МЕЖГОСУДАРСТВЕННЫЙ СОВЕТ ПО СТАНДАРТИЗАЦИИ, МЕТРОЛОГИИ И СЕРТИФИКАЦИИ
(МГС)
INTERSTATE COUNCIL FOR STANDARDIZATION, METROLOGY AND CERTIFICATION
(ISC)

МЕЖГОСУДАРСТВЕННЫЙ СТАНДАРТ

ГОСТ 33246— 2015 (ISO/IEC 12785-1: 2009)

Информационные технологии

ОБУЧЕНИЕ, ОБРАЗОВАНИЕ И ПОДГОТОВКА. УПАКОВКА КОНТЕНТА

Часть 1

Информационная модель

(ISO/IEC 12785-1:2009, MOD)

Издание официальное

Москва
Стандартинформ
2019

gost.e-ecolog.ru

Предисловие

Цели, основные принципы и основной порядок проведения работ по межгосударственной стандартизации установлены в ГОСТ 1.0—2015 «Межгосударственная система стандартизации. Основные положения» и ГОСТ 1.2—2015 «Межгосударственная система стандартизации. Стандарты межгосударственные, правила и рекомендации по межгосударственной стандартизации. Правила разработки, принятия, обновления и отмены»

Сведения о стандарте

1 ПОДГОТОВЛЕН Федеральным государственным бюджетным образовательным учреждением высшего профессионального образования «Московский государственный технологический университет «СТАНКИН» на основе собственного перевода на русский язык англоязычной версии стандарта, указанного в пункте 5

2 ВНЕСЕН Федеральным агентством по техническому регулированию и метрологии

3 ПРИНЯТ Межгосударственным советом по стандартизации, метрологии и сертификации (протокол от 18 июня 2015 г. № 47)

За принятие проголосовали:

Краткое наименование страны по МК (ИСО 3166) 004—97	Код страны по МК (ИСО 3166) 004—97	Сокращенное наименование национального органа по стандартизации
Армения	AM	Минэкономики Республики Армения
Киргизия	KG	Кыргызстандарт
Россия	RU	Росстандарт
Узбекистан	UZ	Узстандарт

4 Приказом Федерального агентства по техническому регулированию и метрологии от 17 ноября 2015 г. № 1835-ст межгосударственный стандарт ГОСТ 33246—2015 (ISO/IEC 12785-1:2009) введен в действие в качестве национального стандарта Российской Федерации с 1 ноября 2016 г.

5 Настоящий стандарт является модифицированным по отношению к международному стандарту ISO/IEC 12785-1:2009 «Информационные технологии. Обучение, образование и подготовка. Упаковка контента. Часть 1. Информационная модель» («Information technology — Learning, education and training — Content packaging — Part 1: Information model», MOD) путем исключения справочных приложений C и D.

Сопоставление структуры настоящего стандарта со структурой примененного в нем международного стандарта приведено в дополнительном приложении ДА

6 ВВЕДЕН ВПЕРВЫЕ

7 ПЕРЕИЗДАНИЕ. Январь 2019 г.

*Информация об изменениях к настоящему стандарту публикуется в ежегодном информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячном 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 (*gost.e-ecolog.ru

© ISO, 2009 — Все права сохраняются
© Стандартинформ, оформление, 2016, 2019

В Российской Федерации настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины и определения . . . 1
4 Сокращения . . . 5
5 Концептуальная модель упаковки контента (CPCM) . . . 5
6 Описание классов и требований к взаимосвязи . . . 6
6.1 Основные термины и понятия . . . 6
6.2 Платформонезависимая модель информационной модели упаковки контента . . . 8
6.3 Класс InterchangePackage . . . 8
6.4 Семейство классов Manifest . . . 10
6.5 Семейство классов Organizations . . . 14
6.6 Семейство классов Resources . . . 19
6.7 Класс Metadata . . . 22
6.8 Класс Variant . . . 23
6.9 Класс IPointer . . . 25
6.10 Класс Extension . . . 26
6.11 Классы характеристик . . . 28
7 Соответствия . . . 37
7.1 Формулировки . . . 37
7.2 Экземпляры пакетов обмена . . . 37
7.3 Считыватели пакетов . . . 37
7.4 Записыватели пакетов . . . 37
7.5 Расширения (расширение соответствий) . . . 37
Приложение А (справочное) Первоисточник стандарта . . . 39
Приложение В (справочное) Подтверждение прав интеллектуальной собственности . . . 40
Приложение ДА (справочное) Сопоставление структуры настоящего стандарта со структурой примененного в нем международного стандарта . . . 41
Библиография . . . 42

Введение

ИСО (Международная организация по стандартизации) и МЭК (Международная электротехническая комиссия) являются частью специализированной системы всемирной стандартизации. Национальные организации, которые являются участниками ИСО или МЭК, принимают участие в разработке международных стандартов посредством технических комитетов, основанных соответствующими организациями, для работы с отдельными отраслями технической деятельности. Сотрудничество технических комитетов лежит в сфере общих интересов. Другие международные организации, как государственные, так и коммерческие, поддерживают связь с ИСО и МЭК и также участвуют в их работе. В сфере информационных технологий ИСО и МЭК создали объединенный технический комитет — ИСО/МЭК СТК 1.

Международные стандарты разрабатываются в соответствии с правилами, описанными в директивах ИСО/МЭК, часть 2.

Главная задача объединенного технического комитета — подготовка международных стандартов. Предварительные проекты международных стандартов, утвержденные объединенным техническим комитетом, передаются в государственные организации для голосования. Для выхода международного стандарта требуется как минимум 75 % голосов организаций, участвующих в голосовании.

При исключительных обстоятельствах объединенный технический комитет может предложить публикацию технического отчета об одном из следующих типов:

- тип 1, когда необходимая поддержка не может быть получена для публикации международного стандарта, несмотря на повторные усилия;
- тип 2, когда предмет все еще является объектом технического развития или где по любой другой причине есть будущее, но не непосредственная возможность соглашения по международному стандарту;
- тип 3, когда объединенный технический комитет собрал данные различного вида от того, что обычно издано как международный стандарт («состояние», например).

Технические отчеты о типах 1 и 2 подвергаются рассмотрению в течение трех лет после публикации, чтобы решить, могут ли они быть преобразованы в международные стандарты. Технические отчеты о типе 3 не обязательно должны быть рассмотрены, пока данные, которые они обеспечивают, как полагают, больше не действительны или не полезны.

Особое внимание уделяется ситуации, когда некоторые части документа могут быть субъектом патентного права. ИСО и МЭК не несут ответственности за идентификацию некоторых или всех патентных прав.

Настоящий стандарт разработан на основе спецификации IMS Global Learning Consortium (IMS GLC) (Упаковка контента версии 1.2) [1]—[4]. Спецификация IMS Content Packaging широко используется по всему миру для поддержки технологии обучения. IMS Content Packaging является неотъемлемой основой SCORM [5] от его зарождения до текущей версии. Но самое главное, IMS Content Packaging также широко используется за пределами SCORM на самостоятельной основе. IMS Content Packaging также используется во многих других высокоуровневых учебных целях, например для архивирования MIT OpenCourseWare, распространения контента данных, которые не содержат исполняемых модулей и метаданных для Федерации обучения Австралии, и предоставления общенациональных услуг электронного обучения для Кибернетической системы домашнего образования в Корее. IMS GLC предоставляет постоянную помощь для использования, профилирования и совершенствования стандарта упаковки контента по адресу: gost.e-ecolog.ru

Стандарт ИСО/МЭК 12785 «Информационные технологии. Обучение, образование и подготовка. Упаковка контента» состоит из следующих частей:

- Часть 1. Информационная модель;
- Часть 2. XML-привязка;
- Часть 3. Лучшие практики и руководство по применению.

Информационная модель упаковки контента IMS, являющаяся источником и базовой спецификацией данного стандарта, описывает структуры данных, которые могут быть использованы для обмена данными между системами, которые хотели бы импортировать, экспортировать, объединять и разъединять пакеты образовательного контента.

Спецификация упаковки контента IMS была первоначально задумана для упаковки образовательного контента.

Спецификация поддерживает описание контента, связанного с заданной образовательной деятельностью, размещение контента и как эти части контента могут быть сформированы для лучшего образовательного эффекта. Как результат широкого внедрения спецификации миллионы IMS

пакетов контента образовательного содержания используются в различных программных приложениях. IMS упаковка контента является основной спецификацией, указанной в эталонной модели контента для совместного пользования (SCORM) инициативной группы ADL. Внедряющие IMS упаковку контента распространили ее использование не только для упаковки учебного контента. На IMS упаковку контента теперь ссылаются другие спецификации IMS для упаковки и обмена другими типами данных.

Запросы на основные функциональные дополнения не были включены в серию IMS упаковки контента версии 1.1.x и накапливались как практика, созревшая вокруг реализации IMS упаковки контента. Оценка этих запросов в 2006 г. в сочетании с обратной связью от более широкого сообщества пользователей указали IMS, что это правильное время, чтобы сделать совместно с ИСО/МЭК СТК1/ПК36 существенное обновление и окончательный релиз этой спецификации.

Новая функциональность и разъяснения, содержащиеся в настоящей спецификации упаковки контента:

a) уточнены значения терминов, используемых в спецификации;

b) уточнено и усовершенствовано использование (суб)манифеста, теперь называемого дочерний манифест:

- уточнена интерпретация объекта, указывающего на дочерний манифест;
- добавлена новая функциональность, позволяющая точно ссылаться и интерпретировать компоненты дочернего манифеста;
- добавлена поддержка внешних дочерних манифестов;

c) добавлена поддержка внешних справочных файлов метаданных;

d) все внутренние словари удалены и в настоящее время поддерживаются за счет процесса регистрации IMS-словарей (см. gost.e-ecolog.ru);

e) добавлен новый тип ресурса «stand alone resource», позволяющий использовать другие пакеты, как часть содержимого LET;

f) уточнены синтаксис и использование классов информационной модели Base, Parameter, IsVisible и Href;

g) добавлена поддержка вариантов ресурсов. Это включает в себя поддержку альтернативных ресурсов для доступности LET-контента;

h) добавлена поддержка названия организации и пункта на нескольких языках;

i) уточнена поддержка для обмена пакетами, которые содержат только контент, и для обмена пакетами, которые не имеют локальных файлов содержания.

Этот документ возникает в среде активной реализации постоянно растущего внедрения IMS упаковки контента. Основной целью этого документа является обеспечение дальнейшего роста, упорядочивая текущую практику. В связи с этим следующее определение обратной совместимости вело развитие информационной модели упаковки контента:

a) с точки зрения информационной модели IMS упаковки контента информационная модель IMS упаковки контента версии 1.1.4 [6], [7] есть собственное подмножество этой информационной модели IMS упаковки контента;

b) семантика компонентов информационной модели упаковки контента сохраняется между версиями за исключением случаев, когда необходимо обеспечить неоднозначность.

Структура остальной части настоящего стандарта:

1 Область применения	Определяет область применения настоящего стандарта
2 Нормативные ссылки	Перечень ссылок на другие стандарты и RFC, приведенные в настоящем стандарте
3 Термины и определения	Определяет термины и определения
4 Сокращения	Раскрывает сокращения, используемые в настоящем стандарте
5 Концептуальная модель упаковки контента	Иллюстрирует концептуальную структуру информационной модели упаковки контента
6 Описание классов и требований к взаимосвязи	Охватывает структурные отношения, тип данных, пространство значений и число допускаемых вхождений для каждого вида информационного объекта
7 Соответствие	Набор заявлений соответствия, которые необходимо соблюдать для соответствия экземпляров систем и документов

ISO/IEC 12785 был принят объединенным техническим комитетом ИСО/МЭК СТК 1 «Информационные технологии», подкомитетом ПК 36 «Информационная технология для обучения, образования и подготовки».

От Российской Федерации функции постоянно действующего национального рабочего органа ИСО/МЭК СТК 1 ПК 36 выполняет ТК 461 «Информационно-коммуникационные технологии в образовании» (ИКТО), активно участвующий в разработке международных стандартов, осуществляющий разработку комплекса национальных стандартов ИКТО.

М Е Ж Г О С У Д А Р С Т В Е Н Н Ы Й С Т А Н Д А Р Т

Информационные технологии

ОБУЧЕНИЕ, ОБРАЗОВАНИЕ И ПОДГОТОВКА. УПАКОВКА КОНТЕНТА

Часть 1

Информационная модель

Information technology. Learning, education, and training. Content packaging. Part 1. Information model

Дата введения — 2016—11—01

1 Область применения

Областью применения настоящего стандарта является описание структур данных, которые могут быть использованы для обмена образовательным контентом между системами, которые могут импортировать, экспортировать, объединять и разъединять пакеты образовательного контента.

Информационная модель упаковки контента сохраняет IMS информационную модель упаковки контента версии 1.2 в качестве базовой модели и предоставляет новые функциональные возможности путем расширения этой базовой модели.

Эта спецификация заменяет IMS информационную модель упаковки контента версии 1.1.4 и отменяет все предыдущие описания и определения IMS информационной модели упаковки контента.

2 Нормативные ссылки

В настоящем стандарте использована нормативная ссылка на следующий межгосударственный стандарт:

ГОСТ 7.75—97 Система стандартов по информации, библиотечному и издательскому делу. Коды наименований языков

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если ссылочный стандарт заменен (изменен), то при пользовании настоящим стандартом следует руководствоваться заменяющим (измененным) стандартом. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

3 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями:

3.1 **дочерний манифест:** Полный, подчиненный манифест, содержащийся в родительском манифесте или на который имеется ссылка из родительского манифеста. child manifest

Примечания

1 Манифест может содержать несколько дочерних манифестов (по IMS Content Packaging версии 1.2).

2 Дочерний манифест может быть внешним по отношению к пакету обмена только по ссылке из родительского, а не содержаться в нем.

Издание официальное

3 Дочерний манифест описывает полный логический пакет, который является частью более крупного логического пакета, определенного родительским манифестом.

4 Дочерний манифест может быть локальным или удаленным.

См. также: пакет обмена, местный, логический пакет, манифест, удаленный.

3.2 **файл контента:** Набор файлов, содержащий по крайней мере один файл манифеста и соответствующий требованиям настоящего стандарта. content file

Примечание — Файл контента может быть локальным или удаленным.

См. также: местные, логические пакет, удаленный.

3.3 **организация:** Логические отношения, такие как иерархическое дерево из единиц контента. organization

Примечание — В манифесте может быть описано более одной логической организации.

См. также: ресурс.

3.4 **управляющий файл:** Единичный компьютерный файл, который определяет привязку информационной модель упаковки контента (CPIM), чтобы сделать ее пригодной для машинной обработки. control file

Примечание — Программный компонент может ссылаться на управляющий файл при оценке достоверности связанного экземпляра информационной модели или для руководства созданием связанного экземпляра информационной модели.

Пример — Файл, содержащий XML-схему, может быть использован в качестве управляющего файла для XML-привязки манифеста.

3.5 **пакет обмена:** Набор полезного (многоразового) образовательного контента, которым обмениваются между информационными системами для обучения, образования и подготовки. interchange package

Примечание — Пакет обмена может быть создан в одном сжатом бинарном файле (файл пакета обмена) или в виде набора файлов на портативных носителях (например, CD, DVD, флеш-память).

См. также: логический пакет, файл пакета обмена.

3.6 **исполняемый URI:** Представление унифицированного локатора ресурса (URL), который может быть включен в описание ресурса и используется, чтобы найти и получить доступ к контенту, описываемому ресурсом. launchable URI

Примечание — Исполняемый унифицированный идентификатор ресурса (URI) не предназначен для восстановления считывателем пакета.

См. также: пакет обмена, считыватель пакета.

3.7 **контент:** Отдельный файл или несколько файлов, используемые в обучении, образовании и подготовке. content

Примечания

1 Логический объект полезной (и многоразового использования) информации может быть описан логическим пакетом.

2 Логический пакет может содержать один или более объектов контента.

См. также: логический пакет.

3.8 **локальный:** Компонент логического пакета, который содержится в пакете обмена. local

См. также: логический пакет, пакет обмена.

3.9 **логический пакет:** Представление одного или нескольких объектов полезного (и многоразового использования) образовательного контента.

Примечание — Логический пакет включает в себя полный набор компонентов, описанных манифестом и дочерним манифестом, включая локальные компоненты и внешние компоненты, включенные по ссылкам.

См. также: дочерний манифест, локальный, манифест, внешний.

3.10 **манифест:** Описание файлов и логических отношений между ними, которые содержатся или упоминаются в пакете контента. manifest

См. также: локальный, логический пакет, удаленный.

3.11 **манифест документа:** Манифест с контентом, который структурирован в соответствии с различными связанными технологиями. manifest document

3.12 **метаданные (по упаковке контента):** Описательная информация упаковки контента о логических пакетах, логических организациях, содержании и файлах. metadata (in content packaging)

Примечания

1 Метаданные могут быть назначены любому компоненту логического пакета, включая манифест.

2 Допускается любая привязка объектов метаданных. Каждый объект метаданных может быть локальным или удаленным.

См. также: локальный, логический пакет, удаленный.

3.13 **пространство имен:** Пространство имен XML, определенное URI-ссылкой. namespace

Примечание — Пространство имен в упаковке контента соответствует рекомендации W3C по пространству имен в XML 1.0 (Второе издание) [8], [9].

3.14 **пакет:** Объект полезного (и многоразового использования) контента. package

Примечания

1 Пакет может быть частью учебного курса, который имеет учебные релевантности вне содержания образовательной агрегации и может быть доставлен независимо как весь курс обучения или в виде набора учебных курсов.

2 Пакет должен быть автономен, то есть он должен содержать всю информацию, необходимую, чтобы использовать содержимое для обучения, образования и подготовки после его распаковки.

См. также: обмен пакетами, логический пакет.

3.15 **файл пакета обмена;** ФПО: Экземпляр пакета обмена, который физически инкапсулирован как сжатый двоичный файл, соответствующий RFC1951 (1996) [10]. package interchange file (PIF)

Пример — Пакет обмена может быть создан в виде набора файлов на съемных носителях, например CD, DVD, карты памяти USB, или в сжатом виде с помощью форматов, таких как .zip, .tar, .jar, .cab.

Примечания

1 Пакет обмена может быть создан в формате, отличном от файла пакета обмена (ФПО).

2 Обычно представление (привязка) выражается в XML.

См. также: пакет обмена.

3.16 **считыватель пакета:** Программное обеспечение, которое обрабатывает пакет обмена путем проверки объявлений в манифесте на соответствие содержанию и организации. package reader

Примечания

1 Считыватель пакетов обрабатывает как логический, так и физический пакеты.

2 Термин «обрабатывать» может включать в себя получение и сохранение информации, на которую ссылается манифест, декомпрессию или распаковку локальных файлов из PIF, извлечение или протоколирование адресов удаленных файлов.

См. также: пакет обмена, логический пакет.

3.17 **записыватель пакета:** Программное обеспечение, которое создает или изменяет экземпляр пакета обмена и собирает файл(ы) контента и другие файлы, заявленные локально, в пакете обмена и записывает их в указанную привязку пакета обмена или делегирует эту задачу другому типизированному программному процессу. package writer

См. также: пакет обмена.

3.18 **ссылочный манифест:** Манифест или компонент манифеста, на который есть ссылка из другого манифеста. referenced manifest

Примечания

1 Манифест может содержать ссылку на структуру в другом манифесте.

2 Манифест, содержащий компонент, на который имеется ссылка, называется манифест-ссылка.

3 Манифест может содержать ссылки на локальные или удаленные компоненты.

См. также: локальный, удаленный.

3.19 **относительная ссылка:** Выражение URI-ссылки, относящееся к пространству имен другого иерархического URI [IETF RFC 3986 (2005)] [11]. relative reference

Пример — relative/path/to/resource.txt является относительной ссылкой, которая интерпретируется с точки зрения контекста (алгоритм для разрешения относительных ссылок с точки зрения контекста определен в подпункте 5 RFC 3986:2005).

Примечание — Для создания целевой URI объединяются расширение и контекст.

3.20 **удаленный:** Компонент логического пакета, который находится вне пакета обмен. remote

См. также: обмен пакета, логический пакет.

3.21 **ресурс (в упаковке контента):** Один URL-адрес точки входа и ноль или более ссылок на файлы, которые необходимо выполнить до запуска содержимого. resource (in content packaging)

Примечание — Файлы, описываемые ресурсом, могут быть локальными или удаленными.

См. также: исполняемый URI, локальный, удаленный.

3.22 **автономный ресурс:** Ресурс, который позволяет манифесту объявить связь с другим манифестом так, что связанные манифесты обрабатываются как отдельные, но взаимосвязанные наборы данных. stand alone resource

Примечания

1 Связанные манифесты могут содержаться в одном пакете контента или быть доступны как URI адресуемый внешний ресурс.

2 Каждый манифест представляет собой автономный учебный ресурс, который может быть объединен с другим учебным ресурсом для создания произвольного богатого опыта обучения.

3.23 **универсальный идентификатор ресурса;** URI: Компактная последовательность символов, которая определяет абстрактный или физический ресурс (IETF RFC 3986). uniform resource identifier (URI)

3.24 **унифицированный указатель ресурса;** URL: Указатель, как подмножество URI, обеспечивает средства размещения ресурса путем описания основных механизмов доступа (IETF RFC 3986). uniform resource locator (URL)

3.25 **вариант:** Контейнер для ссылки и описания частично отличающегося образовательного контента. variant

Пример — Примеры включают, но не ограничиваются ими, языковые варианты, визуальные или звуковые варианты, варианты восстановления и варианты платформы доставки.

Примечания

1 Конкретный ресурс может иметь варианты различных форматов и различного предназначения.

2 Список вариантов в пределах ресурса определяет альтернативные наборы файлов для ресурса.

3 Метаданные используются для описания предполагаемого использования оригинального ресурса и предполагаемого использования вариантов.

См. также: метаданные ресурса.

3.26 **единица контента:** Файл или группа файлов, которые могут быть представлены в манифесте. unit of content

4 Сокращения

В настоящем стандарте применены следующие сокращения:

ABNF — расширенная форма Бэкуса — Наура (Augmented Backus-Naur Form);
ADL — перспективное распределенное обучение (Advanced Distributed Learning);
CPCM — концептуальная модель упаковки контента (Content Packaging Conceptual Model);
CPIM — информационная модель упаковки контента (Content Packaging Information Model);
IETF — рабочая группа инженеров Интернета (Internet Engineering Task Force);
LET — обучение, образование и подготовка (Learning, Education and Training);
MPEG — экспертная группа по вопросам движущегося изображения (Moving Picture Experts Group);
MPEG-21 — ISO/IEC 21000 (все части);
PIF — файл пакета обмена (ФПО, Package Interchange File);
PIM — платформонезависимая модель (Platform Independent Model);
RFC — запрос на комментарии (Request for Comments);
SCORM — эталонная модель контента для совместного пользования (Sharable Content Object Reference Model);
UML — унифицированный язык моделирования (Unified Modeling Language);
URI — универсальный идентификатор ресурса (Uniform Resource Identifier, IETF RFC 3986);
URL — универсальный указатель ресурса (Uniform Resource Locator, IETF RFC 3986);
URN — унифицированное имя ресурса (Uniform Resource Name, IETF RFC 3986);
W3C — Консорциум World Wide Web;
XML — расширяемый язык разметки (Extensible Markup Language, W3C XML).

5 Концептуальная модель упаковки контента (CPCM)

Пакет контента представляет собой единицу полезного (и многоразового) образовательного контента, как это определено в информационной модели упаковки контента. Содержание пакета состоит из логического описания пакета (манифеста) и физических ресурсов.

На рисунке 1 приведена концептуальная схема, иллюстрирующая структуру информационной модели упаковки контента (CPIM).

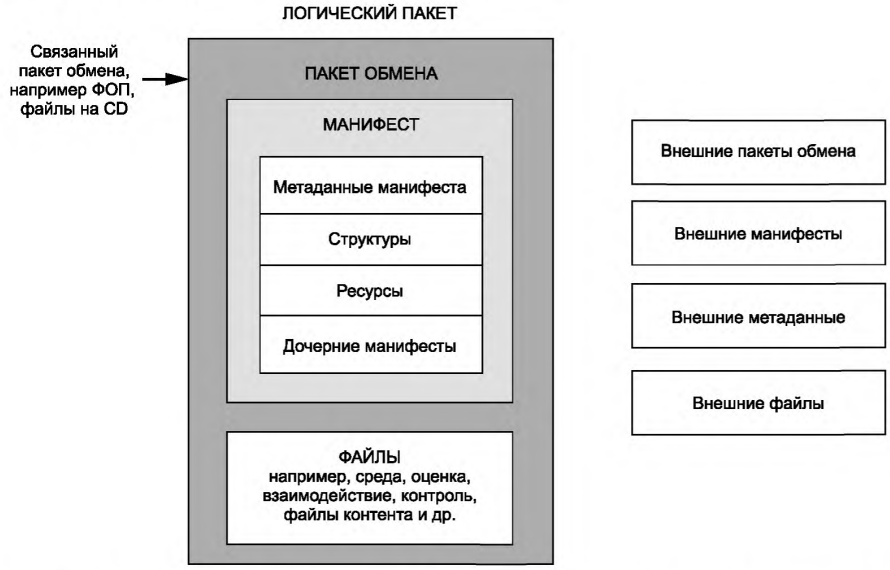


Рисунок 1 — Схематическое представление концептуальной модели упаковки контента (CPCM)

Основные структурные компоненты:

- логический пакет — представление одной или более единиц полезного (и многоразового) образовательного контента. Логический пакет включает в себя полный набор компонентов, описываемых манифестом, в том числе локальные компоненты и удаленные компоненты, включенные по ссылке;
- пакет обмена — набор связанных компонентов образовательного контента, которые должны обмениваться между системами, включая манифест и другие выбранные файлы;
- манифест — компонент, который описывает полный экземпляр логического пакета. Манифест может содержать ссылки на локальные или удаленные компоненты;
- структура — логические связи между единицами образовательного контента. Могут быть описаны несколько логических структур;
- ресурсы — описание образовательного контента и файлов ресурсов, используемых в логическом пакете. Файлы могут быть локальными или удаленными;
- дочерние манифесты — полные и подчиненные манифесты, которые содержатся внутри или на которые ссылаются из другого манифеста. Каждый описывает полный логический пакет, который является частью более крупного логического пакета. Дочерние манифесты могут быть локальными или удаленными;
- файлы — компьютерные файлы, которые содержат образовательный контент, описываемый логическим пакетом, или руководят привязкой других файлов, чтобы сделать их пригодными для машинной обработки. Содержание файлов может быть локальным или удаленным;
- метаданные (по упаковке контента) — описательная информация о пакетах контента, логических структурах, контенте или файлах. В этой схеме блок метаданных представляет набор локальных и/или удаленных объектов метаданных, которые содержатся в логическом пакете. Допускается любая привязка объектов метаданных. Метаданные могут быть определены для любой из основных структур в логическом пакете, включая манифест.

Информационная модель определяет эти основные структурные компоненты для описания и организации LET-содержания для обмена. Важным является обоснование поддержки расширяемости. Пользователи настоящего стандарта могут применять эти механизмы расширения для определения новых терминов словаря и структуры.

6 Описание классов и требований к взаимосвязи

Информативный обзор всей информационной модели упаковки контента (CPIM) предоставляется в виде платформонезависимой модели (PIM), выраженной в конструкциях UML. Все UML-диаграммы, выраженные как «платформонезависимая модель», не являются нормативными. Нормативные таблицы, определяющие классы в этой информационной модели, следуют информативным диаграммам UML.

Полное определение профиля UML и термины, используемые в нормативных табличных описаниях в этом документе для описания PIM, можно найти в руководстве IMS Profile UML [12].

В таблицах настоящего раздела последовательность символов «N/A» используется для обозначения значения «не применяются». Любые поля, отмеченные таким образом, не относятся к определяемому классу. Функции, имеющие такое значение, должны игнорироваться при привязке класса, определяемого этой информационной моделью.

Расширенная форма Бэкуса — Наура (ABNF) используется для определения определенных правил в таблицах. Расширенная форма Бэкуса — Наура определена в [13].

6.1 Основные термины и понятия

В информационной модели упаковки контента используются классы четырех абстрактных типов. Четыре типа абстрактных классов:

- класс «container»: класс «container» может быть родителем одного или нескольких дочерних классов;
- класс «value»: класс «value» не может быть родителем. То есть он не должен объединять классы типа «characteristic», «container», «value» или «неопределенность». Класс «значение» всегда должен быть дочерним класса «container» и иметь семантические значения в рамках семантических значений своего родительского класса;
- класс «characteristic»: класс «characteristic» не должен быть родительским. Класс «characteristic» должен декларировать особенности или значения, которые являются неотъемлемой функцией или

частью класса «container». Класс «characteristic» тесно связан с классом «container», который он изменяет, или с одним из аспектов, который он описывает;

- класс «unspecified»: класс «unspecified» может быть родительским классом. Класс «unspecified» служит точкой расширения для этой информационной модели.

В случае, если число элементов больше одного, важность упорядочения равнозначных элементов указывается путем добавления «упорядочено» или «не упорядочено». «Упорядочено» определяет последовательность равнозначных элементов как список, «не упорядочено» определяет набор или портфель равнозначных элементов. Порядок данных не важен.

В таблице 1 перечислены классы дескрипторов, используемых для описания абстрактных классов и определения дескрипторов.

Таблица 1 — Классы дескрипторов

Дескриптор	Определение
Class name (Имя класса)	Описывается имя, данное классу
Class type (Тип класса)	Тип абстрактного класса для этого класса
Data type (Тип данных)	Разрешенная структура допустимых значений класса для значений и характеристик классов. Допустимые типы данных: - URI: любой синтаксически правильный экземпляр URI, соответствующий RFC 3986. Примечание — Многие основополагающие спецификации, стандарты и рекомендации, упоминаемые в информационной модели, используют RFC 3986 и RFC 2732 [14] для определения URI. Это отменено RFC 3986, но многие основополагающие документы не были обновлены, чтобы ссылаться на RFC 3986; - LUID: идентификатор, локально уникальный в пределах манифеста. Это будет основано на типе данных String, который имеет ограниченную область значений; - LUIDref: ссылка на LUID, который был определен в другом месте в пределах манифеста. Значения LUID и LUIDref, которые на его ссылаются, должны быть одинаковыми; - Boolean: простой, двузначный тип данных, который использует ключевые слова «true» и «false», чтобы указать логическое состояние объекта; - String: последовательность печатных символов; - Unspecified: тип данных, который не известен или не важен
Value space (Область значений)	Диапазон допустимых значений для этого класса. Если область значений не определена, то это не известно или не важно
Multiplicity (Кратность)	Собственность класса, указывающая, сколько раз он может быть использован или появляться в данном родительском контексте. Значения этого свойства выражаются в виде диапазона или сокращенного диапазона использования этой нотации: - «0 .. 1» [опционально; ограничено]; - «0 .. неограничено» [опционально; неограничено]; - «1 .. 1» [обязательно; ограничено]; - «1 .. неограничено» [обязательно; неограничено]. Кратность может также появиться в сокращенных нотациях моделей UML. Сокращенные эквиваленты должны быть (без учета комментариев в квадратных скобках): - «*» [опционально; неограничено]; - «1» [обязательно; ограничено]; - «* .. 1» [обязательно; неограничено]. Где кратность больше единицы, важность упорядочения родственных значений указывается путем добавления «ordered» или «unordered»
Characteristic classes (Классы характеристик)	Списки классов характеристик, связанных с этим классом в виде “{“ characteristic *”,“ characteristic ”}”. Одна или более характеристик могут быть представлены в фигурных скобках. Характеристики должны разделяться запятыми. Когда имеются две и более характеристики, важность упорядочения родственных элементов указывается добавлением «ordered» или «unordered»
Parents (Родители)	Списки классов, которые могут быть родительскими для этого класса

Окончание таблицы 1

Дескриптор	Определение
Children (Наследники)	Перечисление возможных дочерних классов этого класса в виде "[" child *"," child "]". Один или несколько дочерних классов могут быть представлены в квадратных скобках. Каждый дочерний класс должен быть разделен запятыми. Когда имеются два и более наследников, важность упорядочения родственных элементов указывается добавлением «ordered» или «unordered»
Description (Описание)	Содержит описания, связанные с классом, и его область значений

6.2 Платформонезависимая модель информационной модели упаковки контента

На рисунке 2 приведено структурное представление информационной модели упаковки контента (CPIM).

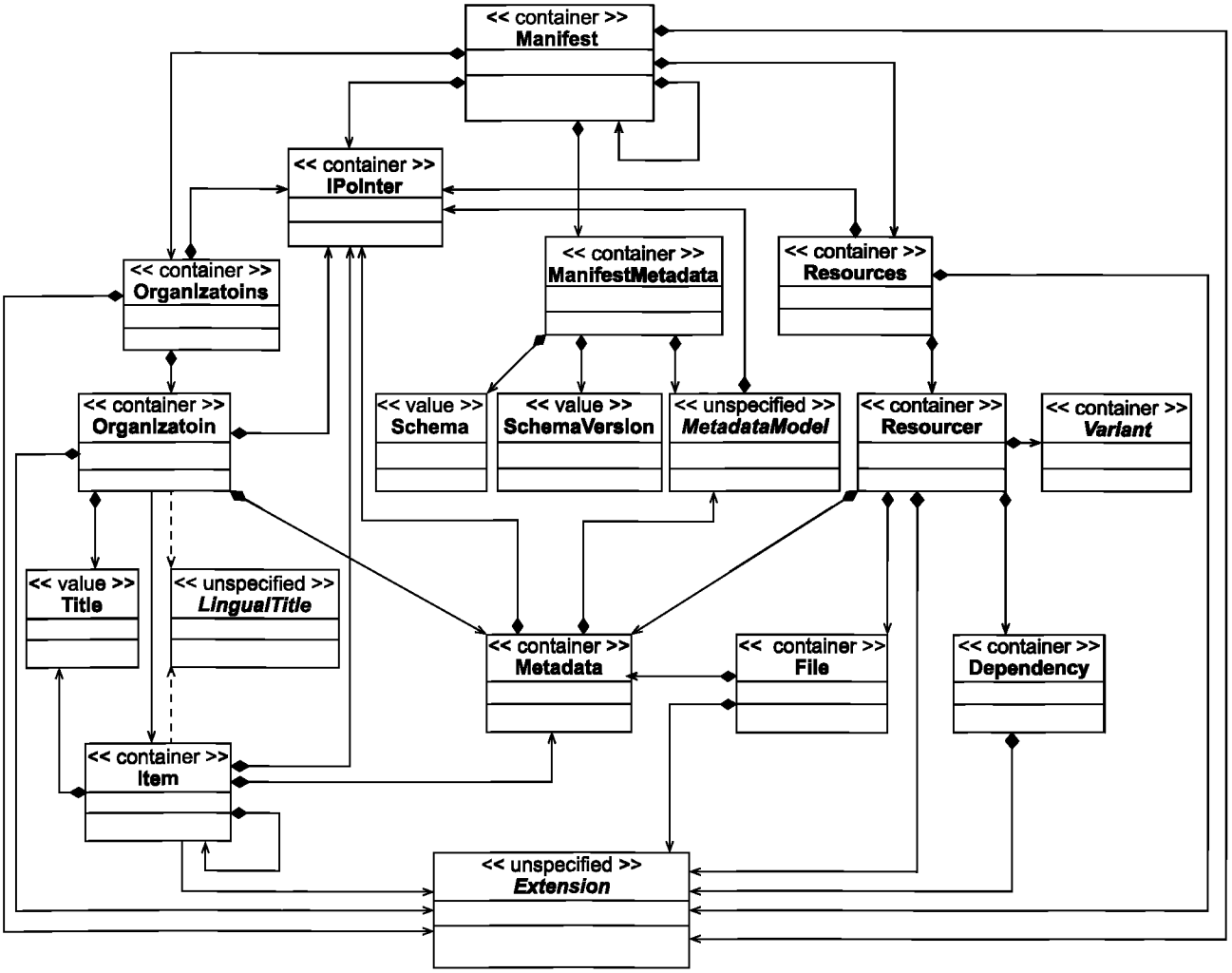


Рисунок 2 — Структурное представление упаковки контента PIM (обзор)

Структура на рисунке 2 ориентирована на отношения между начальными классами и не содержит сведений о кратности, упорядоченности или атрибутах для классов.

6.3 Класс InterchangePackage

Пакет обмена идентичен логическому пакету, когда все компоненты в логическом пакете являются локальными для пакета обмена. Критерии для определения компонент логического пакета, которые

должны быть включены в пакет обмена, выходят за рамки настоящего стандарта и должны быть определены исполнителем. Такие факторы могут включать в себя размер экземпляра пакета обмена, архитектурные особенности и/или бизнес-политики.

В таблице 2 приведен класс InterchangePackage в информационной модели упаковки контента (CPIM).

Таблица 2 — Определение класса InterchangePackage

Дескриптор	Определение
Class name (Имя класса)	InterchangePackage
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	1..1
Characteristic classes (Классы характеристик)	N/A
Parents (Родители)	Нет
Children (Наследники)	[Manifest]
Description (Описание)	Объект InterchangePackage является подмножеством логического пакета, который обменивается между системами. InterchangePackage должен соответствовать следующим условиям: a) InterchangePackage должен включать объект Manifest и может включать в себя файлы контента и управляющие файлы; b) любой файл, описанный в Manifest с использованием URI, которые должны располагаться в InterchangePackage, должен быть включен в InterchangePackage; c) все файлы, включенные в InterchangePackage, должны быть описаны в Manifest; d) файлы, содержащиеся в InterchangePackage, могут содержать внутренние ссылки на другие файлы (например, файл контента HTML может ссылаться на файл контента JPEG или файлы управления XML Schema могут ссылаться на другие файлы управления XML Schema). Когда файл, содержащийся в InterchangePackage ссылается на другой файл с помощью URI, находящийся в InterchangePackage, указанный файл должны быть описан в Manifest InterchangePackage. Файл пакета обмена (PIF) является частным случаем InterchangePackage. Считыватели и записыватели пакетов не обязаны поддерживать чтение или запись PIF. PIF должны соответствовать следующим условиям: a) физическая инкапсуляция должна быть в виде сжатого двоичного файла, соответствующего RFC 1951; b) PIF должен содержать один объект Manifest документа в корне PIF. Этот объект должен быть привязан как документ XML с именем «imsmanifest.xml». Этот документ называется корневой Manifest документа для PIF; c) любые управляющие файлы, включенные в PIF, должны размещаться в корне PIF; d) все ссылки из корня Manifest документа на файлы, содержащиеся в PIF, должны быть выполнены посредством относительных ссылок. Ссылки должны быть относительны корня PIF; e) PIF не должен включать никаких файлов с абсолютным путем (объявленным или полученным по относительной ссылке) ранее, чем объект Manifest документа в том же иерархическом пути, или когда их абсолютный путь полностью отличается от расположения Manifest документа

6.4 Семейство классов Manifest

Этот раздел определяет следующие классы в информационной модели упаковки контента (CPIM):
- Manifest;
- ManifestMetadata;
- Schema;
- SchemaVersion;
- MetadataModel.

Этот раздел определяет отношения следующих классов, определенных в 6.5.1, 6.6.1, 6.9 и 6.10 соответственно:
- Organizations;
- Resources;
- IPointer;
- Extension.

На рисунке 3 показан класс Manifest платформонезависимой модели (PIM).

6.4.1 Класс Manifest

В таблице 3 определен класс Manifest в информационной модели упаковки контента (CPIM).

Таблица 3 — Определение класса Manifest

Дескриптор	Определение
Class name (Имя класса)	Manifest
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	1..1 в случае, если дочерний InterchangePackage 0..unbounded в случае, если дочерний Manifest, unordered
Characteristic classes (Классы характеристик)	{ Identifier, Version, Base, Other }, unordered
Parents (Родители)	InterchangePackage Manifest
Children (Наследники)	[ManifestMetadata, Organizations, Resources, Manifest, IPointer, Extension], ordered
Description (Описание)	Объект Manifest представляет собой контейнер для структуры данных, который описывает полный экземпляр логического пакета. Manifest может содержать дочерние объекты класса Manifest (дочерний манифест). Дочерний манифест определяет полные экземпляры логических пакетов, которые являются частью большого логического пакета. Manifest также может содержать дочерние объекты класса IPointer, которые ссылаются на дочерние манифесты, являющиеся внешними по отношению к Manifest. Разрешено любое сочетание дочерних манифестов и дочерних манифестов по внешним ссылкам, их порядок в Manifest не имеет существенного значения. Соответствующие цели для IPointer, объявленные как прямые наследники Manifest, определены в 6.9. Manifest может содержать ссылки на локальные или удаленные компоненты родительского объекта InterchangePackage. Ссылки делаются через дочерние объекты Resource, File и IPointer. Resource и File используются для ссылок на файлы контента и управляющие файлы. IPointer используется для идентификации ссылочного манифеста. Manifest должен содержать объекты File, которые описывают все управляющие файлы, необходимые для интерпретации манифеста

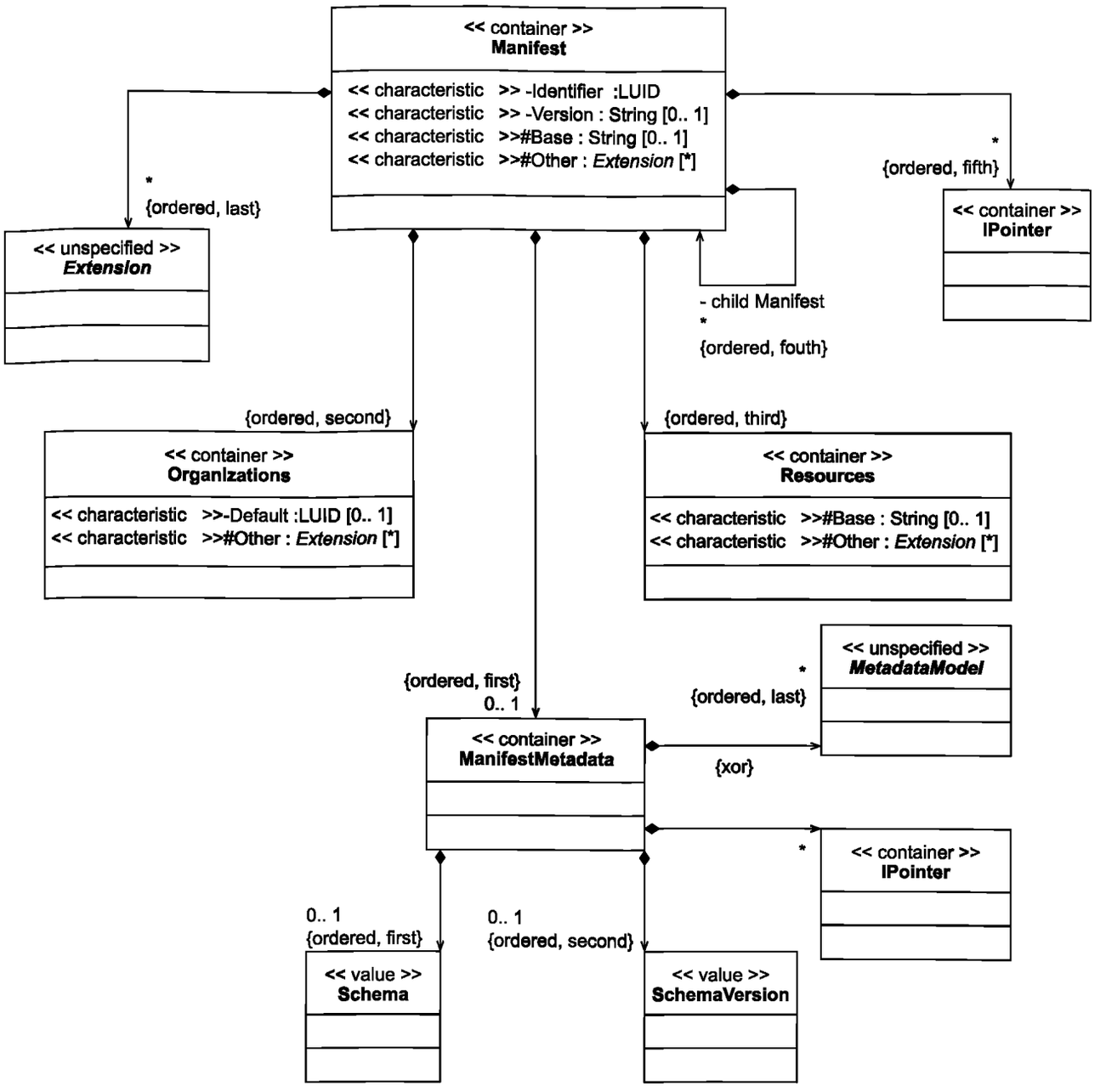


Рисунок 3 — Упрощенное представление класса Manifest в PIM

6.4.2 Класс ManifestMetadata

В таблице 4 определен класс ManifestMetadata в информационной модели упаковки контента (CPIM).

Таблица 4 — Определение класса ManifestMetadata

Дескриптор	Определение
Class name (Имя класса)	ManifestMetadata
Class type (Тип класса)	Container
Data type (Тип данных)	N/A

Окончание таблицы 4

Дескриптор	Определение
Value space (Область значений)	N/A
Multiplicity (Кратность)	0..1
Characteristic classes (Классы характеристик)	N/A
Parents (Родители)	Manifest
Children (Наследники)	[Schema, Schema Version, IPointer, Metadata Model], ordered
Description (Описание)	Объект ManifestMetadata содержит описательную информацию о родительских объектах Manifest. Целью ManifestMetadata является весь логический пакет, описываемый родительским Manifest. Schema и SchemaVersion, дочерние ManifestMetadata, представляют информацию о спецификации или профиле, который регулирует значение родительского Manifest. MetadataModel, дочерний для ManifestMetadata, служит в качестве заполнителя для общей описательной информации о своем родительском Manifest. MetadataModel является точкой расширения, которая разрешает метаданные с информационной структурой, определенные в другом пространстве имен. Множественные, отличающиеся модели метаданных могут быть объявлены как расширения, содержащиеся в одном объекте ManifestMetadata. Если метаданные определены во внешнем объекте Metadata, то ссылка на этот объект делается с использованием объекта IPointer. Разрешена любая комбинация Metadata Model и метаданных по внешним ссылкам, их порядок в объекте Metadata не имеет существенного значения. Соответствующие цели для IPointer, объявленные как наследники ManifestMetadata, определены в 6.9

6.4.3 Класс Schema

В таблице 5 определен класс Schema в информационной модели упаковки контента (CPIM).

Таблица 5 — Определение класса Schema

Дескриптор	Определение
Class name (Имя класса)	Schema
Class type (Тип класса)	Value
Data type (Тип данных)	String
Value space (Область значений)	Набор печатных символов из [UCS]
Multiplicity (Кратность)	0..1
Characteristic classes (Классы характеристик)	N/A
Parents (Родители)	ManifestMetadata
Children (Наследники)	N/A

Окончание таблицы 5

Дескриптор	Определение
Description (Описание)	Объект Schema декларирует имя спецификации или профиля для своего родительского объекта Manifest. Значение Schema информирует считыватель пакета о спецификациях или профиле, которые регулируют значение Manifest. Это значение не следует путать с названием или меткой, присвоенными данной схеме метаданных. Требования к содержанию или синтаксису строки, декларируемой как значение Schema, не предъявляются, за исключением того, что строка не должна включать информацию о версиях. Значение по умолчанию должно быть «LET контент»

6.4.4 Класс SchemaVersion

В таблице 6 определен класс SchemaVersion в информационной модели упаковки контента (CPIM).

Таблица 6 — Определение класса SchemaVersion

Дескриптор	Определение
Class name (Имя класса)	SchemaVersion
Class type (Тип класса)	Value
Data type (Тип данных)	String
Value space (Область значений)	Набор печатных символов из [UCS]
Multiplicity (Кратность)	0..1
Characteristic classes (Классы характеристик)	N/A
Parents (Родители)	ManifestMetadata
Children (Наследники)	N/A
Description (Описание)	Объект SchemaVersion декларирует версию спецификации или профиля, которые декларированы как значения для их родственного объекта Schema. Это значение сообщает считывателю пакетов о версии модели или профиля, определенных родственной Schema. Значение для SchemaVersion не должно включать никакой другой информации. Значение по умолчанию должно быть «ISO/IEC 12785:2009». «LET контент» объявлено как значение для родственной Schema, Manifest документа регулируется привязкой этой информационной модели. Ни синтаксис для информации о версиях, ни эвристические характеристики для применения любых версий синтаксиса не определяются данной информационной моделью

6.4.5 Класс MetadataModel

В таблице 7 определен класс MetadataModel в информационной модели упаковки контента (CPIM).

Таблица 7 — Определение класса MetadataModel

Дескриптор	Определение
Class name (Имя класса)	MetadataModel
Class type (Тип класса)	Unspecified

Окончание таблицы 7

Дескриптор	Определение
Data type (Тип данных)	Unspecified
Value space (Область значений)	Unspecified
Multiplicity (Кратность)	0..unbounded, ordering also unspecified
Characteristic classes (Классы характеристик)	Unspecified, ordering also unspecified
Parents (Родители)	ManifestMetadata Metadata
Children (Наследники)	Unspecified, ordering also unspecified
Description (Описание)	Объект MetadataModel является заполнителем. Он информирует привязки этой информационной модели о допустимых местоположениях для включения метаданных в пакет обмена. Фактические имена расширенных контейнеров или значения типов классов, содержащих одну или более моделей метаданных, будут известны только тогда, когда привязка этих классов будет импортирована в связанный экземпляр этой информационной модели. Таким образом, фактический тип класса не определен в этой информационной модели. MetadataModel используется двумя контейнерами метаданных: ManifestMetadata и Metadata. MetadataModel является единственным средством для выражения модели метаданных и ассоциированной с ней информации в связанном экземпляре этой информационной модели. Записыватель пакетов может представлять столько различных моделей метаданных, сколько это необходимо для адекватного описания контента, инкапсулируемого в родительский объект MetadataModel. MetadataModel должен использоваться только для объявления одной или нескольких моделей метаданных и информации, связанной с ними. Семантика MetadataModel должна оставаться в пределах семантики его родителей в этой информационной модели. Семантика MetadataModel не должна перекрывать или переопределять семантику родителей, как определено в данной информационной модели

6.5 Семейство классов Organizations

Этот раздел определяет следующие классы в информационной модели упаковки контента (CPIM):
- Organizations;
- Organization;
- Title;
- Lingual Title;
- Item.

Он также определяет отношения классов, определенных в 6.7, 6.9 и 6.10 соответственно:
- Metadata;
- IPointer;
- Extension.

На рисунке 4 показан класс Organizations платформонезависимой модели (PIM).

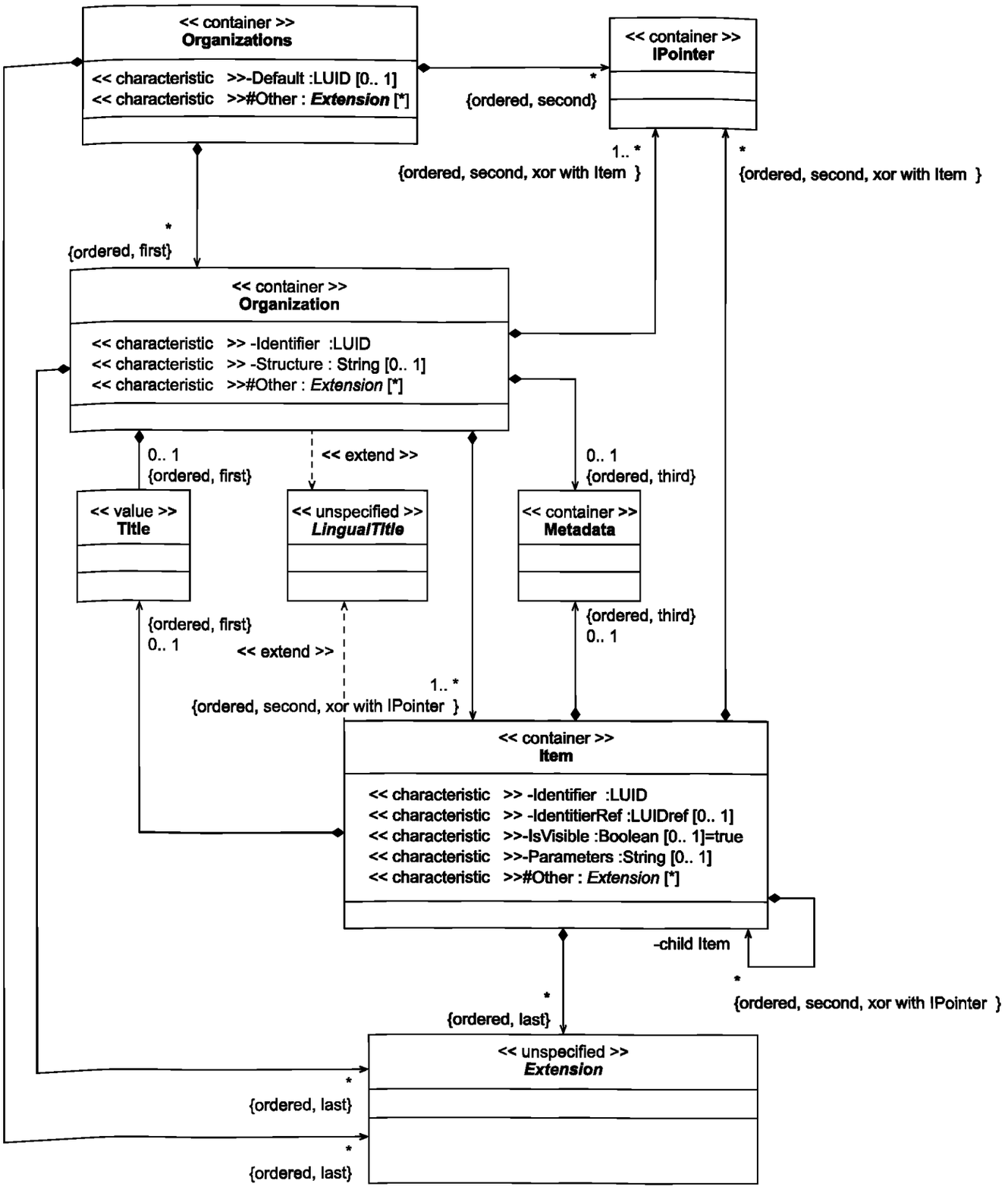


Рисунок 4 — Упрощенное представление класса PIM

6.5.1 Класс Organizations

В таблице 8 определен класс Organizations в информационной модели упаковки контента (CPIM).

Таблица 8 — Определение класса Organizations

Дескриптор	Определение
Class name (Имя класса)	Organizations
Class type (Тип класса)	container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	1..1
Characteristic classes (Классы характеристик)	{ Default, Other }, unordered
Parents (Родители)	Manifest
Children (Наследники)	[Organization, IPointer, Extension], ordered
Description (Описание)	Объект Organizations является контейнером для классов, описывающих логические отношения между объектами Resources. Более одной логической организации может быть описано с использованием Organization или IPointer, являющихся дочерними для Organizations. Порядок объектов Organization и IPointer не имеет значения. Organizations, определенные в ссылочном манифесте, идентифицируются с помощью IPointer. Соответствующие цели для IPointer, объявляемые как наследники Organizations, определены в 6.9

6.5.2 Класс Organization

В таблице 9 определен класс Organization в информационной модели упаковки контента (CPIM).

Таблица 9 — Определение класса Organization

Дескриптор	Определение
Class name (Имя класса)	Organization
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	1..unbounded, ordered
Characteristic classes (Классы характеристик)	{ Identifier, Structure, Other }, unordered
Parents (Родители)	Organizations
Children (Наследники)	[Title, LingualTitle, Item, IPointer, Metadata, Extension], ordered

Окончание таблицы 9

Дескриптор	Определение
Description (Описание)	Объект Organization представляет собой контейнер для классов, описывающих конкретные логические отношения между объектами Resource, инкапсулируемыми объектом Manifest. Несколько объектов Organization имеют эквивалентное назначение. Каждый показывает способ для структурирования такого же набора объекта Resource в пределах данного Manifest, хотя каждая Organization должна представлять различную структуру. Дочерние объекты Item используются для представления структурных узлов в Organization. Объекты Item, определенные в ссылочном манифесте, идентифицируются с помощью объектов IPointer. Допускаются любые комбинации объектов Item и IPointer, порядок объектов является важным. Соответствующие цели для IPointer, декларируемые как наследники Organization, определены в 6.9. По крайней мере один дочерний Item или IPointer должны быть объявлены для каждой Organization

6.5.3 Класс Title

В таблице 10 определен класс Title в информационной модели упаковки контента (CPIM).

Таблица 10 — Определение класса Title

Дескриптор	Определение
Class name (Имя класса)	Title
Class type (Тип класса)	Value
Data type (Тип данных)	String
Value space (Область значений)	Набор печатных символов из [15]
Multiplicity (Кратность)	1..0
Characteristic classes (Классы характеристик)	N/A
Parents (Родители)	Organization Item
Children (Наследники)	N/A
Description (Описание)	Объект Title содержит текстовое значение, применимое к объектам Organization или Item для имен или меток структур каждого представления. Текстовое значение не имеет языкового типа. Значение Title представляет собой строку символов. Исполнители, которые предполагают использование одной или более строк символов, чувствительных к языку, для заголовков должны использовать класс LingualTitle

6.5.4 Класс LingualTitle

В таблице 11 определен класс LingualTitle в информационной модели упаковки контента (CPIM).

Таблица 11 — Определение класса LingualTitle

Дескриптор	Определение
Class name (Имя класса)	LingualTitle
Class type (Тип класса)	Value

Окончание таблицы 11

Дескриптор	Определение
Data type (Тип данных)	String
Value space (Область значений)	Набор печатных символов из [15]
Multiplicity (Кратность)	0..unbounded, unordered
Characteristic classes (Классы характеристик)	{ Language }
Parents (Родители)	Organization Item
Children (Наследники)	N/A
Description (Описание)	Объект LingualTitle содержит текстовое значение конкретного языка применительно к объектам Organization или Item для имен или маркировок структур каждого представления. Язык LingualTitle определяется объектом Language

6.5.5 Класс Item

В таблице 12 определен класс Item в информационной модели упаковки контента (CPIM).

Таблица 12 — Определение класса Item

Дескриптор	Определение
Class name (Имя класса)	Item
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	1..unbounded, ordered
Characteristic classes (Классы характеристик)	{ Identifier, IdentifierRef, IsVisible, Parameters, Other }, unordered
Parents (Родители)	Organization Item
Children (Наследники)	[Title, Item, IPointer Metadata, Extension], ordered
Description (Описание)	Объект Item представляет собой контейнер для представления структурных узлов в конкретной Organization или в другой объект Item. Объекты Item могут содержать дочерний Item или объект IPointer, каждый из которых представляет уникальный структурный узел. Объекты Item, определенные в ссылочном манифесте, идентифицируются с использованием класса IPointer. Допускаются любые комбинации Item и IPointer в пределах Item, порядок объектов является значительным. Соответствующие цели для IPointer, декларируемые как дочерние Item, определены в 6.9. Item может использоваться как объект IdentifierRef для представления внутренней ссылки на объект Resource, который должен быть связан с его структурной ролью и положением. Item может использовать IdentifierRef для представления внутренней ссылки на объект дочернего манифеста, который должен быть связан с его структурной ролью и положением.

Окончание таблицы 12

Дескриптор	Определение
Description (Описание)	В этом случае ссылка формируется с учетом того, что: a) ссылающийся Item и все его наследники заменяются коллекцией объектов, содержащихся в Organization по умолчанию ссылочного дочернего манифеста или первой Organization в ссылочном дочернем манифесте, если ссылочный дочерний манифест не имеет Organization по умолчанию; b) родственные объекты Item заменяемого ссылочного Item смещаются в последовательность дочерних объектов косвенно ссылаемого Item и его родственных объектов. Считывателю пакетов не нужно фактически создавать модель памяти или хранения данных моделей, которая имитирует фактическое склеивание или соединения двух структур, описанных выше. Тем не менее считыватель пакетов должен интерпретировать ссылки так, как если бы описанная выше структура была бы создана

6.6 Семейство классов Resources

Этот раздел определяет следующие классы в информационной модели упаковки контента (CPIM):

- Resources;
- Resource;
- File;
- Dependency.

Он также определяет отношения следующих классов, определенных в 6.7, 6.8, 6.9 и 6.10 соответственно:

- Metadata;
- Variant;
- IPointer;
- Extension.

На рисунке 5 показан класс Resources платформонезависимой модели (PIM).

6.6.1 Класс Resources

В таблице 13 определен класс Resources в информационной модели упаковки контента (CPIM).

Таблица 13 — Определение класса Resources

Дескриптор	Определение
Class name (Имя класса)	Resources
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	1..1
Characteristic classes (Классы характеристик)	{ Base, Other }, unordered
Parents (Родители)	Manifest
Children (Наследники)	[Resource, IPointer, Extension], ordered
Child grouping model (Модель группировки наследников)	Ordered
Description (Описание)	Объект Resources является контейнером для всей информации о файлах, используемых в родительском объекте Manifest. Эти файлы могут быть локальными или удаленными.

Окончание таблицы 13

Дескриптор	Определение
Description (Описание)	Resources описывают файлы для своих родительских Manifest. Ссылочные файлы в любом другом объекте Manifest, включая объекты Manifest, которые являются родственными, потомками или предками родительского Manifest, выходят за рамки области действия объекта Resources. Наследники Resources и IPointer объекта Resources используются для описания конкретного набора файлов. IPointer используется для идентификации Resource в ссылочном манифесте. Порядок объектов дочерних Resource и IPointer значения не имеют. Соответствующие цели для дочернего IPointer объекта Resources определены в 6.9

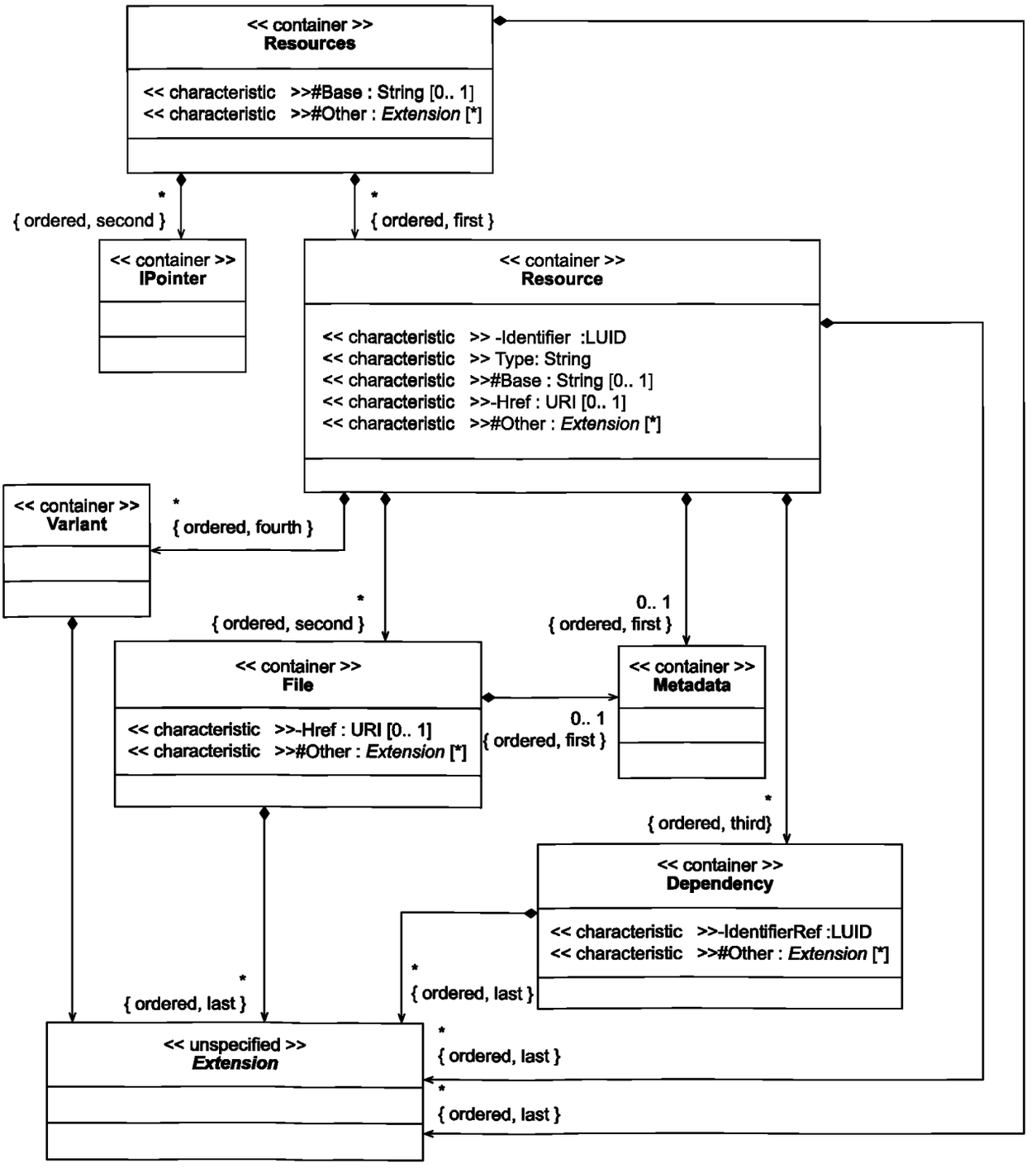


Рисунок 5 — Упрощенный вид класса Resources PIM

6.6.2 Класс Resource

В таблице 14 определен класс Resource в информационной модели упаковки контента (CPIM).

Таблица 14 — Определение класса Resource

Дескриптор	Определение
Class name (Имя класса)	Resource
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	0..unbounded, unordered
Characteristic classes (Классы характеристик)	{ Identifier, Type, Base, Href, Other }, unordered
Parents (Родители)	Resources
Children (Наследники)	[Metadata, File, Dependency, Variant, Extension], ordered
Description (Описание)	Объект Resource является контейнером для информации, относящейся к конкретному набору файлов, используемых предком объекта Manifest. Варианты ресурсов определяются с помощью класса Variant. Отношения между родительским Resource и набором объектов Variant должны быть описаны с использованием метаданных в варианте ресурсов. Значение, заявленное объектом Resource { Href }, является исполняемым URI. Файл ссылок, декларированный в Resource { Href }, должен иметь связанную декларацию в объекте File в том же Resource. Href соответствующего объекта File должен ссылаться на тот же файл, что и Resource { Href }. Resource { Href } и соответствующий File { Href } URI могут отличатся тем, что Resource { Href } может содержать «исполняемые» параметры в URI

6.6.3 Класс File

Таблица 15 определяет класс File в информационной модели упаковки контента (CPIM).

Таблица 15 — Определение класса File

Дескриптор	Определение
Class name (Имя класса)	File
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	0..unbounded, unordered
Characteristic classes (Классы характеристик)	{ Href, Other }, unordered
Parents (Родители)	Resource

Окончание таблицы 15

Дескриптор	Определение
Children (Наследники)	[Metadata, Extension], ordered
Description (Описание)	Объект File является контейнером для всей информации, относящейся к одиночному компьютерному файлу, инкапсулированному его родительским объектом Resource. File содержит метаданные, описывающие компьютерный файл, и ссылку на местоположение файла

6.6.4 Класс Dependency

В таблице 16 определен класс Dependency в информационной модели упаковки контента (CPIM).

Таблица 16 — Определение класса Dependency

Дескриптор	Определение
Class name (Имя класса)	Dependency
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	0..unbounded, unordered
Characteristic classes (Классы характеристик)	{ IdentifierRef, Other }, unordered
Parents (Родители)	Resource
Children (Наследники)	[Extension]
Description (Описание)	Объект Dependency позволяет объекту Resource ссылаться на набор файлов, описанных в родственных объектах Resource. Dependency должен использовать объект IdentifierRef для ссылки на объекты Resource или IPointer. Ссылочные Resource или IPointer должны быть инкапсулированы прародительским объектом Resources объекта Dependency. Если имеется ссылка на IPointer, тогда IPointer должен ссылаться на Resource в пределах ссылочного манифеста. Результатом ссылки является то, что файлы, включенные в сферу ссылочного Resource, должны рассматриваться в рамках ссылочных родительских для Dependency Resource. Объекты характеристик, связанные со ссылочным Resource, не должны рассматриваться в рамках ссылочных родительских для Dependency Resource

6.7 Класс Metadata

В этом разделе определяется абстракция класса Metadata, который является частью информационной модели упаковки контента (CPIM). Он также определяет отношения с IPointer, определенным в 6.9.

На рисунке 6 показан класс Metadata платформонезависимой модели (PIM).

В таблице 17 определен класс Metadata в информационной модели упаковки контента (CPIM).

Таблица 17 — Определение класса Metadata

Дескриптор	Определение
Class name (Имя класса)	Metadata

Окончание таблицы 17

Дескриптор	Определение
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	0..1
Characteristic classes (Классы характеристик)	нет
Parents (Родители)	Organization Item Resource File
Children (Наследники)	[IPointer] или [MetadataModel]
Description (Описание)	Объект Metadata содержит описательную информацию о своих родительских объектах класса Container. Область действия Metadata — только класс родительского контейнера. Дочерний объект MetadataModel служит в качестве контейнера для общей описательной информации о его родителях. MetadataModel является точкой расширения, которая разрешает метаданные с информационной структурой, которая определена в другом пространстве имен. Несколько отличающихся моделей метаданных могут быть объявлены как расширения, содержащиеся в пределах одного объекта MetadataModel. Если метаданные определены во внешнем объекте, то ссылка на этот объект достигается с помощью объекта IPointer. Разрешена любая комбинация MetadataModel и внешних ссылочных метаданных, их порядок в объекте Metadata не существенен. Соответствующие цели для IPointer, объявляемые как наследники Metadata, определены в 6.9

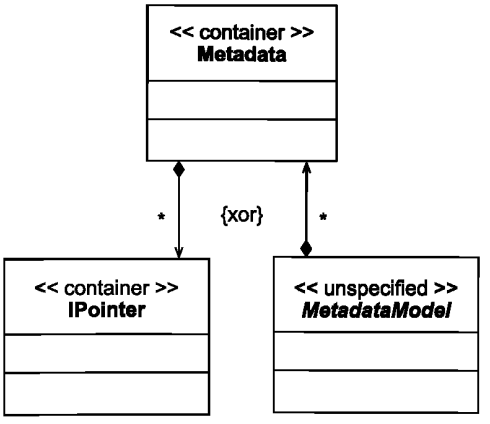


Рисунок 6 — Упрощенное представление класса Metadata PIM

6.8 Класс Variant

В этом разделе определяется класс Variant, который является частью информационной модели упаковки контента (CPIM).

Он также определяет отношения со следующими классами, определенными в 6.6.2 и 6.7 соответственно:

- Resource;
- Metadata.

На рисунке 7 показан класс Variant платформонезависимой модели (PIM).

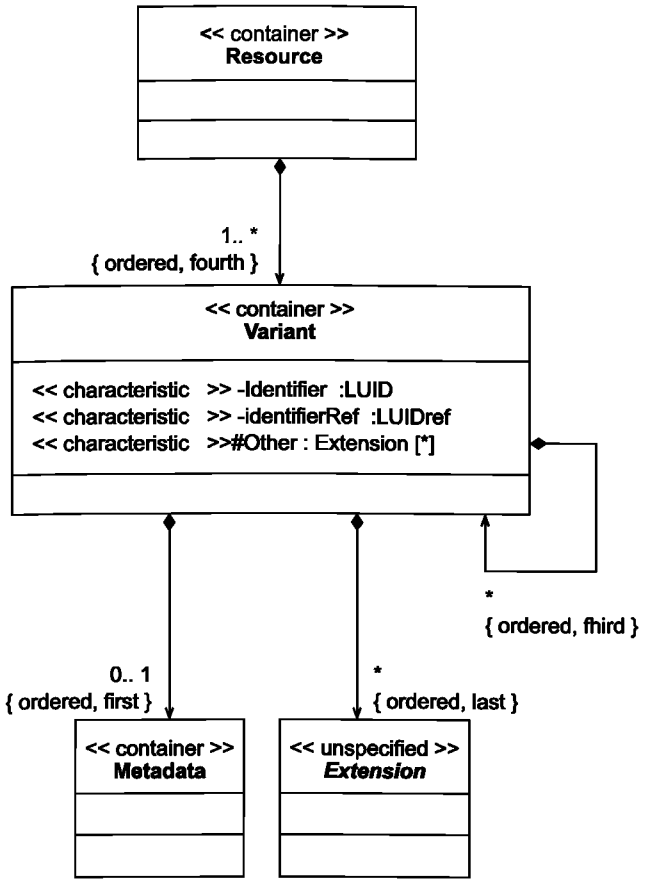


Рисунок 7 — Упрощенное представление класса Variant PIM

В таблице 18 определен класс Variant в информационной модели упаковки контента (CPIM).

Таблица 18 — Определение класса Variant

Дескриптор	Определение
Class name (Имя класса)	Variant
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	1..unbounded, unordered
Characteristic classes (Классы характеристик)	{ Identifier, IdentifierRef, Other }, unordered
Parents (Родители)	Resource
Children (Наследники)	[Metadata, Extension], ordered

Окончание таблицы 18

Дескриптор	Определение
Description (Описание)	Объект Variant позволяет объекту Resource ссылаться и описывать варианты объекта Resource. Variant должен использовать объект IdentifierRef для ссылки на объекты Resource или IPointer. Ссылочные Resource или IPointer должны быть инкапсулированы прародительским объектом Resources объекта Variant. Если ссылаются на IPointer, то IPointer должен ссылаться на Resource в ссылочном манифесте. Дочерний объект Metadata для Variant должен быть использован для описания отношения между Variant и родительским Resource

6.9 Класс IPointer

В этом разделе определяется класс IPointer, который является частью информационной модели упаковки контента (CPIM).

Он также определяет отношения со следующими классами, определенными в 6.4.1, 6.4.2, 6.5.1, 6.5.2, 6.5.5, 6.6.1 и 6.7 соответственно:

- Manifest;
- ManifestMetadata;
- Metadata;
- Organizations;
- Organization;
- Item;
- Resources.

На рисунке 8 показан класс IPointer платформонезависимой модели (PIM).

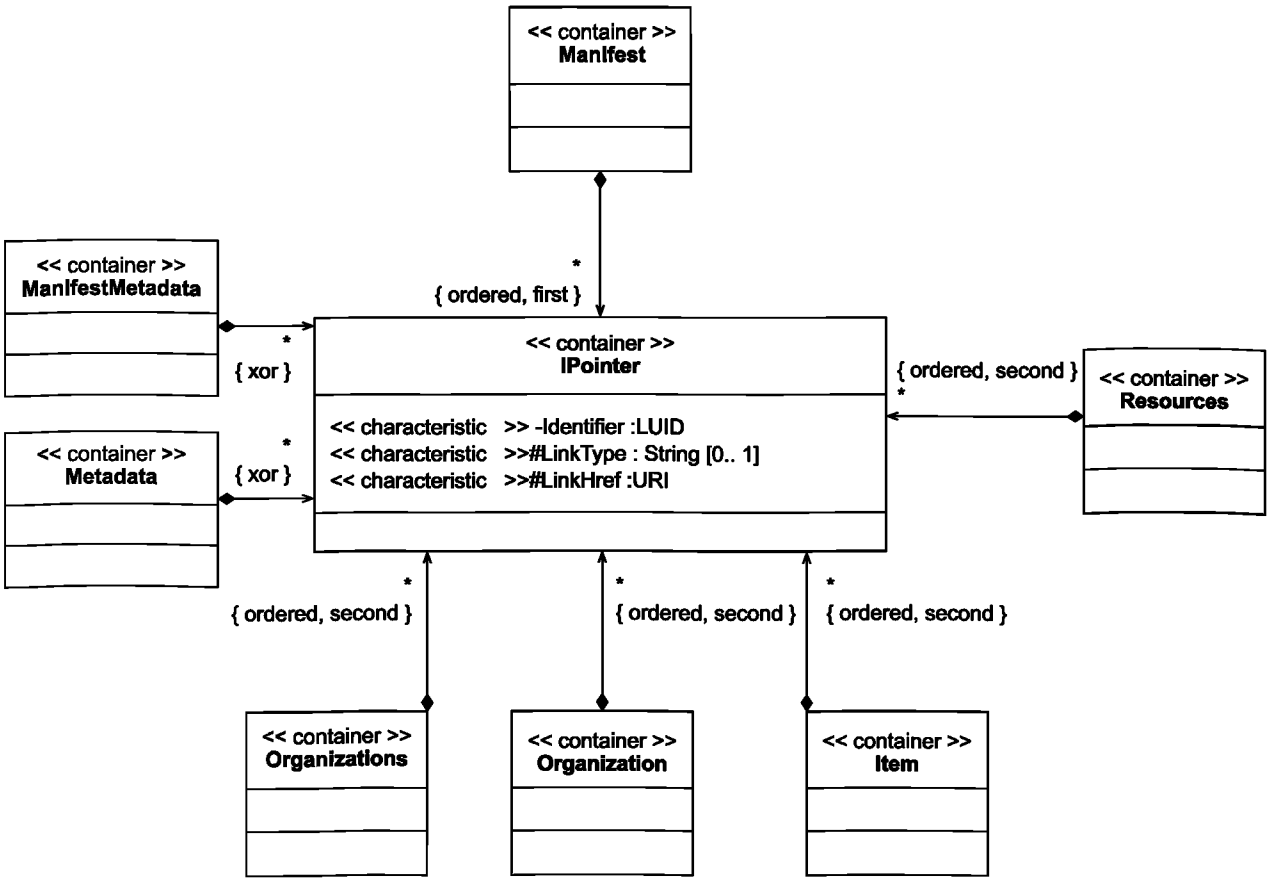


Рисунок 8 — Упрощенное представление класса IPointer PIM

В таблице 19 определен класс IPointer в информационной модели упаковки контента (CPIM).

Таблица 19 — Определение класса IPointer

Дескриптор	Определение
Class name (Имя класса)	IPointer
Class type (Тип класса)	Container
Data type (Тип данных)	N/A
Value space (Область значений)	N/A
Multiplicity (Кратность)	0..unbounded, ordered
Characteristic classes (Классы характеристик)	{ Identifier, LinkType, LinkHref, Other }, unordered
Parents (Родители)	Manifest ManifestMetadata Metadata Organizations Organization Item Resources
Children (Наследники)	N/A
Description (Описание)	Объект IPointer является связующим объектом. Его целью является определение набора узлов в документе Manifest и привязка этого набора узлов к его родителям. Источник определяемого набора узлов может быть локальным или удаленным. Набор узлов, определяемый IPointer, должен быть действительным наследником родительского класса для IPointer (см. таблицу 20)

В таблице 20 определены разрешенные комбинации связей класса-родителя и класса-цели, допустимые при использовании класса IPointer.

Таблица 20 — Допустимые сочетания связей родительского/целевого классов

Родительский класс класса IPointer	Действительный целевой набор узлов для класса IPointer
Manifest	Manifest
Organizations	Organization
Organization	Item
Item	Item
Resources	Resource
ManifestMetadata	MetadataModel
Metadata	MetadataModel

6.10 Класс Extension

В этом разделе определяется класс Extension, который является частью информационной модели упаковки контента (CPIM). На рисунке 9 показан класс Extension платформонезависимой модели (PIM).

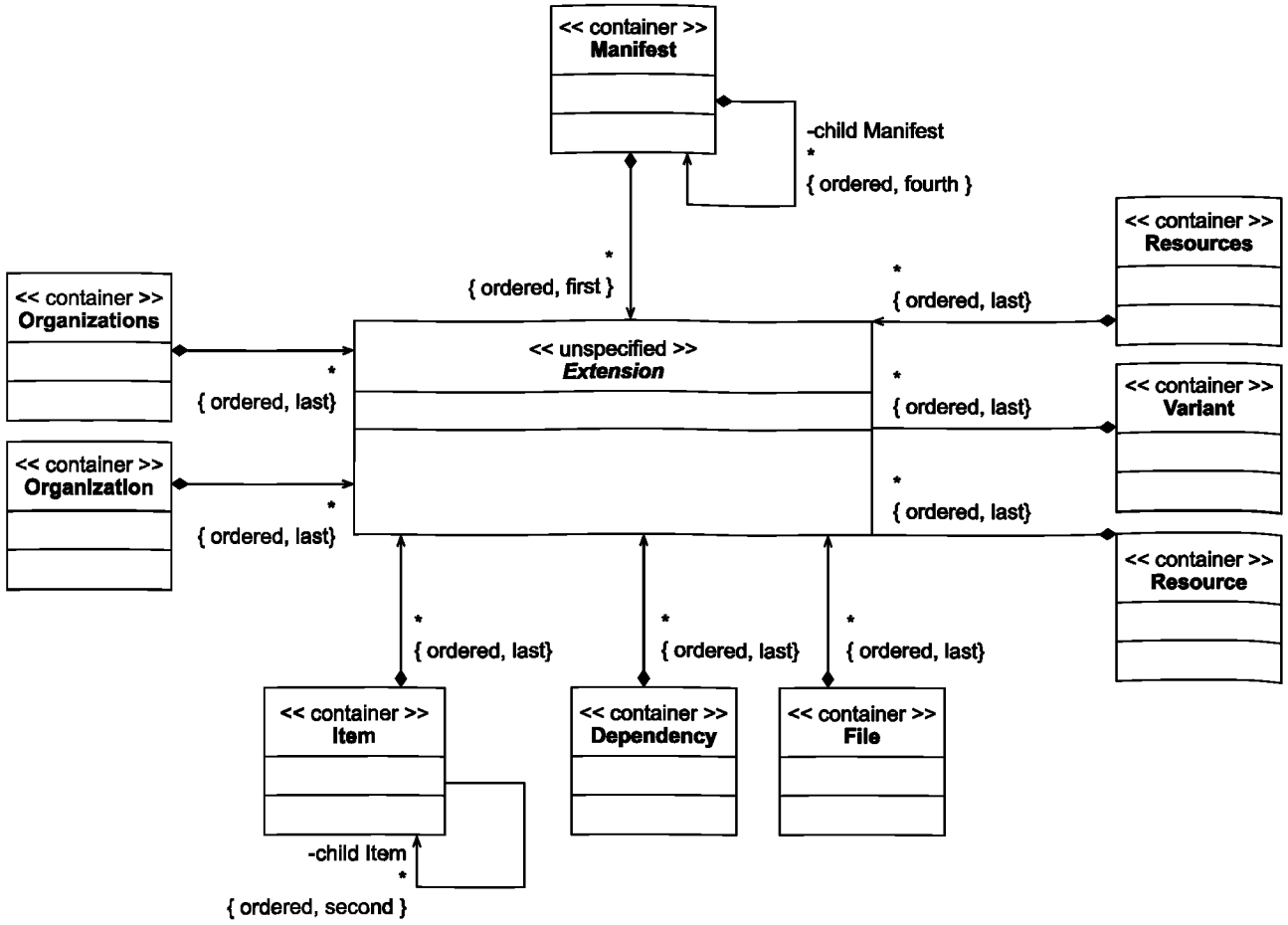


Рисунок 9 — Упрощенное представление класса Extension PIM

В таблице 21 определен класс Extension в информационной модели упаковки контента (CPIM).

Таблица 21 — Определение класса Extension

Дескриптор	Определение
Class name (Имя класса)	Extension
Class type (Тип класса)	Unspecified
Data type (Тип данных)	Unspecified
Value space (Область значений)	Unspecified
Multiplicity (Кратность)	0..unbounded, ordered
Characteristic classes (Классы характеристик)	Unspecified, ordering, также unspecified
Parents (Родители)	Manifest Organizations Organization Item Resources Resource

Окончание таблицы 21

Дескриптор	Определение
Parents (Родители)	Variant File Dependency
Children (Наследники)	Unspecified, ordering, также unspecified
Description (Описание)	Объект Extension является заполнителем. Он информирует привязки этой информационной модели о допустимом местоположении для охвата значений или контейнерных классов, которые расширяются любым классом типа Container в этой информационной модели. Класс Extension является одним из двух механизмов для расширения любого класса типа Container в этой информационной модели. Вторым механизмом расширения является класс Other. Объект Extension используется для расширения классов типа Container. Фактическое имя расширяемого контейнера или значение типа класса, используемое вместо объекта Extension, будет известно только тогда, когда привязка этого объекта импортируется в связанный экземпляр этой информационной модели. Таким образом, фактический тип класса не определен в этой информационной модели. Семантика Extension должна оставаться в пределах сферы семантики его родительского объекта в этой информационной модели. Семантика Extension не должна отменять или переопределять семантику любого родительского объекта, определенного в этой информационной модели

6.11 Классы характеристик

6.11.1 Класс Base

В таблице 22 определен класс Base в информационной модели упаковки контента (CPIM).

Таблица 22 — Определение класса Base

Дескриптор	Определение
Class name (Имя класса)	Base
Class type (Тип класса)	Characteristic
Data type (Тип данных)	URI
Value space (Область значений)	Зависит от языковой привязки
Multiplicity (Кратность)	0..1
Parents (Родители)	Manifest Resources Resource
Description (Описание)	Объект Base используется для указания базы объектов URI для объекта, с которым связан Base с целью восстановления относительных ссылок, которые появляются в потомках ассоциированного объекта. Область значений Base должна применяться ко всем потомкам объекта, с которым ассоциирован Base, если значения, объявленные для Base, заменяются или дополняются более поздним объектом Base в разрешенном наследнике. Все относительные сегменты пути, выраженные в виде значений для Base, должны направляться в документ манифеста, содержащий эти выражения. Относительные сегменты пути должны быть сконструированы таким образом, чтобы сегмент, выраженный Base, мог быть добавлен к относительному сегменту пути, выраженному в ближайшем предке объекта, выражающего значения в его объекте Base, и так далее до корневого объекта Manifest. Результирующая последовательность собранных сегментов относительного пути будет добавлена в место, в котором находится документ Manifest, или будет обрабатываться для создания абсолютного пути. Относительные ссылки, появляющиеся в области применения объекта Base, должны интерпретироваться в соответствии с правилами, определенными в RFC 3986

6.11.2 Класс Default

В таблице 23 определен класс Default в информационной модели упаковки контента (CPIM).

Таблица 23 — Определение класса Default

Дескриптор	Определение
Class name (Имя класса)	Default
Class type (Тип класса)	Characteristic
Data type (Тип данных)	Зависит от языковой привязки (по умолчанию — string)
Value space (Область значений)	Значения из Organizations { Default }, объект Organization { Identifier }
Multiplicity (Кратность)	0..1
Parents (Родители)	Organizations
Description (Описание)	Объект Default обозначает единичный дочерний объект Organization в объекте Organizations в качестве первичной или организационной структуры по умолчанию для данного объекта Manifest. Обозначение Organization по умолчанию должно быть сделано с помощью ссылки на значение объекта Identifier целевого объекта Organization. Целевой Organization должен быть дочерним для объекта Organizations, который имеет Default. Другие ссылки Default не допускаются. Когда Default не объявлен для объекта Organizations, первый определеный объект Organization в объекте Organizations должен рассматриваться как основной или организационной структурой по умолчанию

6.11.3 Класс Href

В таблице 24 определен класс Href в информационной модели упаковки контента (CPIM).

Таблица 24 — Определение класса Href

Дескриптор	Определение
Class name (Имя класса)	Href
Class type (Тип класса)	Characteristic
Data type (Тип данных)	URI
Value space (Область значений)	Как определено в RFC 3986
Multiplicity (Кратность)	0..1 : Resource 1..1 : File
Parents (Родители)	Resource File
Description (Описание)	Объект Href используется для определения положения ресурса. Значение, заявленное для Href, должно быть синтаксически правильным URI. Этот документ не гарантирует, что объявленная строка является допустимым URI или будет расположена в актуальном ресурсе или его части в идентифицированном местоположении. Значение, заявленное для Manifest.Resources.Resource{Href}, представляет собой URL, который может быть использован для нахождения и получения доступа к контенту описываемого ресурса.

Окончание таблицы 24

Дескриптор	Определение
Description (Описание)	Считыватель пакетов не требуется восстановления URI. URI предназначен для сохранения с целью последующего использования другими программными компонентами после того, как содержимое пакета обмена будет обработано. Значение, заявленное для Manifest.Resources.Resource.File {Href}, должно определять положение единичного цифрового ресурса без другого значения как использование идентифицированного ресурса

6.11.4 Класс Identifier

В таблице 25 определен класс Identifier в информационной модели упаковки контента (CPIM).

Таблица 25 — Определение класса Identifier

Дескриптор	Определение
Class name (Имя класса)	Identifier
Class type (Тип класса)	Characteristic
Data type (Тип данных)	Привязки зависят от языка локально уникального идентификатора (LUID)
Value space (Область значений)	Зависит от языка привязки
Multiplicity (Кратность)	1..1
Parents (Родители)	Manifest Organization Item Resource IPointer Variant
Description (Описание)	Объект Identifier однозначно определяет родителей объекта Identifier в пределах объекта Manifest. Это означает, что только одно вхождение заданного значения для Identifier может появляться в том же Manifest, в том числе в объекте Manifest дочернего манифеста. Значение Identifier может использоваться для внутренних ссылок из другого объекта с использованием объекта IdentifierRef

6.11.5 Класс IdentifierRef

В таблице 26 определен класс IdentifierRef в информационной модели упаковки контента (CPIM).

Таблица 26 — Определение класса IdentifierRef

Дескриптор	Определение
Class name (Имя класса)	IdentifierRef
Class type (Тип класса)	Characteristic
Data type (Тип данных)	Привязки зависят от языка локально уникального идентификатора (LUID)
Value space (Область значений)	Значение для объекта Identifier в объекте IdentifierRef сразу включает объект Manifest, а также ограничивается определенными ниже правилами внутренних ссылок
Multiplicity (Кратность)	0..1 : Item 1..1 : Dependency 1..1 : Variant

Окончание таблицы 26

Дескриптор	Определение
Parents (Родители)	Item Dependency Variant
Description (Описание)	Объект IdentifierRef должен точно дублировать значение объекта Identifier, который является потомком объекта IdentifierRef объекта-хранилища Manifest. Объект-хранилище Manifest определяется как первое появление Manifest в цепи родительских объектов. Ссылочный Identifier может быть у дочернего манифеста объекта-хранилища Manifest. Кроме того, IdentifierRef ограничивается следующими внутренними правилами привязки: a) дочерний IdentifierRef объекта Item может ссылаться на один из следующих: 1) объект Resource или IPointer, который является потомком объекта-хранилища Manifest ссылочного объекта Item; 2) Manifest или IPointer, который является потомком объекта-хранилища Manifest ссылочного Item; 3) Resource или IPointer, который содержится в объекте дочернего манифеста, который является потомком объекта-хранилища Manifest ссылочного Item. Не допускаются ссылки из IdentifierRef Item на объекты дочернего манифеста на любой объект в родительском манифесте; b) дочерний IdentifierRef объекта Dependency: 1) должен ссылаться только на Resource, который родственен родительскому Resource объекта Dependency; 2) не должен ссылаться на родительский Resource объекта Dependency; 3) не должен ссылаться на любой объект в Manifest, который является потомком или предком объекта-хранилища Manifest объекта Dependency; c) дочерний IdentifierRef объекта Variant: 1) должен ссылаться только на Resource или IPointer, который родственен родительскому Resource объекта Variant; 2) не должен ссылаться на родительский Resource объекта Variant; 3) не должен ссылаться на любой объект в Manifest, который является потомком или предком объекта-хранилища Manifest объекта Variant

Ограничивающие правила для манифестов и (суб)манифестов представлены на рисунке 10.

6.11.6 Класс IsVisible

В таблице 27 определен класс IsVisible в информационной модели упаковки контента (CPIM).

Таблица 27 — Определение класса IsVisible

Дескриптор	Определение
Class name (Имя класса)	IsVisible
Class type (Тип класса)	Characteristic
Data type (Тип данных)	Логический
Value space (Область значений)	Истина (по умолчанию) Ложь
Multiplicity (Кратность)	0..1
Parents (Родители)	Item
Description (Описание)	Объект IsVisible сигнализирует процессу генерации, следует отображать текстовую строку, объявленную в родственном объекте Title, или визуально обозначить наличие в объекте Item в любом другом случае. Этот флаг не предполагает других действий. Отсутствует наследование видимости состояния, декларированного этим IsVisible для любого наследника Item, для родительского Item этого IsVisible. Значение «истина» должно быть значением по умолчанию для IsVisible, даже если он не объявлен в связанном экземпляре Item. То есть при отсутствии IsVisible для

Окончание таблицы 27

Дескриптор	Определение
Description (Описание)	родительского Item означает то же самое, как если бы IsVisible со значением «истина» был заявлен в Item. Значение «истина» должно толковаться как означающее, что контент родственного Title должен отображаться генерирующим приложением (например, Item{IsVisible}.Title). Значение «ложь» должно толковаться как означающее, что содержание контента родственного Title не должно отображаться генерирующим приложением (например, Item{IsVisible}.Title)

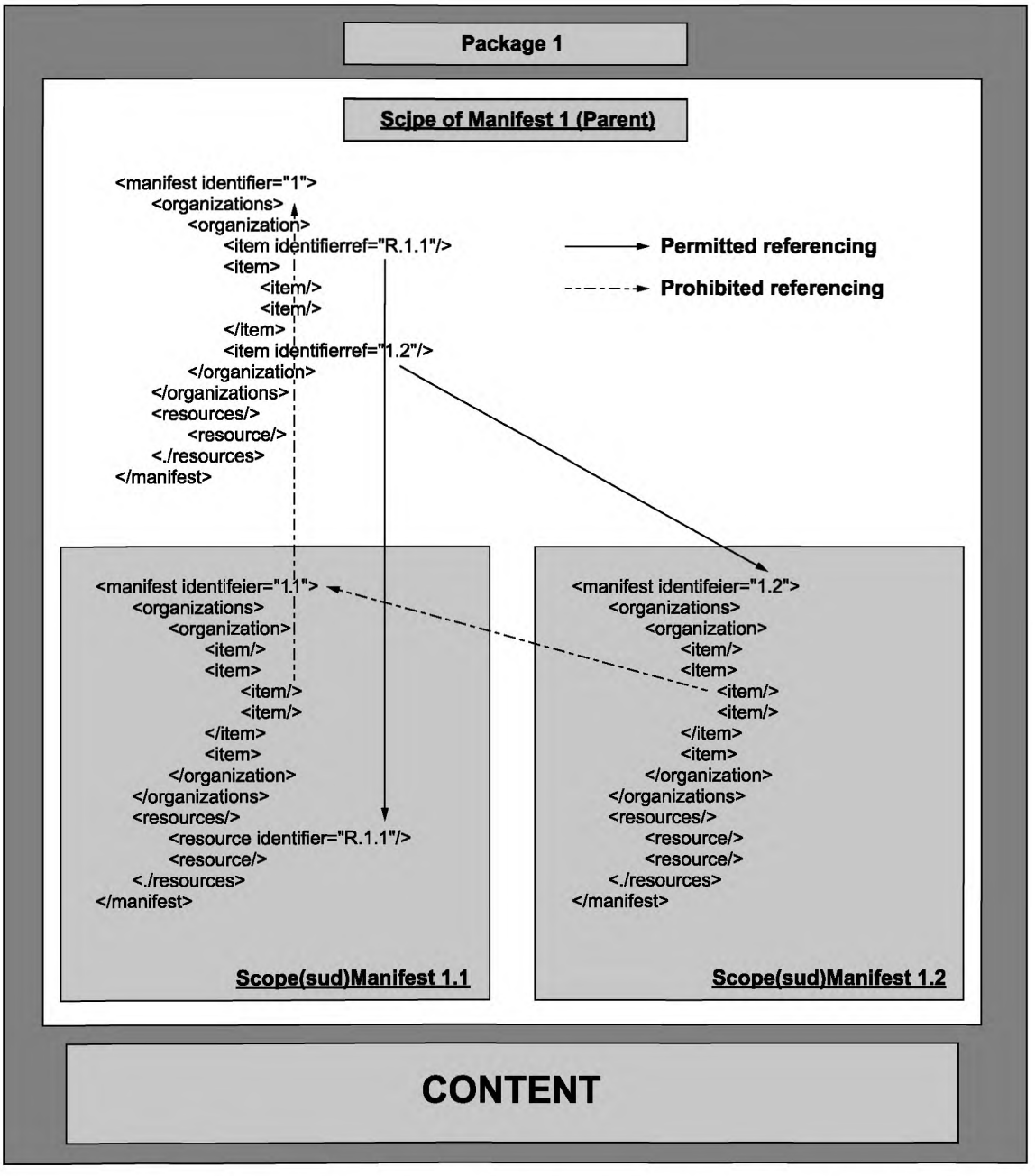


Рисунок 10 — Ограничивающие правила для манифестов и (суб)манифестов

6.11.7 Класс Language

В таблице 28 определен класс Language в информационной модели упаковки контента (CPIM).

Таблица 28 — Определение класса Language

Дескриптор	Определение
Class name (Имя класса)	Language
Class type (Тип класса)	Characteristic
Data type (Тип данных)	String
Value space (Область значений)	Область значений для элемента 1.3 Language определена в IEEE 1484.12.1. [16] Область значений определяется следующими правилами ABNF: language-id = lang-code *lang-sub-code ; lang-code is any language code defined by ISO 639-2/T [17] lang-sub-code = «-» country-code ; country-code is any country code defined by ; ISO 3166-1 [18]
Multiplicity (Кратность)	1
Parents (Родители)	LingualTitle
Description (Описание)	Объект Language определяет язык строки, содержащейся в родительском объекте LingualTitle

6.11.8 Класс LinkHref

В таблице 29 определен класс LinkHref в информационной модели упаковки контента (CPIM).

Таблица 29 — Определение класса LinkHref

Дескриптор	Определение
Class name (Имя класса)	LinkHref
Class type (Тип класса)	Characteristic
Data type (Тип данных)	URI
Value space (Область значений)	Зависит от привязки языка. Ограничивается правилами внешних ссылок, определены ниже
Multiplicity (Кратность)	1
Parents (Родители)	IPointer
Description (Описание)	Объект LinkHref однозначно идентифицирует набор объектов в манифесте либо в том же или другом («удаленном») документе манифеста. Значения, заявленные для LinkHref, должны быть синтаксически правильными абсолютными или относительными ссылками. Так как URL не имеет ограничений по расположению, область применения характеристик является глобальной

6.11.9 Класс LinkType

В таблице 30 определен класс LinkType в информационной модели упаковки контента (CPIM).

Таблица 30 — Определение класса LinkType

Дескриптор	Определение
Class name (Имя класса)	LinkType

Окончание таблицы 30

Дескриптор	Определение
Class type (Тип класса)	Characteristic
Data type (Тип данных)	String
Value space (Область значений)	Зависит от привязки языка
Multiplicity (Кратность)	0..1
Parents (Родители)	IPointer
Description (Описание)	Объект LinkType обеспечивает место для декларирования термина или ключевого слова (например, слово или фраза словаря), которому было назначено особое значение. Упаковка контента должна использовать «простое» ключевое слово в качестве значения для этого объекта. Область применения LinkType ограничивается его родителем. Примечание — Простая ссылка — это ссылка, которая точно связывает два ресурса, один локальный и один удаленный, дугой, идущей от первого к последнему. Таким образом, простая ссылка всегда является исходящей ссылкой

6.11.10 Класс Other

В таблице 31 определен класс Other в информационной модели упаковки контента (CPIM).

Таблица 31 — Определение класса Other

Дескриптор	Определение
Class name (Имя класса)	Other
Class type (Тип класса)	Characteristic
Data type (Тип данных)	Неизвестно для данной информационной модели
Value space (Область значений)	Неизвестно для данной информационной модели
Multiplicity (Кратность)	0..unbounde
Parents (Родители)	Manifest Metadata Organizations Organization Item Resources Resource File Dependency
Description (Описание)	Объект Other является точкой расширения для характеристик контейнерных классов, объявленных как логические эквиваленты характеристических классов. Other является заполнителем. Он информирует привязки этой информационной модели, чтобы указать допустимые места для представления характеристических классов, которые определены информационной моделью, отличной от этой информационной модели. Фактическое имя расширения типа характеристического класса, используемое вместо него, должно быть известно только тогда, когда привязка этого класса импортируется в связанный экземпляр этой информационной модели. Семантики любого расширения характеристического класса должны оставаться в пределах области деятельности семантики класса, родительского по отношению

Окончание таблицы 31

Дескриптор	Определение
Description (Описание)	к данному классу, определенному в этой информационной модели. Семантики любых расширений характеристических классов не должны перекрывать или переопределять семантику любого родительского класса, определенного в этой информационной модели. Область действия Other ограничивается его родительским объектом

6.11.11 Класс Parameters

В таблице 32 определен класс Parameters в информационной модели упаковки контента (CPIM).

Таблица 32 — Определение класса Parameters

Дескриптор	Определение
Class name (Имя класса)	Parameters
Class type (Тип класса)	Characteristic
Data type (Тип данных)	String
Value space (Область значений)	См. описание ниже
Multiplicity (Кратность)	0..1
Parents (Родители)	Item
Description (Описание)	Объект Parameters предоставляет место для объявления достоверной статической, специфичной для приложения информации, которая должна всегда быть связана с объектом Item. Параметры не определены в этой информационной модели. Как правило, параметры определяются для использования приложением, которое работает с информацией, объявленной Item, включая любые файлы, определенные в любом объекте Resource, на который ссылается Item, родительский для Parameters. Символы, представленные как значения, объявленные для Parameters, должны быть URI-закодированы, как определено в RFC 3986. Область параметров ограничивается его родительским объектом

6.11.12 Класс Structure

В таблице 33 определен класс Structure в информационной модели упаковки контента (CPIM).

Таблица 33 — Определение класса Structure

Дескриптор	Определение
Class name (Имя класса)	Structure
Class type (Тип класса)	Characteristic
Data type (Тип данных)	String
Value space (Область значений)	См. описание ниже
Multiplicity (Кратность)	0..1
Parents (Родители)	Organization
Description (Описание)	Объект Structure описывает конкретный способ, которым объекты Item связаны друг с другом внутри объекта Organization. Область действия характеристики Structure ограничивается родительским Organization.

Окончание таблицы 33

Дескриптор	Определение
Description (Описание)	Область значения для Structure включает термины, одобренные Global Learning Consortium, Inc., и доступна для общественности в виде контролируемого списка (по умолчанию список словаря [19], [20] можно найти по адресу: gost.e-ecolog.ru). Синтаксис и семантика принятого перечня терминов должны поддерживаться всеми компонентами программного обеспечения, реализующими эту информационную модель. Область значений Structure может быть расширена. Такое расширение выражений может быть создано и использовано только при отсутствии утвержденного значения, удовлетворяющего сформулированным потребностям реализующего сообщества, определенной формы набора объектов Item в заданной Organization. Расширенные выражения, используемые в качестве значения Structure, должны соответствовать правилу синтаксиса URI, определенному в разделе 3 RFC 3986. Синтаксис для расширяемых терминов области значения Structure должен раскрывать источник термина как URI и сам термин как фрагмент URI: scheme://authority/hierarchy#term

6.11.13 Класс Type

В таблице 34 определен класс Type в информационной модели упаковки контента (CPIM).

Таблица 34 — Определение класса Type

Дескриптор	Определение
Class name (Имя класса)	Type
Class type (Тип класса)	Characteristic
Data type (Тип данных)	String
Value space (Область значений)	См. описание ниже
Multiplicity (Кратность)	1..1
Parents (Родители)	Resource
Description (Описание)	Объект Type обеспечивает термин, ключевое слово или фразу, которые указывают тип ресурса, описываемого родительским объектом Resource. Область значений для Type включает термины, одобренные ИСО/МЭК СТК1/ПК36 и IMS Global Learning Consortium, Inc., и доступна для общественности в виде контролируемого списка. Синтаксис и семантика утвержденного набора терминов должны поддерживаться всеми программными компонентами, реализующими эту информационную модель. Область значений для Type может быть распространена за пределы контролируемого списка. Такое расширение выражений может быть создано и использоваться только при отсутствии утвержденных ИСО/МЭК СТК1/ПК36 и IMS значений, удовлетворяющих сформированным потребностям реализующего сообщества. Расширенные выражения, используемые в качестве значения для Type, должны соответствовать правилу синтаксиса URI, определенному в разделе 3 из RFC 3986. Синтаксис для расширяемых терминов области значения Type должен раскрывать источник термина как URI и сам термин как фрагмент URI: scheme://authority/hierarchy#term. Область применения характеристики Type ограничивается родительским объектом. В частности, семантика, выраженная в значение для Type, не передается или «не наследуется» путем использования ссылки Resource.Dependency {IdentifierRef}

6.11.14 Класс Version

В таблице 35 определен класс Version в информационной модели упаковки контента (CPIM).

Таблица 35 — Определение класса Version

Дескриптор	Определение
Class name (Имя класса)	Version
Class type (Тип класса)	Characteristic
Data type (Тип данных)	String
Value space (Область значений)	Набор печатных символов из [15]
Multiplicity (Кратность)	0..1
Parents (Родители)	Manifest
Description (Описание)	Объект Version содержит значение, которое означает особую версию объекта Manifest. Когда эта характеристика декларируется для Manifest, являющегося дочерним объектом InterchangePackage, значение представляет собой версию документа манифеста. Когда эта характеристика объявлена для любого дочернего манифеста, она относится только к данному Manifest, а не ко всему документу, в котором этот дочерний манифест расположен. Синтаксис информации о версиях не определяется в настоящем стандарте. Исполнители имеют право использовать любые синтаксические представления, которые отвечают их потребностям. Область применения Version ограничена его родительским объектом

7 Соответствия

Этот раздел посвящен соответствию предполагаемых систем или приложений, которые подготавливают или обрабатывают экземпляры пакетов обмена. При реализации требований, описанных настоящей информационной моделью, требования по использованию определенных языков программирования или эвристических наборов для разработчиков приложений не предъявляются.

7.1 Формулировки

В настоящем стандарте формулировку «должен» следует интерпретировать как требование для применения; формулировка «не должен» интерпретируется как запрет. Формулировки «следует», «не следует» и «может» должны интерпретироваться, как описано в RFC 2119 (1997) [21].

7.2 Экземпляры пакетов обмена

Соответствующий экземпляр пакета обмена по настоящему стандарту должен соответствовать информационной модели, описанной в разделе 6.

7.3 Считыватели пакетов

Соответствующий считыватель пакетов обмена должен читать и интерпретировать пакеты обмена, как определено в разделе 6.

7.4 Записыватели пакетов

Соответствующий записыватель пакетов обмена должен записывать пакеты обмена, которые соответствуют информационной модели, описанной в разделе 6.

7.5 Расширения (расширение соответствий)

Соответствующий записыватель пакетов обмена должен удостовериться в том, что класс из расширенного пространства имен совпадает с типом информации целевой точкой расширения в информационной модели упаковки контента (CPIM). Семантика объекта Extension должна оставаться в пределах сферы семантики его родительского объекта в этой информационной модели. Семантика объекта

Extension не должна отменять или переопределять семантику любого родительского объекта, определенного в этой информационной модели.

Соответствующий считыватель пакетов не несет никакой ответственности за обеспечение того, что семантика любого класса расширения сохраняет или расширяет тип контейнера или значение объекта в этом документе.

Соответствующий считыватель пакетов не должен проверять обоснованность любых символьных строк, выраженных в расширении дочерних File или File.Metadata, которые представлены в ссылочных файлах.

Соответствующий считыватель пакетов не должен интерпретировать и обрабатывать содержание классов расширения. Если считыватель пакетов не обрабатывает объект расширения, то он должен игнорировать объект расширения без влияния на обработку оставшейся части пакета обмена.

Приложение А
(справочное)

Первоисточник стандарта

Название	IMS Content Packaging Information Model
Редакторы	Colin Smythe (IMS), Boyd Nielsen (независимый эксперт)
Сопредседатели	Wilbert Kraan (JISC/CETIS), Jan Poston Day (Blackboard), Nigel Ward (DEST)
Версия	v1.2 (CM/DN проект v2.0)
Дата версии	1 марта 2007 г.
Статус	Contributing Member / Developers Network Draft v2.0
Резюме	Настоящий документ описывает объекты IMS информационной модели упаковки контента v1.2, их отношение и назначение. Он также определяет выбранное поведение программных компонентов, которые создают или обрабатывают упаковки контента IMS и документы IMS Manifest.
Информация о пересмотре	4 декабря 2006 г.
Обновление документа	Этот стандарт основан на документе IMS GLC Content Packaging standard. IMS GLC оказывает постоянную поддержку по использованию, профилированию и совершенствованию стандартов по упаковке контента по адресу: gost.e-ecolog.ru

Приложение В
(справочное)

Подтверждение прав интеллектуальной собственности

«UML», «Unified Modeling Language», «OMG» и «Object Management Group» являются зарегистрированными товарными знаками компании Object Management Group, Inc в Соединенных Штатах и/или других странах.

«Unicode» является зарегистрированным товарным знаком компании Unicode, Inc.

«W3C» является товарным знаком (зарегистрирован во многих странах) консорциума World Wide Web; знаки W3C зарегистрированы и находятся во владении его базовых учреждений MIT, INRIA и Keio.

«XML», «HTML», «XHTML» являются торговыми марками W3C; знаки W3C зарегистрированы и находятся во владении его базовых учреждений MIT, INRIA и Keio.

«SCORM» является зарегистрированной торговой маркой компании Advanced Distributed Learning Initiative, Office of the Deputy Under Secretary of Defense (Readiness), Readiness and Training, 1E525, 4000 Defense Pentagon, Washington, D.C. 20301.

Приложение ДА
(справочное)

Сопоставление структуры настоящего стандарта со структурой примененного в нем международного стандарта

Таблица ДА.1

Структура настоящего стандарта	Структура международного стандарта ISO/IEC 12785-1:2009
1 Область применения	1 Область применения
2 Нормативные ссылки	2 Нормативные ссылки
3 Термины и определения	3 Термины и определения
4 Сокращения	4 Сокращения
5 Концептуальная модель упаковки контента (CPCM)	5 Концептуальная модель упаковки контента (CPCM)
6 Описание классов и требований к взаимосвязи	6 Описание классов и требований к взаимосвязи
7 Соответствие	7 Соответствие
Приложение А (справочное) Первоисточник стандарта	Приложение А (справочное) Первоисточник стандарта
Приложение В (справочное) Подтверждение прав интеллектуальной собственности	Приложение В (справочное) Подтверждение прав интеллектуальной собственности
—	Приложение С (справочное) Взаимосвязь с другими стандартами и спецификациями
—	Приложение D (справочное) Альтернативное представление файла манифеста
Приложение ДА (справочное) Сопоставление структуры настоящего стандарта со структурой примененного в нем международного стандарта	—
Библиография	Библиография

Библиография

[1] IMS GLC Content Packaging Information Model (Упаковка контента. Информационная модель) Version 1.2 CM/DN Draft v2.0, C. Smythe, B. Nielsen, IMS GLC, Inc., March 2007

[2] IMS GLC Content Packaging XML Binding (Упаковка контента. XML привязка) Version 1.2 CM/DN Draft 2.0, B. Nielsen, W. Kraan, J. Day and C. Smythe, IMS GLC, Inc., March 2007

[3] IMS GLC Content Packaging Best Practices and Implementation Guide (Упаковка контента. Лучшие практики и руководство по внедрению) Version 1.2 CM/DN Draft 2.0, B. Nielsen, W. Kraan, J. Day and C. Smythe, IMS GLC, Inc., March 2007

[4] IMS GLC Content Packaging Primer (Упаковка контента. Учебник) Version 1.2 CM/DN Draft v2.0, B. Nielsen, W. Kraan, N. Ward, IMS GLC, Inc., March 2007

[5] Sharable Content Object Reference Model (Эталонная модель распределенного объекта контента), SCORM 2004, second edition, Advanced Distributed Learning, July 2004

[6] IMS Content Packaging Information Model (Упаковка контента. Информационная модель) v1.1.4, C. Smythe, A. Jackl, IMS GLC, Inc., October 2004

[7] IMS Content Packaging XML Binding (Упаковка контента. XML-привязка) v1.1.4, C. Smythe, A. Jackl, IMS GLC, Inc., October 2004

[8] W3C Recommendation, Extensible Markup Language (XML) 1.0 (Second Edition) [Рекомендации W3C, расширяемый язык разметки (XML) 1.0 (Второе издание)], 6 October 2000

[9] W3C Recommendation, Namespaces in XML 1.0 (Second Edition) [Рекомендации W3C. Пространство имен в XML 1.0 (Второе издание)], 16 August 2006

[10] IETF RFC 1951 (1996) DEFLATE Compressed Data Format Specification version 1.3 (Спецификация формата сжатия DEFLATE версия 1.3)

[11] IETF RFC 3986 (2005) Uniform Resource Identifier (URI): Generic Syntax [Универсальный идентификатор ресурса (URI): Общий синтаксис]

[12] IMS GLC Specification Development Note 7: UML Profile for Platform Independent Model Descriptions of Specifications for Data Models (Спецификация развития, Примечание 7: UML-профиль для платформы независимой модели описания спецификаций для моделей данных) C.Smythe, v1.0, IMS GLC, October 2006

[13] IETF RFC 2234 (1997) Augmented BNF for Syntax Specifications: ABNF [Расширенная спецификация синтаксиса Бэкуса — Наура (ABNF)]

[14] IETF RFC 2732 (1999) Format for Literal IPv6 Addresses in URL's (Формат для буквенного адреса IPv6 в URL)

[15] ISO/IEC 10646:2003 Information technology — Universal Multiple-Octet Coded Character Set (UCS) [Информационные технологии. Универсальный многооктетный набор кодированных символов (UCS)]

[16] ANSI/IEEE 1484.12.1:2002 Standard for Learning Object Metadata (Стандарт для метаданных образовательных объектов)

[17] ISO 639-2:1998 Codes for the representation of names of languages — Part 2: Alpha-3 code (Коды для представления названий языков. Часть 2. Трехбуквенный код)

[18] ISO 3166-1:1997 Codes for the representation of names of countries and their subdivisions — Part 1: Country codes (Коды для представления названий стран и единиц их административно-территориального деления. Часть 1. Коды стран)

[19] IMS Vocabulary Definition Exchange Best Practice and Implementation Guide (Словарь определение схемы обмена лучшими практиками и руководство по внедрению), Version 1.0 Final Specification, A. Cooper, IMS GLC, Inc., 2005

[20] IMS GLC Specification Development Note 11: Vocabulary Definition, Registration, and Maintenance Procedures (Спецификация развития. Примечание 11: Словарь определения схемы регистрации и процедур технического обслуживания), v1.0

[21] IETF RFC 2119 (1997) Keywords for use in RFCs to Indicate Requirement Levels (Ключевые слова для обозначения уровня требований в RFC)

УДК 681.118.087:006.354 МКС 35.240.99 П80 MOD

Ключевые слова: упаковка контента, информационная модель, пакет обмена, манифест

Редактор *Н.В. Таланова*
Технический редактор *В.Н. Прусакова*
Корректор *Е.Р. Ароян*
Компьютерная верстка *Ю.В. Поповой*

Сдано в набор 16.01.2019. Подписано в печать 30.01.2019. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 5,58. Уч.-изд. л. 5,07.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

ИД «Юриспруденция», 115419, Москва, ул. Орджоникидзе, 11.
gost.e-ecolog.ru gost.e-ecolog.ru

Создано в единичном исполнении ФГУП «СТАНДАРТИНФОРМ»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru