </attribute>
    </optional>
</define>
```

Подпись

Атрибут `chart:data-label-text` определяет, будут ли отображаться подписи для соответствующих рядов.

Значение атрибута может быть `true` или `false`.

```
<define name="style-chart-properties-attlist" combine="interleave">
```

```
17158     <optional>
17159         <attribute name="chart:data-label-text">
17160             <ref name="boolean"/>
17161         </attribute>
17162     </optional>
17163 </define>
```

Символ легенды

Атрибут `chart:data-label-symbol` определяет, будет ли отображаться символ легенды. Значением этого атрибута может быть `true` или `false`.

```
17164 <define name="style-chart-properties-attlist" combine="interleave">
17165     <optional>
17166         <attribute name="chart:data-label-symbol">
17167             <ref name="boolean"/>
17168         </attribute>
17169     </optional>
17170 </define>
```

15.33 Статистические свойства

Статистические свойства могут быть применены к рядам данных или к диаграмме целиком. В последнем случае свойства применяются для всех рядов в диаграмме. Они могут применяться в стилях диаграмм (см. раздел 14.16) и содержаться в элементе `<style:chart-properties>`.

15.33.1 Среднее значение

Атрибут `chart:mean-value` определяет, отображать или нет линию, отражающую статистическое среднее значение всех точек данных в ряду. Значением данного атрибута может быть `true` или `false`.

```
17171 <define name="style-chart-properties-attlist" combine="interleave">
17172     <optional>
17173         <attribute name="chart:mean-value">
17174             <ref name="boolean"/>
17175         </attribute>
17176     </optional>
17177 </define>
```

15.33.2 Категория погрешности

Атрибут `chart:error-category` используется для определения функции, которая будет применена для отображения категории погрешности в точках данных. Доступны следующие функции:

- дисперсия значений равномерно распределенного ряда;

- среднеквадратичное отклонение значений равномерно распределенного ряда;
- использование фиксированного процента каждого значения;
- использование фиксированного процента наибольшего значения – границы погрешности;
- использование абсолютных значений обоих направлений: положительного и отрицательного.

Если значение атрибута одно из возможных, кроме `none`, индикаторы погрешности отображаются. Для определения, каким методом индикаторы указываются, смотри атрибуты `chart:error-upper-indicator` и `chart:error-lower-indicator`.

```
17178 <define name="style-chart-properties-attlist" combine="interleave">
17179     <optional>
17180         <attribute name="chart:error-category" a:defaultValue="none">
17181             <choice>
17182                 <value>none</value>
17183                 <value>variance</value>
17184                 <value>standard-deviation</value>
17185                 <value>percentage</value>
17186                 <value>error-margin</value>
17187                 <value>constant</value>
17188             </choice>
17189         </attribute>
17190     </optional>
17191 </define>
```

Процентная погрешность

Атрибут `chart:error-percentage` определяет процент, который используется для отображения индикатора погрешности для каждой точки рядов данных.

```
17192 <define name="style-chart-properties-attlist" combine="interleave">
17193     <optional>
17194         <attribute name="chart:error-percentage">
17195             <ref name="double"/>
17196         </attribute>
17197     </optional>
17198 </define>
```

Граница погрешности

Атрибут `chart:error-margin` определяет предел погрешности в процентах, который используется при отображении индикатора погрешности в соответствии с наибольшим значением ряда.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:error-margin">
            <ref name="double"/>
        </attribute>
    </optional>
</define>
```

Постоянная верхняя и нижняя граница погрешности

Если категория погрешности установлена в `constant`, атрибуты `chart:error-lower-limit` и `chart:error-upper-limit` определяют абсолютное значение в положительном и отрицательном направлениях, которое используется для отображения индикатора погрешности.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:error-lower-limit">
            <ref name="double"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="chart:error-upper-limit">
            <ref name="double"/>
        </attribute>
    </optional>
</define>
```

Индикаторы погрешности

Атрибуты `chart:error-lower-indicator` и `chart:error-upper-indicator` определяют, в каком направлении следует отобразить индикаторы.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:error-upper-indicator">
            <ref name="boolean"/>
        </attribute>
    </optional>
    <optional>
        <attribute name="chart:error-lower-indicator">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.34 Свойства области построения диаграммы

Свойства, описанные в данном разделе, могут применяться к элементам области построения диаграммы (см. раздел 10.5). Они могут использоваться в стилях диаграмм (см. раздел 14.16), содержатся они в элементе `<style:chart-properties>`.

15.34.1 Источник рядов данных

Атрибут `chart:series-source` определяет содержание рядов данных в постолбцовом или построчном виде.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:series-source" a:defaultValue="columns">
            <choice>
                <value>columns</value>
                <value>rows</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.35 Свойства кривой регрессии

Свойства, описанные в данном разделе, могут применяться к элементам кривой регрессии диаграммы (см. раздел 10.14). Они могут использоваться в стилях диаграмм (см. раздел 14.16) и содержаться в элементе `<style:chart-properties>`.

15.35.1 Тип регрессии

Используйте атрибут `chart:regression-type`, чтобы отобразить регрессию для рядов. Регрессия может использоваться для аппроксимации точек данных в рядах посредством математической функции. Доступны следующие модели для аппроксимации:

- линейная регрессия — аппроксимирует значения в рядах, используя модель $y = A \cdot x + B$;
- логарифмическая регрессия — аппроксимирует значения рядов, используя модель $y = A \cdot \log(x) + B$;
- экспоненциальная регрессия — аппроксимирует значения рядов, используя модель $y = A \cdot e^{B \cdot x}$;
- потенциальная регрессия — аппроксимирует значения рядов, используя модель $y = A \cdot x^{B}$.

Это свойство используется только для диаграмм XY, потому что регрессия для вычислений требует оба значения, как *x*, так и *y*.

```
<define name="style-chart-properties-attlist" combine="interleave">
    <optional>
        <attribute name="chart:regression-type" a:defaultValue="none">
```

```
            <choice>
                <value>none</value>
                <value>linear</value>
                <value>logarithmic</value>
                <value>exponential</value>
                <value>power</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.36 Атрибуты страницы презентации

Свойства, описанные в данном разделе, могут содержаться в элементах стиля `<style:style>` семейства `drawing-page`. Они содержатся в элементе `<style:style-drawpage-properties>`.

Существуют следующие свойства:

- тип перехода;
- стиль перехода;
- скорость перехода;
- длительность показа страницы;
- видимость страницы;
- звук;
- размер фона;
- видимость фоновых объектов;
- видимость фона;
- отображение заголовка;
- отображение нижнего колонтитула;
- отображение номера страницы;
- отображение даты и времени.

15.36.1 Тип перехода

Режим перехода, например вручную, может быть задан атрибутом `presentation:transition-type`:

- `manual` — переход и эффект начинаются отдельно по команде пользователя слайда;
- `automatic` — переход и эффект начинаются автоматически;

- `semi-automatic` — переход осуществляется автоматически, эффект — по команде пользователя.

```
<define name="style-drawing-page-properties-attlist"
        combine="interleave">
    <optional>
        <attribute name="presentation:transition-type">
            <choice>
                <value>manual</value>
                <value>automatic</value>
                <value>semi-automatic</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.36.2 Стиль перехода

Атрибут `presentation:transition-style` определяет способ перехода от предыдущего слайда к следующему, например замена слева направо или затухание.

- `none` — без эффекта;
- `fade-*` — страница переходит из видимого или скрытого состояния в скрытое или видимое состояние определенным образом;
- `move-*` — страница перемещается определенным способом в свою конечную позицию;
- `uncover-*` — страница проявляется в указанном направлении;
- `*-stripes` — страница появляется за счет прорисовки вертикальной или горизонтальных полос, которые изменяют свой размер во время эффекта;
- `clockwise` — страница проявляется стрелкой часов, которая движется по часовой стрелке;
- `counterclockwise` — страница проявляется стрелкой часов, которая движется против часовой стрелки;
- `open-*` — страница проявляется, отображаясь линия за линией горизонтально или вертикально, начиная от центра страницы;
- `close-*` — страница проявляется, отображаясь линия за линией горизонтально или вертикально, начиная от границ страницы;
- `wavyline-*` — страница проявляется, прорисовываясь малыми блоками подобно змейке;

- `spiralin-*` — страница проявляется, прорисовываясь блоками по спирали, начиная от края страницы;
- `spiralout-*` — страница проявляется, прорисовываясь блоками по спирали, начиная от центра страницы;
- `roll-*` — страница перемещается в указанном направлении, выталкивая предыдущую;
- `stretch-*` — страница проявляется, изменяя свой размер во время эффекта;
- `*-lines` — страница проявляется, отображаясь линия за линией горизонтально или вертикально в произвольном порядке;
- `dissolve` — страница исчезает путем прорисовки маленьких квадратиков в произвольном порядке;
- `random` — эффект проявления страницы выбирается случайным образом;
- `*-checkerboard` — страница проявляется, прорисовываясь блоками шахматной доски, которые увеличиваются в размерах по горизонтали или по вертикали;
- `interlocking-horizontal-*` — новая страница проявляется из 4 горизонтальных полос (т. е. высота делится на 4 так же, как при эффекте «горизонтальные линии»), но эти линии идут попеременно слева и справа и пересекаются в середине экрана;
- `interlocking-vertical-*` — такой же эффект с вертикальными линиями, пересекающими друг друга;
- `fly-away` — страница сначала уменьшается в размере (пока оставаясь в центре экрана) и затем «улетает» (немного поворачиваясь и перемещаясь в нижний правый угол экрана), следующая страница находится под ней;
- `open` — комбинация открытия горизонтального и вертикального, т. е. своего рода вид открытия знаком плюс;
- `close` — комбинация закрытия горизонтального и вертикального, т. е. своего рода вид закрытия знаком плюс;
- `melt` — небольшие вертикальные полосы, перемещающиеся вниз со случайной скоростью, создающие эффект «таяния» на текущей странице.

```
<define name="style-drawing-page-properties-attlist"
         combine="interleave">
    <optional>
        <attribute name="presentation:transition-style">
            <choice>
                <value>none</value>
                <value>fade-from-left</value>
                <value>fade-from-top</value>
                <value>fade-from-right</value>
                <value>fade-from-bottom</value>
                <value>fade-from-upperleft</value>
                <value>fade-from-upperright</value>
                <value>fade-from-lowerleft</value>
                <value>fade-from-lowerright</value>
                <value>move-from-left</value>
                <value>move-from-top</value>
                <value>move-from-right</value>
                <value>move-from-bottom</value>
                <value>move-from-upperleft</value>
                <value>move-from-upperright</value>
                <value>move-from-lowerleft</value>
                <value>move-from-lowerright</value>
                <value>uncover-to-left</value>
                <value>uncover-to-top</value>
                <value>uncover-to-right</value>
                <value>uncover-to-bottom</value>
                <value>uncover-to-upperleft</value>
                <value>uncover-to-upperright</value>
                <value>uncover-to-lowerleft</value>
                <value>uncover-to-lowerright</value>
                <value>fade-to-center</value>
                <value>fade-from-center</value>
                <value>vertical-stripes</value>
                <value>horizontal-stripes</value>
                <value>clockwise</value>
                <value>counterclockwise</value>
                <value>open-vertical</value>
                <value>open-horizontal</value>
                <value>close-vertical</value>
                <value>close-horizontal</value>
                <value>wavyline-from-left</value>
                <value>wavyline-from-top</value>
                <value>wavyline-from-right</value>
                <value>wavyline-from-bottom</value>
                <value>spiralin-left</value>
                <value>spiralin-right</value>
                <value>spiralout-left</value>
                <value>spiralout-right</value>
                <value>roll-from-top</value>
                <value>roll-from-left</value>
                <value>roll-from-right</value>
                <value>roll-from-bottom</value>
                <value>stretch-from-left</value>
                <value>stretch-from-top</value>
                <value>stretch-from-right</value>
                <value>stretch-from-bottom</value>

                <value>vertical-lines</value>
                <value>horizontal-lines</value>
                <value>dissolve</value>
                <value>random</value>
                <value>vertical-checkerboard</value>
                <value>horizontal-checkerboard</value>
```

```
                <value>interlocking-horizontal-left</value>
                <value>interlocking-horizontal-right</value>
                <value>interlocking-vertical-top</value>
                <value>interlocking-vertical-bottom</value>
                <value>fly-away</value>
                <value>open</value>
                <value>close</value>
                <value>melt</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.36.3 Скорость перехода

Атрибут `presentation:transition-speed` управляет скоростью, с которой предыдущий слайд заменяется следующим (см. также раздел 9.7.2).

```
<define name="style-drawing-page-properties-attlist"
        combine="interleave">
    <optional>
        <attribute name="presentation:transition-speed">
            <ref name="presentationSpeeds"/>
        </attribute>
    </optional>
</define>
```

15.36.4 Тип или семейство переходов

Атрибут [SMIL20] `smil:type` используется для определения типа или семейства перехода. Для получения дополнительной информации см. § 12.4.1 [SMIL20]. Список поддерживаемых типов см. в § 12.8 [SMIL20].

Если этот атрибут установлен, то атрибуты `presentation:transition-type` и `presentation:transition-style` должны игнорироваться.

```
<define name="style-drawing-page-properties-attlist " combine="interleave">
    <optional>
        <attribute name="smil:type">
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

15.36.5 Подтип перехода

Атрибут [SMIL20] `smil:subtype` используется для определения подтипа перехода. Для получения дополнительной информации см. § 12.4.1 [SMIL20], для получения списка поддерживаемых подтипов см. § 12.8 [SMIL20] .

```
<define name="style-drawing-page-properties-attlist" combine="interleave">
    <optional>
        <attribute name="smil:subtype">
```

```
            <ref name="string"/>
        </attribute>
    </optional>
</define>
```

15.36.6 Управление переходом

Атрибут [SMIL20] `smil:direction` используется для определения управления перехода. Для получения дополнительной информации см. § 12.4.1 [SMIL20].

```
<define name="style-drawing-page-properties-attlist" combine="interleave">
    <optional>
        <attribute name="smil:direction" a:defaultValue="forward">
            <choice>
                <value>forward</value>
                <value>reverse</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.36.7 Цвет выцветания

Атрибут [SMIL20] `smil:fadeColor` используется для определения цвета выцветания при переходе, который может применяться как начальный или конечный цвет. Для получения дополнительной информации см. § 12.4.1 [SMIL20].

```
<define name="style-drawing-page-properties-attlist" combine="interleave">
    <optional>
        <attribute name="smil:fadeColor">
            <choice>
                <value>forward</value>
                <value>reverse</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.36.8 Длительность показа страницы

Атрибут `presentation:page-duration` задает период времени, в течение которого отображается страница. Значение этого атрибута должно удовлетворять формату периода, описанному в § 3.2.6 [xmlschema-2].

```
<define name="style-drawing-page-properties-attlist"
        combine="interleave">
    <optional>
        <attribute name="presentation:duration">
            <ref name="duration"/>
        </attribute>
    </optional>
</define>
```

15.36.9 Видимость страницы

Страница может быть помечена как скрытая во время показа презентации с использованием атрибута `presentation:visibility`. Страница, отмеченная данным атрибутом, отображается только в момент редактирования документа, но не видна во время показа презентации.

```
<define name="style-drawing-page-properties-attlist"
         combine="interleave">
    <optional>
        <attribute name="presentation:visibility">
            <choice>
                <value>visible</value>
                <value>hidden</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.36.10 Звук

Звуковые эффекты могут быть добавлены на странице презентации использованием элемента `presentation:sound`. Он должен включаться в элемент `<style:presentation-properties>`.

```
<define name="style-drawing-page-properties-elements"
         combine="interleave">
    <optional>
        <ref name="presentation-sound"/>
    </optional>
</define>
```

15.36.11 Размер фона

Атрибут `draw:background-size` определяет, заполняет ли фон всю страницу или только внутреннюю часть обрамления страницы.

```
<define name="style-drawing-page-properties-attlist"
         combine="interleave">
    <optional>
        <attribute name="draw:background-size">
            <choice>
                <value>full</value>
                <value>border</value>
            </choice>
        </attribute>
    </optional>
</define>
```

15.36.12 Видимость объектов фона

Атрибут `presentation:background-objects-visible` определяет, будут ли отображаться объекты фона мастер-страницы при показе страницы презентации.

```
<define name="style-drawing-page-properties-attlist"
         combine="interleave">
    <optional>
        <attribute name="presentation:background-objects-visible">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.36.13 Видимость фона

Атрибут `presentation:background-visible` определяет, будет ли отображаться фон мастер-страницы при показе страницы презентации.

```
<define name="style-drawing-page-properties-attlist"
         combine="interleave">
    <optional>
        <attribute name="presentation:background-visible">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.36.14 Отображение заголовков

Атрибут `presentation:display-header` задает видимость презентационных фигур мастер-страницы в классе `header` (см. раздел 9.6.1).

```
<define name="style-drawing-page-properties-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:display-header">
            <ref name="boolean"/>
        </attribute>
    </optional>
</define>
```

15.36.15 Отображение колонтитулов

Атрибут `presentation:display-footer` задает видимость презентационных фигур мастер-страницы в классе `footer` (см. раздел 9.6.1).

```
<define name="style-drawing-page-properties-attlist" combine="interleave">
    <optional>
        <attribute name="presentation:display-footer">
            <ref name="boolean"/>
        </attribute>
    </optional>
```

```
17447 </define>
```

15.36.16 Отображение номера страницы

Атрибут `presentation:display-page-number` задает видимость презентационных фигур мастер-страницы в классе `page-number` (см. раздел 9.6.1).

```
17448 <define name="style-drawing-page-properties-attlist" combine="interleave">
17449     <optional>
17450         <attribute name="presentation:display-page-number">
17451             <ref name="boolean"/>
17452         </attribute>
17453     </optional>
17454 </define>
```

15.36.17 Отображение даты и времени

Атрибут `presentation:display-date-time` задает видимость презентационных фигур мастер-страницы в классе `date-time` (см. раздел 9.6.1).

```
17455 <define name="style-drawing-page-properties-attlist" combine="interleave">
17456     <optional>
17457         <attribute name="presentation:display-date-time">
17458             <ref name="boolean"/>
17459         </attribute>
17460     </optional>
17461 </define>
```

16 Типы данных и определения в схеме

16.1 Типы данных

В данной спецификации используются следующие типы данных.

- Определения типов данных W3C-схемы даны в [xmlschema-2] (ссылка на элементы `<ref>` также присваивает имя соответствующим типам данных)
  - string (строка);
  - date (дата);
  - time (время);
  - dateTime (датаВремя);
  - duration (длительность);
  - integer (целочисленный);
  - nonNegativeInteger (целочисленный неотрицательный);
  - positiveInteger (целочисленный положительный);
  - double (численный с двойной точностью);
  - anyURI (общепринятый идентификатор ресурса);
  - base64Binary (последовательность бинарных октетов произвольной длины);
  - ID (метка);
  - IDREF (ссылка на метку).

Определения типов данных Relax-NG для W3C-схемы:

```
<define name="string">
    <data type="string"/>
</define>
<define name="date">
    <data type="date"/>
</define>
<define name="time">
    <data type="time"/>
</define>
<define name="dateTime">
    <data type="dateTime"/>
</define>
<define name="duration">
    <data type="duration"/>
</define>
<define name="integer">
    <data type="integer"/>
</define>
```

```
17480 <define name="nonNegativeInteger">
17481     <data type="nonNegativeInteger"/>
17482 </define>
17483 <define name="positiveInteger">
17484     <data type="positiveInteger"/>
17485 </define>
17486 <define name="double">
17487     <data type="double"/>
17488 </define>
17489 <define name="anyURI">
17490     <data type="anyURI"/>
17491 </define>
17492 <define name="base64Binary">
17493     <data type="base64Binary"/>
17494 </define>
17495 <define name="ID">
17496     <data type="ID"/>
17497 </define>
17498 <define name="IDREF">
17499     <data type="IDREF"/>
17500 </define>
```

- Типы данных пользователя (обычно это конкретизация типов данных W3C- схемы)

— boolean (логический) — может принимать только одно из двух значений — `true` (истина) или `false` (ложь);

— dateOrDateTime (дата или датаВремя) — значение dateOrDateTime по существу такое же, как в схеме [xmlschema-2], это значение даты и времени с необязательным компонентом времени, другими словами, оно может содержать только дату или дату и время;

— timeOrDateTime (время или датаВремя) — значение timeOrDateTime по существу такое же, как в схеме [xmlschema-2], это значение даты и времени с необязательным компонентом даты, другими словами, оно может содержать только время или дату и время;

— language (язык) — то же самое, что тип данных language в схеме W3C, за исключением того, что значения описываются в [RFC3066] вместо [RFC1766];

— countryCode (код страны) — значение countryCode — код страны в соответствии с [RFC3066], как определено в [XSL];

— languageCode (код языка) — значение languageCode — код языка в соответствии с [RFC3066], как определено в [XSL];

— character (символьный) — значение character — строка, состоящая только из одного символа;

— length (длина) — физическая длина (положительная или отрицательная), состоящая из величины и единицы измерения, в соответствии с § 5.9.11 в [XSL], корректное значение длины может быть «2.54 cm» или «1 inch», число без единицы измерения — некорректное значение длины, например, «3.2», поддержка значений в пикселях необязательна;
— nonNegativeLength (неотрицательная длина) — похож на тип данных length, за исключением того, что значение должно быть нулевым или положительным;
— positiveLength (положительная длина) — похож на тип данных length, за исключением того, что значение должно быть положительным;
— percent (процентный) — значение в процентах (положительное или отрицательное) в соответствие с § 5.9.11 в [XSL], например, «40 %»;
— relativeLength (относительная длина) — положительное целое число, следующее за символом « * »;
— coordinate (координатный) — похож на тип данных length, за исключением того, что физическая длина обозначает конкретную точку;
— distance (расстояние) — похож на тип данных length, за исключением того, что физическая длина указывает расстояние между точками;
— color (цвет) — RGB-цвет в соответствие с § 5.9.11 в [XSL], т. е. RGB-цвет в "#rrggbb" нотации, где rr, gg и bb — шестнадцатеричные цифры;
— styleName (имя стиля) — NCName, как определено в [xmlschema-2], — это имя стиля;
— StyleNameRef (ссылка на имя стиля) — NCName, как определено в [xmlschema-2], — ссылка на имя стиля или пустое значение;
— StyleNames (имена стилей) — список, состоящий из NCNames, разделенный пробельными символами, как определено в [xmlschema-2] — имена стилей;
— VariableName (имя переменной) — строка, определяющая имя переменной;
— formula (формула) — строка, содержащая формулу, формулы не имеют встроенного синтаксиса, но должны начинаться с префикса

пространства имен, который конкретизирует синтаксис, используемый в пределах формулы;

— valueType (тип значения) — список поддерживаемых значений для определенных общих значений типов, таких как string или date;

— targetFrameName (имя целевого фрейма) — имя целевого фрейма в соответствии с § 6.16 [HTML4];

— points (пункты) — последовательность пунктов, пункты — две целых координаты, разделенные запятой, пункты разделяются пробельным символом;

— pathData (путь к данным) — путь к данным, как описано в § 8 из [SVG];

— vector3D (трехмерный вектор) — вектор из трех элементов с плавающей точкой, представленный координатами x,y,z, заключенными в круглые скобки, записанными в порядке x, y, z и разделенными пробельными символами, если значение представлено как нормальное, то оно должно быть нормализовано.

Пример —

Направленный вектор с координатами x = 0.5, y = 0 и z = 1 выглядит как "(0.5 0 1)".

— namespacedToken (маркер пространства имен) — маркер пространства имен — это идентификатор маркера, который использует механизм пространства имен XML для целей компоновки.

Пример —

Предопределенные типы диаграмм используют пространство имен диаграмм `urn:oasis:names:tc:opendocument:xmlns:chart:1.0`. Допускается описание пространства имен `xmlns:chart="urn:oasis:names:tc:opendocument:xmlns:chart:1.0"`, столбиковая диаграмма была бы идентифицирована как `chart:bar`.

Определения типов данных Relax-NG для типов данных пользователя:

```
<define name="boolean">
    <choice>
        <value>true</value>
        <value>false</value>
    </choice>
</define>
<define name="dateOrDateTime">
    <choice>
        <data type="date"/>
```

```
17510         <data type="dateTime"/>
17511     </choice>
17512 </define>
17513 <define name="timeOrDateTime">
17514     <choice>
17515         <data type="time"/>
17516         <data type="dateTime"/>
17517     </choice>
17518 </define>
17519 <define name="language">
17520     <data type="token">
17521         <param name="pattern">[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*</param>
17522     </data>
17523 </define>
17524 <define name="countryCode">
17525     <data type="token">
17526         <param name="pattern">[A-Za-z0-9]{1,8}</param>
17527     </data>
17528 </define>
17529 <define name="languageCode">
17530     <data type="token">
17531         <param name="pattern">[A-Za-z]{1,8}</param>
17532     </data>
17533 </define>
17534 <define name="character">
17535     <data type="string">
17536         <param name="length">1</param>
17537     </data>
17538 </define>
17539 <define name="length">
17540     <data type="string">
17541         <param name="pattern">-?([0-9]+(\.[0-9]*)?|\.[0-9]+)((cm)|(mm)|(in)|
17542 (pt)|(pc)|(px))</param>
17543     </data>
17544 </define>
17545 <define name="nonNegativeLength">
17546     <data type="string">
17547         <param name="pattern">([0-9]+(\.[0-9]*)?|\.[0-9]+)((cm)|(mm)|(in)|(pt)|
17548 (pc)|(px))</param>
17549     </data>
17550 </define>
17551 <define name="positiveLength">
17552     <data type="string">
17553         <!-- A zero value is not allowed here -->
17554         <param name="pattern">([0-9]+(\.[0-9]*)?|\.[0-9]+)((cm)|(mm)|(in)|(pt)|
17555 (pc)|(px))</param>
17556     </data>
17557 </define>
17558 <define name="percent">
17559     <data type="string">
17560         <param name="pattern">-?([0-9]+(\.[0-9]*)?|\.[0-9]+)%</param>
17561     </data>
17562 </define>
17563 <define name="relativeLength">
17564     <data type="string">
17565         <param name="pattern">[0-9]+\*</param>
17566     </data>
17567 </define>
17568 <define name="coordinate">
17569     <ref name="length"/>
17570 </define>
17571 <define name="distance">
```

```
17572     <ref name="length"/>
17573 </define>
17574 <define name="color">
17575     <data type="string">
17576         <param name="pattern">#[0-9a-fA-F]{6}</param>
17577     </data>
17578 </define>
17579 <define name="styleName">
17580     <data type="NCName"/>
17581 </define>
17582 <define name="styleNameRef">
17583     <choice>
17584         <data type="NCName"/>
17585         <empty/>
17586     </choice>
17587 </define>
17588 <define name="styleNameRefs">
17589     <list>
17590         <zeroOrMore>
17591             <data type="NCName"/>
17592         </zeroOrMore>
17593     </list>
17594 </define>
17595 <define name="variableName">
17596     <data type="string"/>
17597 </define>
17598 <define name="formula">
17599     <!-- A formula should start with a namespace prefix, -->
17600     <!-- but has no restrictions-->
17601     <data type="string"/>
17602 </define>
17603
17604 <define name="targetFrameName">
17605     <choice>
17606         <value>_self</value>
17607         <value>_blank</value>
17608         <value>_parent</value>
17609         <value>_top</value>
17610         <ref name="string"/>
17611     </choice>
17612 </define>
17613
17614 <define name="valueType">
17615     <choice>
17616         <value>float</value>
17617         <value>time</value>
17618         <value>date</value>
17619         <value>percentage</value>
17620         <value>currency</value>
17621         <value>boolean</value>
17622         <value>string</value>
17623     </choice>
17624 </define>
17625
17626 <define name="points">
17627     <data type="string">
17628         <param name="pattern">-?[0-9]+,-?[0-9]+([ ]+-?[0-9]+,-?[0-9]+)*</param>
17629     </data>
17630 </define>
17631 <define name="pathData">
17632     <data type="string"/>
17633 </define>
17634
```

```
<define name="vector3D">
    <data type="string">
        <param name="pattern">\([ ]*-?([0-9]+(\.[0-9]*)?|\.[0-9]+)([ ]+-?([0-9]+(\.[0-9]*)?|\.[0-9]+)){2}[ ]*\)</param>
    </data>
</define>

<define name="namespacedToken">
    <data type="string">
        <param name="pattern">[0-9a-zA-Z_]+:[0-9a-zA-Z._\-]+</param>
    </data>
</define>
```

16.2 Другие определения

Для того чтобы предусмотреть расширяемость формата, в некоторых случаях возможно включение содержимого пользователя. Следующие определения позволяют добавление произвольных свойств или элементов (с произвольными моделями содержимого).

```
<define name="anyAttListOrElements">
    <zeroOrMore>
        <attribute>
            <anyName/>
            <text/>
        </attribute>
    </zeroOrMore>
    <ref name="anyElements"/>
</define>
<define name="anyElements">
    <zeroOrMore>
        <element>
            <anyName/>
            <mixed>
                <ref name="anyAttListOrElements"/>
            </mixed>
        </element>
    </zeroOrMore>
</define>
```

16.3 Суффикс схемы Relax-NG

Суффикс для нормативной схемы Relax-NG:

```
</grammar>
```

17 Пакеты

В этом разделе описан формат пакета, который может быть необязательно использован в OpenDocument. Он содержит следующие подразделы:

- введение;
- структура zip-файла;
- шифрование;
- изображение предварительного просмотра;
- файл декларации.

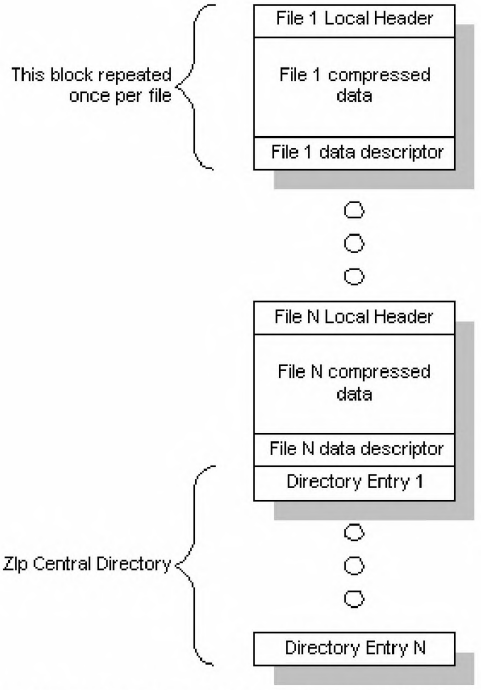
17.1 Введение

Поскольку XML не имеет никакой собственной поддержки двоичных объектов типа изображений, [OLE]-объектов, или других медиа-типов, а также несжатые XML-файлы могут стать очень большими, OpenDocument использует пакетный файл, чтобы хранить содержимое XML-документа вместе со своими связанными двоичными данными, и произвольно сжимает содержимое XML. Пакет — это стандартный zip-файл, структура которого рассмотрена ниже.

Информация о файлах, содержащихся в пакете, сохраняется в XML-файле, называемом файлом декларации. Файл декларации всегда сохраняется в каталоге META-INF с именем файла META-INF/manifest.xml. В декларации записываются следующие основные информационные части:

- список всех файлов в пакете;
- медиа-тип каждого файла в пакете;
- если файл, сохраненный в пакете, зашифрован, в декларации сохраняется информация, необходимая для его расшифровки.

17.2 Структура zip-файла



Zip-файл начинается с последовательности файлов, каждый из которых может быть сжат или сохранен в необработанном формате. Каждый файл, непосредственно перед своими данными, имеет локальный заголовок, который содержит наибольшее количество информации о файле, включая временные метки, метод сжатия и имя файла. Содержание сжатого файла следует непосредственно далее и заканчивается необязательным дескриптором данных. Дескриптор данных содержит CRC (циклический избыточный код) и размер сжатого файла, которые часто не доступны при записи локального заголовка файла. Дескриптор данных может быть пропущен, если эти детали уже были включены в заголовок.

В приведенном формате каждый файл в архиве располагается последовательно и сопровождается центральным каталогом в конце zip-архива. Центральный каталог — это непрерывное множество элементов каталога, каждый из которых содержит всю информацию в локальном заголовке файла, плюс дополнения, типа комментариев и атрибутов файла. Очень важно, что центральный каталог содержит указатели на позицию каждого файла в архиве, которые делают навигацию по zip-файлу быстрой и простой.

Для получения более подробной информации о формате zip-файла см. [ZIP].

17.3 Шифрование

Процесс шифрования состоит из нескольких стадий.

- Создание 20-байтового SHA1 результирующего дайджеста пароля, введенного пользователем, и его передача компоненту пакета.
- Компонент пакета инициализирует генератор случайных чисел с текущим временем.
- Генератор случайных чисел используется, чтобы генерировать случайный 8-байтовый вектор инициализации и 16-байтовый шум для каждого файла.
- Шум используется вместе с 20-байтовым SHA1 результирующим дайджестом пароля для получения уникального для каждого файла 128-битного ключа. Алгоритмом получения ключа является алгоритм PBKDF2, использующий HMAC-SHA-1 (см. [RFC2898]) с количеством итераций, равным 1024.
- Полученный ключ используется вместе с вектором инициализации для шифрования файла, с использованием алгоритма Blowfish в режиме CFB (cipher-feedback).

Каждый зашифрованный файл сжимается перед шифрованием. Для разрешения проверки содержимого файла пакета необходимо, чтобы зашифрованные файлы были помечены как 'STORED' а не 'DEFLATED'. Так как элементы, помеченные как 'STORED', должны иметь размер, равный сжатому размеру, необходимо хранить их несжатый размер в декларации. Сжатый размер сохраняется как в локальном заголовке файла, так и в записи центрального каталога zip-файла.

17.4 Поток типа MIME

Если для документа, который использует пакеты, существует тип MIME, то в пакете **следует** размещать поток, названный «mimetype». Этот поток **следует** размещать первым потоком zip-файла пакета, он **не должен** быть сжатым и **не должен** использовать дополнительную область в своем заголовке (см. [ZIP]).

Цель состоит в том, чтобы позволить упакованным файлам быть идентифицированными через механизм «магических чисел», похожий на Unix-утилиту file/magic. Если zip-файл в начале содержит поток, который распакован, и не имеет никаких дополнительных данных в заголовке, то имя потока и его содержимое можно найти в фиксированных позициях. Более подробно:

- строка 'PK' в нулевой позиции всех zip-файлов;
- строка 'mimetype' в 30-ой позиции всех таких файлов пакета;
- тип MIME непосредственно в 38-й позиции такого пакета.

17.5 Использование унифицированных идентификаторов ресурса (IRI) в пакетах

Относительные унифицированные идентификаторы ресурса (IRI) используются в пределах файла, содержащегося в пакете, чтобы сослаться на другие файлы пакета, но могут также применяться для обращения к файлам в пределах файловой системы.

Для IRI, которые используются в пределах пакета, существуют следующие ограничения:

- можно сослаться только на файлы в пределах того же самого пакета;
- IRI, которые ссылаются на файл пакета, **должны** быть относительными и **не должны** содержать пути, которые находятся за пределами пакета, это означает, что **не должно** быть ссылок на файлы пакета с абсолютными IRI;
- на файл пакета нельзя ссылаться снаружи пакета, например из файловой системы или из другого пакета.

Ссылка относительного пути (как описано в § 6.5 [RFC3987]), которая встречается в файле из пакета, должна быть разрешена точно так же, как если бы целый пакет был бы разархивирован в каталог с текущим местоположением. Чтобы получить (разархивированный) файл, который содержит ссылку относительного пути, должны использоваться базовые IRI для разрешения ссылок относительного пути.

Все другие типы ссылок IRI, а именно, начинающиеся с протокола (например http:), разделителя (т. е. //) или абсолютного пути (т. е. /) не нуждаются в какой-либо специальной обработке. Это означает, что абсолютные пути не являются ссылками файлов внутри пакета, а содержатся в пределах иерархии, в которой находится пакет, например файловая система. Ссылки IRI в пакете могут выходить за пределы пакета, но как только они вышли за пределы пакета, они никогда не смогут вернуться в него или в другой пакет.

17.6 Изображение предварительного просмотра

По умолчанию, когда файл сохраняется, должно быть сформировано изображение эскиза документа. Оно должно отображать первую страницу документа, первый лист, и т.д. Для обеспечения возможности всестороннего использования эскизов они должны быть сформированы без каких-либо эффектов, окружающих рамок, или обрамления. Такие эффекты могут наложиться на эффекты, добавленные к эскизам различными проводниками файловой системы или, в некоторых случаях, вообще могут быть не желательны для использования.

Эскиз должен быть сохранен как «thumbnail.png» в отдельной папке по имени «Thumbnails».

Папка «Thumbnails» не должна получить медиа-тип в файле manifest.xml, так как она, фактически, не является частью документа.

Зашифрованные файлы предназначены для того, чтобы неавторизованные пользователи не смогли их прочесть, именно поэтому эскиз для таких файлов не генерируется. Вместо того чтобы сохранять эскиз первой страницы, для зашифрованного файла сохраняется отображение, которое не зависит от содержания документа и делает очевидным тот факт, что соответствующий файл зашифрован.

Чтобы соответствовать стандарту управления эскизами (TMS) на www.freedesktop.org, эскизы должны быть сохранены как 24-битное построчное изображение PNG с полной альфа-прозрачностью. Необходимый для эскизов размер — 128x128 пикселей.

17.7 Файл декларации

Элементы и атрибуты в файле декларации находятся в пространстве имен: urn:oasis:names:tc:opendocument:xmlns:manifest:1.0.

17.7.1 Схема Relax-NG

В данной спецификации представлена нормативная XML-схема для файлов декларации OpenDocument. Она может быть получена из данного документа

спецификации связыванием всех фрагментов схемы, содержащихся в этих разделах. Все фрагменты схемы имеют нумерацию строк и серый цвет фона.

Язык схемы, используемый в пределах этой спецификации, — Relax-NG (см. [RNG]).

Префикс для нормативной схемы Relax-NG декларации:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!--
3      OASIS OpenDocument v1.0 (Second Edition)
4      Спецификация комитета №1, 19 Jul 2006
5      Relax-NG Manifest Schema
6
7      $Id$
8
9      © 2002-2005 OASIS Open
10     © 1999-2005 Sun Microsystems, Inc.
11 -->
12
13 <grammar
14     xmlns="http://relaxng.org/ns/structure/1.0"
15
16     datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes"
17
18     xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0">
```

17.7.2 Корневой элемент декларации

Корневой элемент называют декларацией (manifest). Корневой элемент содержит фиксированный атрибут, который определяет пространство имен, как описано выше, и множественные `<manifest:file-entry>` элементы, каждый из которых описывает один файл в пакете.

```
19 <define name="manifest">
20     <element name="manifest:manifest">
21         <oneOrMore>
22             <ref name="file-entry"/>
23         </oneOrMore>
24     </element>
25 </define>
26
27 <start>
28     <choice>
29         <ref name="manifest"/>
30     </choice>
31 </start>
```

17.7.3 Включение файла

Элемент `<manifest:file-entry>` представляет один файл, хранит в пакете его местоположение, тип MIME и необязательные данные, требуемые для расшифровки этого файла.

Каталоги получают свои `<manifest:file-entry>` включения, только если они имеют наследуемую семантику. Например, каталог, представляющий поддокумент, на который ссылаются как на объект из основного документа, содержал бы `<manifest:file-entry>` с подходящим медиа-типом. Каталог для целей администрирования или целей удобства, типа каталога, который содержит различные загрузочные модули, не получил бы включение в файле декларации.

```
<define name="file-entry">
    <element name="manifest:file-entry">
        <ref name="file-entry-attlist"/>
        <optional>
            <ref name="encryption-data"/>
        </optional>
    </element>
</define>
```

С элементом `<manifest:file-entry>` связаны следующие атрибуты:

- полный путь;
- размер;
- медиа-тип.

Полный путь

Атрибут `manifest:full-path` описывает местоположение файла внутри пакета.

```
<define name="file-entry-attlist" combine="interleave">
    <attribute name="manifest:full-path">
        <data type="string"/>
    </attribute>
</define>
```

Размер

Атрибут `manifest:size` присутствует, только если файл сохранен в шифрованном формате. Причина, из-за которой требуется этот атрибут, объясняется в разделе 17.3. Этот атрибут используется только для зашифрованных файлов.

```
<define name="file-entry-attlist" combine="interleave">
    <optional>
        <attribute name="manifest:size">
            <data type="nonNegativeInteger"/>
        </attribute>
    </optional>
</define>
```

Медиа-тип

Атрибут `manifest:media-type` указывает тип MIME определяемого файла. Для ознакомления с полным списком типов MIME см. http://www.isi.edu/in-notes/iana/assignments/media-types/media-types. Как пример, все потоки XML имеют медиа-тип «text/xml».

```
<define name="file-entry-attlist" combine="interleave">
    <attribute name="manifest:media-type">
        <data type="string"/>
    </attribute>
</define>
```

17.7.4 Данные шифрования

Элемент `<manifest:encryption-data>` содержит полную информацию, необходимую для расшифровки файла.

```
<define name="encryption-data">
    <element name="manifest:encryption-data">
        <ref name="encryption-data-attlist"/>
        <ref name="algorithm"/>
        <ref name="key-derivation"/>
    </element>
</define>
```

Элемент `<encryption-data>` содержит в себе следующие элементы:

- алгоритм;
- источник ключа.

Тип контрольной суммы

Атрибут `manifest:checksum-type` определяет название алгоритма формирования дайджеста и может быть использован для проверки пароля. На текущий момент поддерживается только алгоритм SHA1 формирования дайджеста.

```
<define name="encryption-data-attlist" combine="interleave">
    <attribute name="manifest:checksum-type">
        <data type="string"/>
    </attribute>
</define>
```

Контрольная сумма

Атрибут `manifest:checksum` определяет BASE64-кодированный дайджест (определенный в [RFC2045]), который может использоваться, чтобы проверить

правильность пароля способом, описанным в атрибуте manifest:checksum-type.

```
<define name="encryption-data-attlist" combine="interleave">
    <attribute name="manifest:checksum">
        <data type="base64Binary"/>
    </attribute>
</define>
```

17.7.5 Алгоритм

Элемент <manifest:algorithm> содержит информацию об алгоритме, который используется для шифрования данных.

```
<define name="algorithm">
    <element name="manifest:algorithm">
        <ref name="algorithm-attlist"/>
        <empty/>
    </element>
</define>
```

С <manifest:algorithm> связаны следующие атрибуты:

- название алгоритма;
- вектор инициализации.

Название алгоритма

Атрибут manifest:algorithm-name определяет название алгоритма, который используется для шифрования файла, он также определяет, каким способом используется данный алгоритм. На текущий момент поддерживается только алгоритм Blowfish в режиме обратной связи шифра CFB.

```
<define name="algorithm-attlist" combine="interleave">
    <attribute name="manifest:algorithm-name">
        <data type="string"/>
    </attribute>
</define>
```

Вектор инициализации

Атрибут manifest:initialisation-vector определяет 8 байт, использующихся как вектор инициализации для шифра потока. Вектор инициализации — это 8-байтовая двоичная последовательность, закодированная в тип BASE64 (определенный в [RFC2045]) в момент записи в файл декларации.

```
<define name="algorithm-attlist" combine="interleave">
    <attribute name="manifest:initialisation-vector">
        <data type="base64Binary"/>
```

```
    </attribute>
</define>
```

17.7.6 Источник ключа

Элемент `<manifest:key-derivation>` содержит информацию, которая была использована для формирования ключа шифра файла из пароля, заданного пользователем.

```
<define name="key-derivation">
    <element name="manifest:key-derivation">
        <ref name="key-derivation-attlist"/>
        <empty/>
    </element>
</define>
```

С элементом `<manifest:key-derivation>` связаны следующие атрибуты:

- название источника ключа;
- шум;
- количество итераций.

Название источника ключа

Атрибут `manifest:key-derivation-name` определяет название алгоритма, который был использован для получения источника ключа. В настоящее время пакеты поддерживают использование только метода PBKDF2 получения ключа. Для получения более подробной информации см. [RFC2898].

```
<define name="key-derivation-attlist" combine="interleave">
    <attribute name="manifest:key-derivation-name">
        <data type="string"/>
    </attribute>
</define>
```

Шум

Атрибут `manifest:salt` определяет 16-байтовую последовательность, которая используется как «шум» (salt) алгоритмом получения ключа. Шум — это 16-байтовая двоичная последовательность, кодированная в BASE64 (определенный в [RFC2045]) до записи в файл декларации,

```
<define name="key-derivation-attlist" combine="interleave">
    <attribute name="manifest:salt">
        <data type="base64Binary"/>
    </attribute>
</define>
```

Количество итераций

Атрибут `manifest:iteration-count` определяет количество итераций, использующихся алгоритмом получения ключа.

```
<define name="key-derivation-attlist" combine="interleave">
    <attribute name="manifest:iteration-count">
            <data type="nonNegativeInteger"/>
        </attribute>
</define>
```

Пример декларации:

```
<manifest:manifest
    xmlns:manifest="urn:oasis:names:tc:opendocument:xmlns:manifest:1.0">
    <manifest:file-entry
        manifest:media-type="application/vnd.oasis.opendocument.text"
        manifest:full-path="/"/>
    <manifest:file-entry manifest:media-type="image/jpeg"
        manifest:full-path="Pictures/10000000000003200000258912EB1C3.jpg"
         manifest:size="66704">
        <manifest:encryption-data>
            <manifest:algorithm manifest:algorithm-name="Blowfish CFB"
                manifest:initialisation-vector="T+miu403484="/>
            <manifest:key-derivation manifest:key-derivation-name="PBKDF2"
                manifest:iteration-count="1024"
                manifest:salt="aNYdmqv4cObAJSJjm4RzqA=="/>
        </manifest:encryption-data>
    </manifest:file-entry>
    <manifest:file-entry
        manifest:media-type="text/xml" manifest:full-path="content.xml"
        manifest:size="3143">
        <manifest:encryption-data>
            <manifest:algorithm manifest:algorithm-name="Blowfish CFB"
                manifest:initialisation-vector="T+miu403484="/>
            <manifest:key-derivation manifest:key-derivation-name="PBKDF2"
                manifest:iteration-count="1024"
                manifest:salt="aNYdmqv4cObAJSJjm4RzqA=="/>
        </manifest:encryption-data>
    </manifest:file-entry>
    <manifest:file-entry manifest:media-type="text/xml"
        manifest:full-path="styles.xml" manifest:size="5159">
        <manifest:encryption-data>
            <manifest:algorithm manifest:algorithm-name="Blowfish CFB"
                manifest:initialisation-vector="bChL2No5I+A="/>
            <manifest:key-derivation manifest:key-derivation-name="PBKDF2"
                manifest:iteration-count="1024"
                manifest:salt="/kfasyu7X0Ae+1uopdeCtA=="/>
        </manifest:encryption-data>
    </manifest:file-entry>
    <manifest:file-entry
        manifest:media-type="text/xml" manifest:full-path="meta.xml"/>
    <manifest:file-entry
        manifest:media-type="text/xml"
        manifest:full-path="settings.xml" manifest:size="5317">
        <manifest:encryption-data>
            <manifest:algorithm manifest:algorithm-name="Blowfish CFB"
                manifest:initialisation-vector="JQxEm6rD+4c="/>
            <manifest:key-derivation manifest:key-derivation-name="PBKDF2"
                manifest:iteration-count="1024"
                manifest:salt="PlpDaxloh4KUKx+v1g4V9g=="/>
        </manifest:encryption-data>
```

```
    </manifest:file-entry>
</manifest:manifest>
```

17.7.7 Суффикс схемы Relax-NG

Суффикс для нормативной схемы декларации Relax-NG:

```
111 </grammar>
```

Приложение А. Строгая схема Relax NG

Схема Relax-NG (см. [RNG]), приведенная в данном приложении, эквивалентна схеме, описанной в разделах с 1 по 16 данной спецификации, но ограничивает содержимое элементов метаинформации и элементов свойств форматирования в атрибутах и элементах, определенных в данной спецификации (см. также раздел 1.5).

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    OASIS OpenDocument v1.0 (Second Edition)
    Спецификация комитета №1, 19 Jul 2006
    Strict Relax-NG Schema

    $Id$

    © 2002-2005 OASIS Open
    © 1999-2005 Sun Microsystems, Inc.
-->

<grammar xmlns="http://relaxng.org/ns/structure/1.0">
    <include href="OpenDocument-schema-v1.0-cs1.rng">
        <define name="office-meta-content">
            <ref name="office-meta-content-strict"/>
        </define>
        <define name="style-page-layout-properties-content">
            <ref name="style-page-layout-properties-content-strict"/>
        </define>
        <define name="style-header-footer-properties-content">
            <ref name="style-header-footer-properties-content-strict"/>
        </define>
        <define name="style-drawing-page-properties-content">
            <ref name="style-drawing-page-properties-content-strict"/>
        </define>
        <define name="style-text-properties-content">
            <ref name="style-text-properties-content-strict"/>
        </define>
        <define name="style-paragraph-properties-content">
            <ref name="style-paragraph-properties-content-strict"/>
        </define>
        <define name="style-ruby-properties-content">
            <ref name="style-ruby-properties-content-strict"/>
        </define>
        <define name="style-section-properties-content">
            <ref name="style-section-properties-content-strict"/>
        </define>
        <define name="style-list-level-properties-content">
            <ref name="style-list-level-properties-content-strict"/>
        </define>
        <define name="style-table-properties-content">
            <ref name="style-table-properties-content-strict"/>
        </define>
        <define name="style-table-column-properties-content">
            <ref name="style-table-column-properties-content-strict"/>
        </define>
```

```
48          <define name="style-table-row-properties-content">
49              <ref name="style-table-row-properties-content-strict"/>
50          </define>
51          <define name="style-table-cell-properties-content">
52              <ref name="style-table-cell-properties-content-strict"/>
53          </define>
54          <define name="style-graphic-properties-content">
55              <ref name="style-graphic-properties-content-strict"/>
56          </define>
57          <define name="style-chart-properties-content">
58              <ref name="style-properties-content"/>
59          </define>
60      </include>
61  </grammar>
```

Приложение В. Указатели

[CSS2] Bert Bos, Håkon Wium Lie, Chris Lilley, Ian Jacobs, *Cascading Style Sheets, level 2*, http://www.w3.org/TR/1998/REC-CSS2-19980512, W3C, 1998.

[CSS3Text] Michel Suignard, *CSS3 Text Module*, http://www.w3.org/TR/2003/CR-css3-text-20030514, W3C, 2003.

[DCMI] -, *Dublin Core Metadata Element Set, Version 1.1: Reference Description*, http://www.dublincore.org/documents/dces/, Dublin Core Metadata Initiative, 2003.

[DOMEvents] Philippe Le Hégaret, Tom Pixley, *Document Object Model (DOM) Level 3 Events Specification*, http://www.w3.org/TR/2003/WD-DOM-Level-3-Events-20030331, W3C, 2003.

[HTML4] Dave Raggett, Arnoud Le Hors, Ian Jacobs, *HTML 4.01 Specification*, http://www.w3.org/TR/1999/REC-html401-19991224, W3C, 1999.

[ISO/IEC Directives] ISO/IEC Directives, Part 2 *Rules for the structure and drafting of International Standards*, 2004

[JDBC] Jon Ellis, Linda Ho, Maydene Fisher, *JDBC 3.0 Specification*, http://java.sun.com/products/jdbc/, Sun Microsystems, Inc., 2001.

[MathML] David Carlisle, Patrick Ion, Robert Miner, Nico Poppelier, *Mathematical Markup Language (MathML) Version 2.0 (Second Edition)*, http://www.w3.org/TR/2003/REC-MathML2-20031021/, W3C, 2003.

[MIMETYPES] , *List of registered MIME types*, ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/, IANA, .

[OLE] Kraig Brockschmidt, Inside OLE, Microsoft Press, 1995, ISBN: 1-55615-843-2

[OOo] , *OpenOffice.org XML File Format 1.0 Technical Reference Manual*, http://xml.openoffice.org/xml_specification.pdf, Sun Microsystems, Inc., 2002.

[PNG] Thomas Boutell, *PNG (Portable Network Graphics) Specification*, http://www.w3.org/TR/REC-png-multi.html, W3C, 1996.

[RFC1766] H. Alvestrand, *Tags for the Identification of Languages*, http://www.ietf.org/rfc/rfc1766.txt, IETF, 1995.

[RFC2045] N. Freed and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, http://www.ietf.org/rfc/rfc2045.txt, IETF, 1996.

[RFC2048] N. Freed, J. Klensin, J. Postel, *Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*, http://www.ietf.org/rfc/rfc2048.txt, IETF, 1996.

[RFC2616] IETF, *Hypertext Transfer Protocol -- HTTP/1.1*, http://www.ietf.org/rfc/rfc2616.txt, IETF, 1999.

[RFC2898] B. Kaliski, *PKCS #5: Password-Based Cryptography Specification Version 2.0*, http://www.ietf.org/rfc/rfc2898, IETF, 2000.

[RFC3066] H. Alvestrand, *Tags for the Identification of Languages*, http://www.ietf.org/rfc/rfc3066.txt, IETF, 2001.

[RFC3987] M. Duerst, M. Suignard, *Internationalized Resource Identifiers (IRIs)*, http://www.ietf.org/rfc/rfc3987.txt, IETF, 2005.

[RNG] ISO/IEC 19757-2 *Document Schema Definition Language (DSDL) -- Part 2: Regular-grammar-based validation -- RELAX NG*, 2003

[RNG-Compat] James Clark, MURATA Makoto, *RELAX NG DTD Compatibility*, http://www.oasis-open.org/committees/relax-ng/compatibility-20011203.html, OASIS, 2001.

[SMIL20] W3C, *Synchronized Multimedia Integration Language 2.0 (SMIL 2.0)*, http://www.w3.org/TR/smil20/, W3C, 2001.

[SVG] Jon Ferraiolo, 藤沢 淳 (FUJISAWA Jun), Dean Jackson, *Scalable Vector Graphics (SVG) 1.1*, http://www.w3.org/TR/2003/REC-SVG11-20030114/, W3C, 2003.

[UNICODE] The Unicode Consortium. The Unicode Standard, Version 4.0.0, defined by: *The Unicode Standard, Version 4.0* (Boston, MA, Addison-Wesley, 2003. ISBN 0-321-18578-1)

[XForms] W3C, *XForms*, http://www.w3.org/TR/xforms/, W3C, 2004.

[XLink] Steve DeRose, Eve Maler, David Orchard, *XML Linking Language*, http://www.w3c.org/TR/xlink/, W3C, 2001.

[xml-names] Tim Bray, Dave Hollander, Andrew Layman, *Namespaces in XML*, http://www.w3.org/TR/REC-xml-names/, W3C, 1999.

[XML1.0] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau , *Extensible Markup Language (XML) 1.0 (Third Edition)*, http://www.w3.org/TR/2004/REC-xml-20040204, W3C, 2004.

[xmlschema-2] Paul V. Biron, Ashok Malhotra, *XML Schema Part 2: Datatypes*, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/, W3C, 2001.

[XSL] W3C, *Extensible Stylesheet Language (XSL)*, http://www.w3.org/TR/2001/REC-xsl-20011015/, W3C, 2001.

[XSLT] James Clark, *XSL Transformations (XSLT) Version 1.0*, http://www.w3.org/TR/1999/REC-xslt-19991116, W3C, 1999.

[XSLT2] Michael Kay, *XSL Transformations (XSLT) Version 2.0*, http://www.w3.org/TR/2003/WD-xslt20-20031112/, W3C, 2003.

[ZIP] *Info-ZIP Application Note 970311*, ftp://ftp.uu.net/pub/archiving/zip/doc/appnote-970311-iz.zip, 1997

Приложение С. Типы MIME и расширения имен файлов (не нормативное)

Типы MIME и расширения файлов, содержащиеся в данном разделе, применимы только для офисных документов, хранящихся в пакете (см. раздел 2.1). Для получения информации об использовании типов MIME для документов, не содержащихся в пакете, см. раздел 1.7.

Следующая таблица содержит список типов MIME и расширений файлов для документов, удовлетворяющих данной спецификации, которые на момент издания спецификации были зарегистрированы в соответствии с [RFC2048]. Для получения сведений о текущем списке зарегистрированных типов MIME см. [MIMETYPES].

Тип MIME	*Расширение*	*Описание*
На момент издания спецификации типы MIME не зарегистрированы.		

Следующая таблица содержит список типов MIME и расширений для офисных документов, соответствующих данной спецификации, и регистрация которых на момент издания спецификации ведется в соответствии с [RFC2048].

Пожалуйста, проверьте [MIMETYPES] перед использованием этих типов MIME. Если тип MIME не указан в таблице, то следует использовать тип MIME, который будет результатом добавления «x-» после символа «/» (т. е. application/x-vnd.oasis.opendocument.text).

Тип MIME	*Расширение*	*Описание*
application/vnd.oasis.opendocument.text	odt	Текстовый документ
application/ vnd.oasis.opendocument.text-template	ott	Текстовый документ, используемый в качестве шаблона
application/vnd.oasis.opendocument.graphics	odg	Графический документ (векторная графика)
application/ vnd.oasis.opendocument.graphics-template	otg	Графический документ (векторная графика), используемый в качестве шаблона
application/vnd.oasis.opendocument.presentation	odp	Документ презентации

Тип MIME	*Расширение*	*Описание*
application/ vnd.oasis.opendocument.presentation-template	otp	Документ презентации, используемый в качестве шаблона
application/vnd.oasis.opendocument.spreadsheet	ods	Электронная таблица
application/ vnd.oasis.opendocument.spreadsheet-template	ots	Электронная таблица, используемая в качестве шаблона
application/vnd.oasis.opendocument.chart	odc	Документ диаграммы
application/ vnd.oasis.opendocument.chart-template	otc	Документ диаграммы, используемый в качестве шаблона
application/vnd.oasis.opendocument.image	odi	Документ растровой графики
application/ vnd.oasis.opendocument.image-template	oti	Документ растровой графики, используемый в качестве шаблона
application/vnd.oasis.opendocument.formula	odf	Документ формулы
application/ vnd.oasis.opendocument.formula-template	otf	Документ формулы, используемый в качестве шаблона
application/vnd.oasis.opendocument.text-master	odm	Составной текстовый документ (см. раздел 2.3.1)
application/vnd.oasis.opendocument.text-web	oth	Текстовый документ, используемый в качестве шаблона для HTML документов

Приложение D. Набор основных возможностей (не нормативное)

Спецификация OpenDocument не описывает, какие элементы и атрибуты приложений, соответствующих схеме, могут или должны поддерживаться. Цель данного положения — установить, что спецификация OpenDocument может быть использована любыми приложениями, даже если они не поддерживают некоторые или большинство элементов и атрибутов, описанных в данной спецификации. Например, приложения просмотра могут не поддерживать все атрибуты и элементы, связанные с редактированием (отслеживание изменений), другие приложения могут поддерживать только атрибуты и элементы, связанные с содержимым, а не со стилями.

Даже обычные офисные приложения могут поддерживать только подмножество элементов и атрибутов, описанных в данной спецификации. Например, они могут не поддерживать списки внутри текстовых блоков, некоторые элементы и атрибуты, связанные с языком.

В следующей таблице представлен обзор элементов и атрибутов, используемых обычными офисными приложениями. Здесь перечислены разделы и подразделы данной спецификации и некоторые классы обычных офисных приложений. Символ «X» означает, что большинство (или по крайней мере значительное число) элементов и атрибутов, описанных в разделах, обычно поддерживаются данными классами приложений. Символ «(X)» означает, что только небольшое число элементов и атрибутов обычно поддерживается.

Раздел	*Название*	*Текст*	*Электронная таблица*	*Графическое изображение*	*Презентация*	*Диаграмма*	*Растровое изображение*
2.2	Метаданные документа	X	X	X	X	X	X
2.3	Элемент основной части документа и типы документов	X	X	X	X	X	X
2.4	Параметры настройки приложений	X	X	X	X	X	X
2.5	Скрипты	X	X	X	X	X	X
2.6	Объявления начертания	X	X	X	X	X	

Раздел	*Название*	*Текст*	*Электронная таблица*	*Графическое изображение*	*Презентация*	*Диаграмма*	*Растровое изображение*
	шрифтов						
2.7	Стили	X	X	X	X	X	X
2.8	Стили и макеты страницы	X	X	X	X		
3	Элементы метаданных	X	X	X	X	X	X
4.1	Абзацы и основная текстовая структура	X	X(1)	X(2)	X(2)	X(3)	
4.1	Заголовки	X					
4.2	Последовательности страниц	X					
4.3	Списки	X		X(2)	X(2)		
4.4	Текстовые разделы	X					
4.5	Графическое наполнение, привязанное к странице	X					
4.6	Отслеживание изменений	X					
4.7	Текстовые объявления	X	(X)	(X)	(X)	(X)	
5.1	Основное текстовое содержимое	X	X(1)	X(2)	X(2)	X(3)	
5.2	Закладки и перекрестные ссылки	X					
5.3	Сноски	X					
5.4	Транскрипция	X					
5.5	Аннотации к тексту	X					
5.6	Индексные метки	X					
5.7	Отслеживание изменений и метки изменений	X					
5.8	Внедренные графические объекты и текстовые блоки	X					
6	Текстовые поля	X	(X)	(X)	(X)		
7	Текстовые индексы	X					
8.1	Общая модель таблиц	X	X				
8.2	Расширенная модель таблиц	X	X				
8.3	Расширенные таблицы		X				
8.4	Расширенные ячейки таблицы		X				
8.5	Содержимое документов электронных таблиц		X				
8.6	Диапазоны баз данных		X				

Раздел	*Название*	*Текст*	*Электронная таблица*	*Графическое изображение*	*Презентация*	*Диаграмма*	*Растровое изображение*
8.7	Фильтры		X				
8.8	Сводные таблицы		X				
8.9	Консолидация		X				
8.10	DDE-ссылки таблиц		X				
8.11	Отслеживание изменений в электронных таблицах		X				
9.1	Расширенные свойства страниц для графических приложений			X	X		
9.2	Векторные фигуры	X	X	X	X		
9.3	Фреймы	X	X	X	X		X(4)
9.4	Трехмерные фигуры	X	X	X	X		
9.5	Специальные фигуры	X	X	X	X		
9.6	Фигуры презентации				X		
9.7	Анимация презентаций				X		
9.8	Анимация SMIL-презентаций				X		
9.9	События презентаций				X		
9.10	Текстовые поля презентаций				X		
9.11	Содержимое документов презентаций				X		
10	Содержимое диаграмм					X	
11	Содержимое форм	X	X	X	X		
12.1	Аннотация	X(5)	X(1)				
12.2	Формат чисел	X	X	X	X		
12.3	Метаданные отслеживания изменений	X	X				
12.4	Таблицы датчиков событий	X	X	X	X		
12.5	Математическое содержимое	X	X	X	X		
12.6	DDE-соединения	X	X				
13	SMIL-анимации				X		
14.1	Элемент стиля	X	X	X	X	X	X
14.2	Стили по умолчанию	X	X	X	X	X	X
14.3	Разметка страницы	X	X	X	X		
14.4	Мастер-страницы	X	X	X	X		
14.5	Шаблоны таблиц	X	X				
14.6	Объявление начертания	X	X	X	X	X	

Раздел	*Название*	*Текст*	*Электронная таблица*	*Графическое изображение*	*Презентация*	*Диаграмма*	*Растровое изображение*
	шрифта						
14.7	Стили данных	X	X	X	X	X	
14.8	Стили текста	X	X(6)	X(6)	X(6)	X(6)	
14.9	Расширенные стили текста	X					
14.10	Стили списков	X		X	X		
14.11	Стили структур	X					
14.12	Стили таблиц	X	X				
14.13	Графические стили	X	X	X	X		
14.14	Элементы расширенных графических стилей	X	X	X	X	X	
14.15	Разметка страницы презентации				X		
14.16	Стили диаграмм					X	
15.2	Свойства форматирования для разметки страницы	X	X	X	X		
15.3	Свойства форматирования верхнего и нижнего колонтитулов	X	(X)				
15.4	Свойства форматирования текста	X	X	X	X	X	
15.5	Свойства форматирования абзаца	X	X	X	X	X	
15.6	Свойства форматирования транскрипции	X					
15.7	Свойства форматирования разделов	X					
15.8	Свойства форматирования таблиц	(X)	X				
15.9	Свойства форматирования столбцов	(X)	X				
15.10	Свойства форматирования строк таблиц	(X)	X				
15.11	Свойства форматирования ячеек таблиц	(X)	X				
15.12	Свойства стилей уровней списков	X		X	X		

Раздел	*Название*	*Текст*	*Электронная таблица*	*Графическое изображение*	*Презентация*	*Диаграмма*	*Растровое изображение*
15.13	Свойства контурной линии	X(7)	X(7)	X	X	X	
15.14	Свойства заполнения	X(7)	X(7)	X	X	X	
15.15	Свойства анимации текста	X(7)	X(7)	X	X		
15.16	Свойства текста и выравнивания текста	X(7)	X(7)	X	X		
15.17	Свойства цвета	X(7)	X(7)	X	X		X
15.18	Свойства тени	X(7)	X(7)	X	X		
15.19	Свойства соединительных элементов	X(7)	X(7)	X	X		
15.20	Свойства измерений	X(7)	X(7)	X	X		
15.21	Свойства примечаний	X(7)	X(7)	X	X		
15.22	Свойства трехмерной геометрии	X(7)	X(7)	X	X	X	
15.23	Свойства трехмерного освещения	X(7)	X(7)	X	X	X	
15.24	Свойства трехмерного текстурирования	X(7)	X(7)	X	X	X	
15.25	Свойства трехмерных материалов	X(7)	X(7)	X	X	X	
15.26	Свойства трехмерных теней	X(7)	X(7)	X	X	X	
15.27	Свойства форматирования фреймов	X	(X)	(X)	(X)	(X)	
15.28	Свойства форматирования плавающего фрейма	X	X	X	X		
15.29	Свойства форматирования диаграмм					X	
15.30	Свойства подтипов диаграмм					X	
15.31	Свойства осей диаграмм					X	
15.32	Общие свойства диаграмм					X	
15.33	Статистические свойства					X	
15.34	Свойства области построения диаграммы					X	
15.35	Свойства кривой регрессии					X	
15.36	Атрибуты страницы презентации				X		

(1) — внутри ячеек;

(2) — внутри текстовых блоков;

(3) — внутри некоторых объектов диаграмм;

(4) — только фреймы, содержащие изображения;

(5) — внутри текста;

(6) — только автоматические стили;

(7) — только для векторных объектов.

Приложение Е. Изменения по сравнению с предыдущими версиями (не нормативное)

Е.1. Изменения, внесенные в «Open Office Specification 1.0 Committee Draft 1»

Ниже перечислены изменения, внесенные в «Open Office Specification 1.0 Committee Draft 1».

- Название данной спецификации изменилось на «Open Document Format for Office Applications (OpenDocument) 1.0».
- Пространства имен URI (раздел 1.3) были адаптированы к новому названию спецификации.
- Рекомендации по использованию MIME-типов были перенесены в необязательное Приложение (Приложение С) и были адаптированы к новому названию спецификации.
- Добавлены новые определения. Среди них:
  — специальные фигуры (раздел 9.5);
  — SMIL анимация (разделы 9.8 и 13);
  — поддержка XForms (раздел 11.2).
- В схеме и в описаниях были исправлены различные ошибки.
- Некоторые описания были переписаны заново для более легкого восприятия.

Е.2. Изменения, внесенные в «Open Document Format for Office Applications (OpenDocument) 1.0 Committee Draft 2»

Ниже перечислены изменения, внесенные в «Open Document Format for Office Applications (OpenDocument) 1.0 Committee Draft 2».

- Для совместимых элементов и атрибутов были добавлены следующие пространства имен:
  — urn:oasis:names:tc:opendocument:xmlns:xsl-fo-compatible:1.0 для атрибутов, совместимых с [XSL];

— urn:oasis:names:tc:opendocument:xmlns:svg-compatible:1.0 для элементов и атрибутов, совместимых с [SVG];
— urn:oasis:names:tc:opendocument:xmlns:smil-compatible:1.0 для атрибутов, совместимых с [SMIL20].

- Следующие пространства имен больше не поддерживаются:
  — http://www.w3.org/1999/XSL/Format
  — http://www.w3.org/2000/svg
  — http://www.w3.org/2001/SMIL20/
- Атрибут `xforms:submission`, описанный в разделе 11.3.11, был переименован в `form:xforms-submission`.
- Информация для специальных фигур (раздел 9.5), SMIL-анимации (разделы 9.8 и 13) и текстовые поля презентации (раздел 9.10) была добавлена к свойствам ядра, описанным в таблице Приложения D; номера разделов в этой таблице были изменены.

E.3. Изменения, внесенные в «Open Document Format for Office Applications (OpenDocument) v1.0»

Ниже перечислены различия между спецификацией «Open Document Format for Office Applications (OpenDocument) v1.0» и спецификацией «Open Document Format for Office Applications (OpenDocument) v1.0 (Second Edition)».

- Использование ключевых слов должно «shall», может «may», и т.д. соответствует приложению H требований ISO.
- Двусмысленные ссылки приведены в точное соответствие номерам разделов и подразделов.
- Исправлены грамматические и синтаксические ошибки.
- Термины «unicode» и «UNICODE» заменены на библиографическую ссылку «[UNICODE]». Библиографическая ссылка добавлена в приложение B.
- Все упоминания термина URI, за исключением одного в приложении E.1, были заменены термином IRI, потому что схема данных W3C «anyURI», используемая в OpenDocument, фактически принимает IRIs а не URIs. Ссылки на RFC2396 были заменены ссылками на [RFC3987]. В

приложении В библиографическая запись для RFC2396 была заменена на [RFC3987].

- Ссылка на RELAX NG DTD Compatibility specification была добавлена во второй параграф раздела 1.4. Библиографическая запись для RELAX NG DTD Compatibility specification была добавлена в приложение В.
- Ссылка [RFC2045] была добавлена в некоторые употребления термина “BASE64”, и употребление «base64» было скорректировано на «BASE64». Библиографическая запись RFC2045 была добавлена в приложение В.
- Описание атрибута `draw:z-index` в разделе 9.2.5 было исправлено.
- Ссылка на W3C CSS3 Text Module уточнена в приложении В. URL “http://www.w3.org/TR/2003/CR-css3-text-20030514” был добавлен в библиографическую запись CSS3Text.
- В приложении В библиографическая запись [RNG] ссылается на ISO Relax-NG specification document вместо OASIS Relax-NG specification document.
- В приложении В изменена библиографическая запись ZIP.
- Список ассистентов был перемещен с титульного листа в приложение.

Приложение F. Список ассистентов

Действующие ассистенты:

Daniel Brotsky, Adobe Systems
Jerome Dumonteil, Ars Aperta
Charles Schulz, Ars Aperta
Jerry Berrier, BayState Council of the Blind (BSCB)
Donglin Wang, Changfeng Open Standards Platform Software Alliance
Rui Zhao, Changfeng Open Standards Platform Software Alliance
Stephen Noble, Design Science, Inc.
John Madden, Duke University
Chieko Asakawa, IBM
Nathaniel Borenstein, IBM
Yue Ma, IBM
Richard Schwerdtfeger, IBM
Robert Weir, IBM
John Barstow, Individual
Patrick Durusau, Individual
Michael Paciello, Individual
Janina Sajka, Individual
David Clark, Institute for Community Inclusion
Waldo Bastian, Intel Corporation
James Mason, ISO/IEC JTC1/SC34
David Faure, KDE e.V
Jody Goldberg, Novell
David Pawson, Royal National Institute for the Blind
Michael Brauer, Sun Microsystems, Inc.
Peter Korn, Sun Microsystems, Inc.
Lars Oppermann, Sun Microsystems, Inc.
Eike Rathke, Sun Microsystems, Inc.
Florian Reuter, Sun Microsystems, Inc.
Malte Timmermann, Sun Microsystems, Inc.
Daniel Bricklin, The OpenDocument Foundation, Inc.
Daniel Carrera, The OpenDocument Foundation, Inc.
Bruce D'Arcus, The OpenDocument Foundation, Inc.
Gary Edwards, The OpenDocument Foundation, Inc.
Richard Kernick, The OpenDocument Foundation, Inc.
Tomas Mecir, The OpenDocument Foundation, Inc.
Thomas Metcalf, The OpenDocument Foundation, Inc.
David A. Wheeler, The OpenDocument Foundation, Inc.
Chris Nokleberg, Tonic Systems, Inc.

Предыдущие консультанты:

Paul Grosso, Arbortext
Tom Magliery, Blast Radius
Doug Alberg, Boeing
Paul Langille, Corel
John Chelsom, CSW Informatics
Monica Martin, Drake Certivo
Jason Harrop, Individual
Uche Ogbuji, Individual
Lauren Wood, Individual

Simon Davis, National Archive of Australia
Mark Heller, New York State Office of the Attorney General
Phil Boutros, Stellent
Daniel Vogelheim, Sun Microsystems, Inc.

УДК 004.057.3 ОКС 35.240.30

Ключевые слова: открытый формат файлов для офисных приложений, XML-схема для офисных приложений, текстовые документы, электронные таблицы

**Издано на основе электронной версии,
изготовленной ассоциацией "ЕВРААС" в формате PDF и подтвержденной
электронно-цифровой подписью Удостоверяющего центра Росстандарта**

Усл. печ. л. 103,69. Уч.-изд. л. 88,13.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
www.gostinfo.ru info@gostinfo.ru