



EBU – TECH 3351

# EBU Class Conceptual Data Model (CCDM)

## **Внимание!**

Данный перевод **НЕ** является официальной версией статьи и может содержать отдельные неточности.

## Концептуальная модель данных классов (CCDM) EBU

**Источник: MIM**

**Статус: Спецификация v.1.0**

**Женева  
Май 2012**

## Система обозначений

Настоящий документ содержит как **нормативный**, так и **информативный** текст.

Весь текст является нормативным, кроме Введения, разделов, отмеченных как «информативные», или отдельных параграфов, начинающихся с «Примечания».

**Нормативный** текст описывает обязательные или непреложные элементы. Он содержит ключевые слова «должен», «следует» или «можно», определяемые следующим образом:

«Должен» или «не должен»: (обязательно)      Указывает требования, которые нужно строго соблюдать и от которых не допускается отклонений для соответствия документу.

«Следует» или «не следует»: (рекомендуется)      Указывает, что один из нескольких вариантов рекомендуется как особенно подходящий, не упоминая и не исключая других.

ИЛИ что определенный ход действий предпочтителен, но не обязателен.

ИЛИ что (в отрицательной форме) определенный вариант или ход действий не рекомендуется, но не запрещается.

«Может» или «не вправе»: (опционально)      Указывает ход действий, допустимый в рамках документа.

**По умолчанию** означает обязательные (в фразах, содержащих «должен») или рекомендуемые (в фразах, содержащих «следует») предустановки, которые могут быть опционально изменены пользователем или иметь другие опции в продвинутых приложениях. Обязательные установки по умолчанию должны поддерживаться. Поддержка рекомендуемых установок предпочтительна, но не обязательна.

**Информативный** текст потенциально полезен для пользователя, но не обязателен и может быть исключен, изменен или дополнен, не влияя на нормативный текст. Информативный текст не содержит ключевых слов соответствия.

Совместимая реализация включает все обязательные условия («должен») и все рекомендуемые условия («следует») в случае их реализации. Совместимая реализация не требует реализации опциональных условий («можно»).

## Сроки и условия использования

EBU CSDM свободно доступен для всех, но необходимо учитывать следующее:

© EBU 2012.

ПЕРЕРАСПРЕДЕЛЕНИЕ И ИСПОЛЬЗОВАНИЕ ДАННОЙ СПЕЦИФИКАЦИИ И СВЯЗАННЫХ С НЕЙ РЕСУРСОВ РАЗРЕШАЕТСЯ ПРИ УСЛОВИИ СОБЛЮДЕНИЯ СЛЕДУЮЩИХ УСЛОВИЙ:

ПЕРЕРАСПРЕДЕЛЕНИЯ ДОЛЖНЫ СОХРАНЯТЬ ВЫШЕУКАЗАННОЕ УВЕДОМЛЕНИЕ ОБ АВТОРСКОМ ПРАВЕ, ДАННЫЙ СПИСОК УСЛОВИЙ И СЛЕДУЮЩЕЕ ЗАЯВЛЕНИЕ ОБ ОГРАНИЧЕНИИ ОТВЕТСТВЕННОСТИ В ДОКУМЕНТАЦИИ И/ИЛИ ДРУГИХ ПЕРЕДАВАЕМЫХ МАТЕРИАЛАХ;

НИ ИМЯ EBU, НИ ИМЕНА ЕГО ВКЛАДЧИКОВ НЕЛЬЗЯ ИСПОЛЬЗОВАТЬ ДЛЯ ПОДДЕРЖКИ ИЛИ ПРОДВИЖЕНИЯ ПРОДУКТОВ, ПОЛУЧЕННЫХ ИЗ ДАННОЙ СПЕЦИФИКАЦИИ И СООТВЕТСТВУЮЩИХ РЕСУРСОВ, БЕЗ ОСОБОГО ПРЕДВАРИТЕЛЬНОГО ПИСЬМЕННОГО РАЗРЕШЕНИЯ.

ОГРАНИЧЕНИЕ ОТВЕТСТВЕННОСТИ: ДАННАЯ СПЕЦИФИКАЦИЯ И СВЯЗАННЫЕ С НЕЙ РЕСУРСЫ ПРЕДОСТАВЛЕНЫ ВЛАДЕЛЬЦЕМ АВТОРСКИХ ПРАВ «КАК ЕСТЬ», И ЛЮБЫЕ ЯВНО ВЫРАЖЕННЫЕ ИЛИ ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ, ПОДРАЗУМЕВАЕМЫЕ ГАРАНТИИ ТОВАРНОГО СОСТОЯНИЯ И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ, НЕ ПРИЗНАЮТСЯ. ВЛАДЕЛЕЦ АВТОРСКИХ ПРАВ НИ В КОЕМ СЛУЧАЕ НЕ ДОЛЖЕН НЕСТИ ОТВЕТСТВЕННОСТЬ ЗА ЛЮБЫЕ ПРЯМЫЕ, НЕПРЯМЫЕ, СЛУЧАЙНЫЕ, СПЕЦИАЛЬНЫЕ, ШТРАФНЫЕ ИЛИ КОСВЕННЫЕ УБЫТКИ (ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ, ПОСТАВКУ ЗАМЕНИТЕЛЕЙ ТОВАРОВ И УСЛУГ; УТРАТУ ЭКСПЛУАТАЦИОННЫХ КАЧЕСТВ, ДАННЫХ ИЛИ ПРИБЫЛИ; ИЛИ ПРЕРЫВАНИЯ ПРОИЗВОДСТВА), ЧЕМ БЫ ОНИ НЕ БЫЛИ ВЫЗВАНЫ И ПО ЛЮБОЙ ТЕОРИИ ОТВЕТСТВЕННОСТИ, БУДЬ ОНА ПО КОНТРАКТУ, ОБЪЕКТИВНОЙ ОТВЕТСТВЕННОСТЬЮ ИЛИ ДЕЛИКТОМ (ВКЛЮЧАЯ ХАЛАТНОСТЬ ИЛИ ИНОЕ) В РЕЗУЛЬТАТЕ ИСПОЛЬЗОВАНИЯ ЭТОГО [ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ], ДАЖЕ ПРИ УВЕДОМЛЕНИИ О ВОЗМОЖНОСТИ ПОДОБНОГО УБЫТКА.

# Содержание

1. Введение .....	4
<b>2. Область рассмотрения .....</b>	<b>4</b>
2.1 Общие задачи .....	4
2.2 Обоснование .....	4
<b>3. Концептуальная модель данных классов .....</b>	<b>4</b>
3.1 Основные принципы .....	4
3.2 Классы, отношения и свойства .....	6
3.2.1 Область логистики .....	6
3.2.2 Область редактирования .....	7
3.2.3 Область производства .....	10
3.2.4 Другие классы .....	14
3.2.5 Область распространения .....	17
<b>4. Принципы реализации / Вопросы и ответы .....</b>	<b>20</b>
4.1 Общие замечания .....	20
4.2 Как приспособить мою модель?.....	20
4.3 Еще вопросы?.....	20
<b>5. Соответствие CCDM .....</b>	<b>20</b>
<b>6. Зона скачивания .....</b>	<b>21</b>
<b>7. Режим лицензирования .....</b>	<b>21</b>
<b>8. Обслуживание .....</b>	<b>21</b>
<b>9. Полезные ссылки .....</b>	<b>21</b>
<b>Приложение А: Шаблон преобразования EBU CCDM и пример .....</b>	<b>22</b>
<b>А.1 Введение .....</b>	<b>22</b>
<b>А.2 Инструкции по использованию шаблона .....</b>	<b>22</b>
<b>Приложение В: Онтология EBU CCDM .....</b>	<b>23</b>

## Концептуальная модель данных классов (CCDM) EBU

Комитет EBU	Первый выпуск	Переработка	Переиздание
MIM	2012		

**Ключевые слова:** Класс, Модель, Метаданные, Бизнес, Объект, Радио, Телевидение, Производство, SOA, семантическая паутина, связанные данные

### 1. Введение

Концептуальная модель данных классов (CCDM) EBU – это общая структура. Пользователи с различными нуждами приглашаются к дальнейшей идентификации классов медиа и их отношений согласно своим потребностям.

CCDM специально разработана как минимальный и гибкий набор классов для широкого спектра вещательных приложений, включая архивы, обмен и производство, ориентированное на медиа услуги, семантическую паутину и связанные данные.

Информация о деятельности EBU в области метаданных содержится на сайте EBU TECHNICAL (<http://tech.ebu.ch/metadata>).

Это версия 1.0 “CCDM”.

### 2. Область рассмотрения

#### 2.1 Общие задачи

Концептуальная модель данных классов (CCDM) EBU – общая структура, помогающая пользователям с различными нуждами идентифицировать классы медиа и их отношения.

CCDM разработана как минимальный и гибкий набор классов для широкого спектра вещательных приложений, включая, но не ограничиваясь, архивы, обмен, производство, ориентированное на медиа услуги, вещание, передачу по Интернет, моделирование семантической паутины и связанные открытые данные (LOD).

Данная спецификация не является спецификацией метаданных. Элементы метаданных и типы данных (кроме отношений между классами) ориентированы. Описательные и технические метаданные для описания классов CCDM определены в Tech 3293 (набор метаданных EBUCore) или в других спецификациях метаданных (например, TV-Anytime).

#### 2.2 Обоснование

Провайдером контента и вещателям жизненно важно иметь четко определенную модель классов. Это необходимый шаг вперед к:

- Большему пониманию бизнеса и рабочих процессов;
- Оптимизации процессов с более легким и надежным обменом данными;
- Более простому и рационализированному описанию классов медиа;
- Более простой реализации архитектур производства, ориентированного на медиа услуги;
- Принятию новых моделей управления информацией, таких как семантическая паутина и связанные данные (обогащение, улучшение поиска и доступности).

CCDM разработан для того, чтобы позволить реализаторам адаптировать имена классов и их отношения в соответствии со своими нуждами. Призываем каждую организацию определить свои потребности путем анализа и создать собственную модель, начав со структуры CCDM.

Примеры реализации CCDM даны в **Приложении А**.

### 3. Концептуальная модель данных классов

#### 3.1 Основные принципы

EBU CCDM состоит из следующего:

- **Классы:** напрямую относятся к медиа (например, к программе, части, клипу, дорожке) или связаны с ними (например, с лицом, местом).
  - *Примечание: эквивалент понятия класса, используемого в моделировании семантической паутины, в определенных проектах также называется "Business Objects".*
- **Отношения:** связывание классов (например, 'Programme hasContributor Person').
  - *Примечание: эквивалент понятия objectProperties, используемого в моделировании семантической паутины.*
- **Свойства:** определение внутренних характеристик классов (например, 'bitrate', выраженной как целое число, или 'name', выраженной в виде строки).
  - *Примечание: эквивалент понятия dataProperties, используемого в моделировании семантической паутины.*

Как показано на Рис. 1, эта модель определяется вокруг четырех основных областей:

1. Область логистики – где происходит планирование, заказ и управление правами. Центральный класс области логистики – это фонд, который устанавливает связь EditorialObject или MediaResource с информацией о его интеллектуальной собственности (IPRights). Предварительная логистика состоит в заказе производства фонда через изготовление объекта в производстве, который заказывается через производственный заказ.
2. Область редактирования – где концепция, определенная в заказе, реализуется путем виртуальной компоновки и редактирования контента. Центральный класс области редактирования – EditorialObject.
3. Область производства – где производственные заказы реализуются путем сбора MediaResource (создания/загрузки или приобретения материала) согласно производственному плану. MediaResource реализуется в форме файлов или пакетов сущности в одном или более форматах. Изготовление заказанного объекта состоит в реализации EditorialObject через конкретизацию и компоновку MediaResources / Essences.
4. Область распространения определяет общую структуру для кросс-медийного распространения MediaResource. Организация вводит службу, которая управляет PublicationChannels и PublicationEvents. PublicationEvent описывается EditorialObject. Он заказывается через заказ на закупку, связанный с фондом, предназначенным для распространения. PublicationEvent публикует MediaResource как сущность в формате, подходящем для средства передачи.

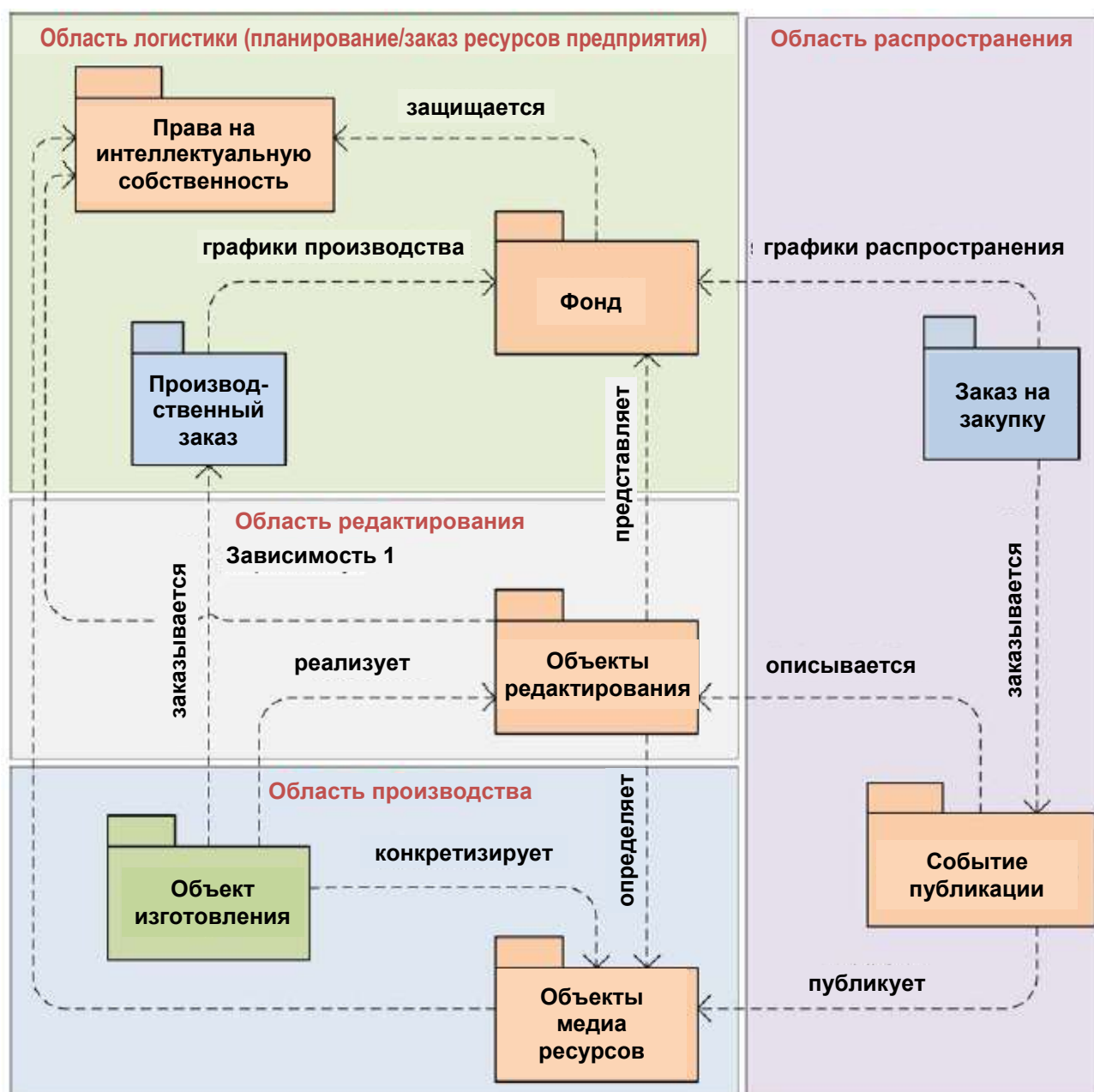


Рис. 1: Области CCDM

EBU CCDM разработан для того, чтобы позволить пользователям адаптировать имена классов и отношений в соответствии со своими нуждами. Например, класс 'EditorialObject' может быть типа 'programme' или 'item' (или иначе). Класс 'Group' может быть типа «цикл», «сериал» или «сезон» и т.д. Определение соответствующих свойств остается за пользователем. Базовый набор свойств предлагается в EBU Tech 3293, EBUCore или в других спецификациях метаданных (например, TV-Anytime).

Соответствие определенной реализации модели EBU CCDM определяется пользователями в их сферах применения и взаимодействия.

## 3.2 Классы, отношения и свойства

### 3.2.1 Область логистики

Область логистики – это область, в которой концепция планируется и заказывается вместе с необходимыми для ее реализации ресурсами. Это также область, в которой интеллектуальная собственность и права управляются и связываются с MediaResource или EditorialObject, чтобы стать фондом.

Центральный класс области логистики – фонд.

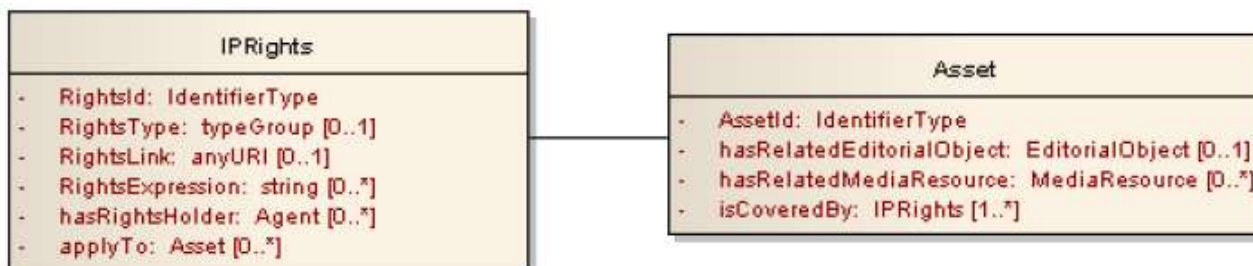


Рис. 2: Фонд

#### 3.2.1.1 Asset (Фонд)

Определение:

Класс "Asset" определяет связь между MediaResource и IPRights.

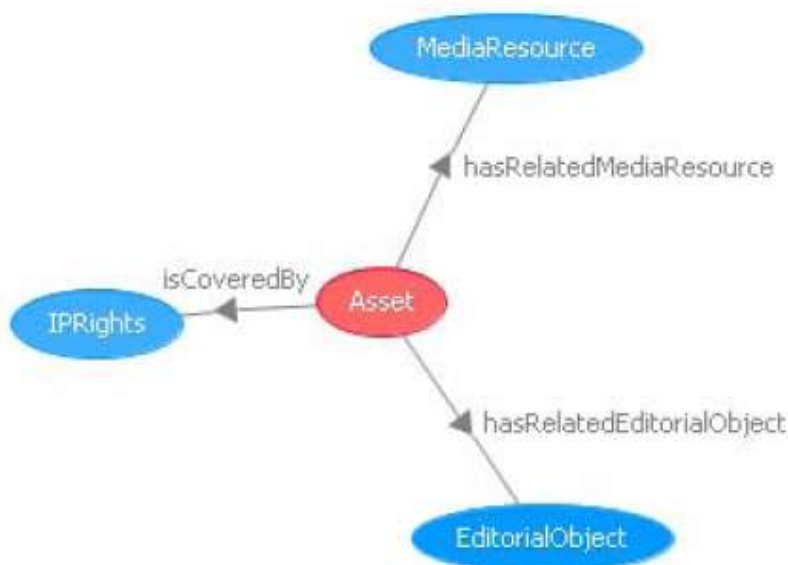


Рис. 3: График отношений Asset.

Отношения класса	
hasRelatedMediaResource	Указатель на MediaResource, который фонд связывает со своим IPRights.
hasRelatedEditorialObject	Указатель на EditorialObject, который фонд связывает со своим IPRights
isCoveredBy	Указатель на IPRights, связанный с MediaResource
И т.д.	С фондом могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
AssetID	Идентификатор, связанный с фондом
И т.д.	С фондом могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.1.2 IPRights (Права на интеллектуальную собственность)

Определение:

Класс "IPRights" определяет права, связанные с MediaResource через определение фонда.

Отношения класса	
applyTo	Указатель на Asset, т.е. MediaResource, к которому применяется IPRights.
hasRightsHolder	Указатель на Agent, являющийся держателем IPRights.
И т.д.	С IPRights могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore
Свойства класса	
RightsID	Идентификатор, связанный с Rights.
RightsType	Тип, связанный с IPRights, например, лицензионные условия.
RightsLink	Ссылка, например, на веб-ресурс, где можно найти правовые условия.
RightsExpression	Права, выраженные свободным текстом.
И т.д.	С IPRights могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.2 Область редактирования

Область редактирования – это область, в которой определяется заказанная концепция перед изготовлением и распространением.

Центральный класс области редактирования – EditorialObject.

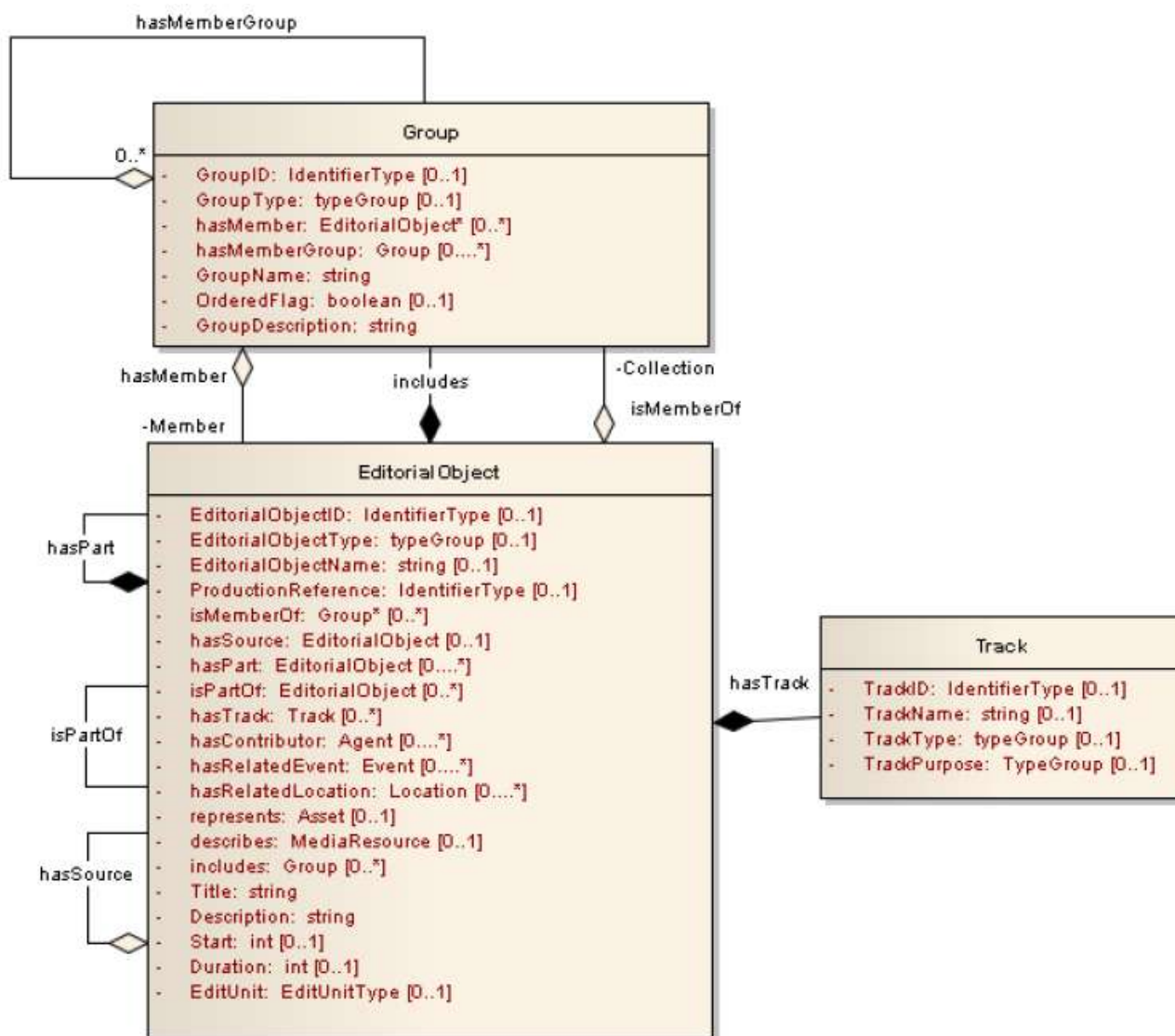


Рис. 4: EditorialObject

### 3.2.2.1 Group (Группа)

#### Определение:

Класс "Group" – общая концепция для определения группировок EditorialObjects. "Group" также может быть образована другими "Group" или быть членом "Group".

GroupType используется для уточнения определения группы согласно нуждам пользователя.

#### Примеры:

Примеры типов группы: цикл, сериал, компиляция, коллекция, группа элементов, блок элементов и т.д.

Отношения класса	
hasMember	Список классов, являющихся членами группы.
hasMemberGroup	Список групп, являющихся членами группы.
И т.д.	С Group могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
GroupID	Идентификатор, связанный с Group
GroupType	Тип Group, например, цикл, сериал, группа элементов, блок элементов, коллекция, компиляция.
GroupName	Имя, связанное с Group.
GroupDescription	Описание Group.
OrderedFlag	При значении 'true' – флажок, указывающий, что Group упорядочен (например, членство подлежит строгой последовательности, как эпизоды в сериале).
И т.д.	С Group могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.2.2 EditorialObject (Объект редактирования)

#### Определение:

Класс "EditorialObject" преобразует заказанную концепцию в редакторское определение MediaResource перед изготовлением (в области производства) и распространением (в области распространения). "EditorialObject" – набор описательных метаданных, суммирующий, например, монтажные решения.

"EditorialObject" также может быть частью другого "EditorialObject", который определяется своим начальным временем и хронометражем.

#### Примеры:

Программа, элемент, план, часть, глава, сегмент и т.д.

EditorialObject, состоящий из частей (которые тоже определяются как EditorialObjects с началом и хронометражем), можно определить следующим образом:

- Чистый план ЕО представляет интервью длиной в 1 минуту. Последние 10 секунд этого плана (50">60") будут использованы в сюжете новостей. Этот сюжет является частью выпуска новостей, включающего и другие сюжеты.

[EO News Broadcast] ----- hasPart ----- [EO NewsItem] ----- hasSource ---- [EO clean capture]

или

[EO News Broadcast] ----- hasPart ----- [EO NewsItem] ----- isPartOf ---- [EO clean capture]

Сюжеты новостей редакторски связаны с выпуском новостей с помощью свойства hasPart. Сюжет может содержать свойства 'Start' / 'Duration', указывающие, в каком месте выпуска он появляется. Чистый план редакторски связан с сюжетом с помощью 'hasSource' и также использует время начала и хронометраж для идентификации используемой порции контента.



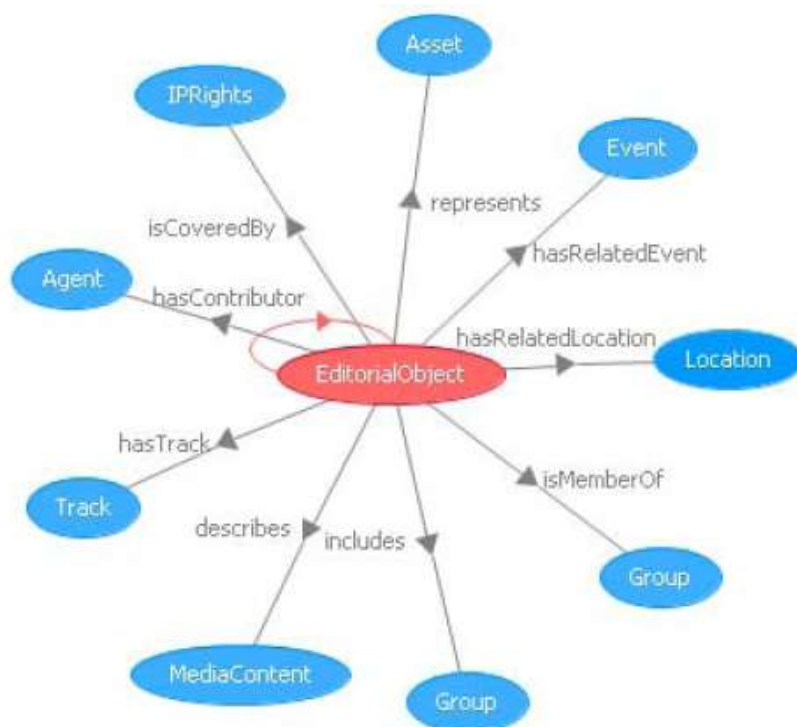


Рис. 5: График отношений EditorialObject

Отношения класса	
isMemberOf	Список групп, членом которых является EditorialObject.
hasSource	Отношение для идентификации другого EditorialObject, из которого упоминается EditorialObject, включая его начальную позицию и хронометраж в исходном EditorialObject.
hasPart	Список EditorialObjects, образующих EditorialObject, например, в виде частей, определяемых по начальной позиции и хронометражу.
isPartOf	Отношение для идентификации всех остальных EditorialObjects, в которых используется этот EditorialObject, определяемый по начальной позиции и хронометражу.
hasTrack	Список дорожек, составляющих EditorialObject.
hasContributor	Опционально, один (или более) Agent, внесший вклад в реализацию EditorialObject. Вклад характеризуется Agent Role. Agent не является классом не-медиа, описанным в другом разделе настоящего документа. Пользователь может добавлять отношения класса, например, hasCreator и т.д.
hasRelatedLocation	Опционально, один (или более) Location, связанный с EditorialObject, характеризующийся типом (например, документальный или художественный).
hasRelatedEvent	Опционально, один (или более) Event, связанный с EditorialObject, характеризующийся типом (например, спортивное событие / встреча).
represents	EditorialObject представляет Asset.
describes	EditorialObject определяет MediaResource.
includes	Опционально включает группу EditorialObjects, например, упорядоченную группу Parts
isCoveredBy	Для идентификации IPRights, применяемых к EditorialObject.
И т.д.	С EditorialObject могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
EditorialObjectType	Тип EditorialObject, например, Programme, Item.
EditorialObjectID	Опционально один (или более) идентификатор, присвоенный EditorialObject.
EditorialObjectName	Имя, присвоенное EditorialObject.
ProductionReference	Опционально одна (или более) ссылок на производство, заказавшее EditorialObject.
Title	Основное название, под которым известен EditorialObject.
Description	Опционально одно (или более) описание EditorialObject.
Start	Начальная точка части в EditorialObject.

Duration	Хронометраж части в EditorialObject.
EditUnit	Единица, используемая для выражения начала и хронометража.
И т.д.	С EditorialObject могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.2.3 Track (Дорожка)

Определение:

“Track” является частью EditorialObject и может состоять из любой комбинации аудио, видео и информационного контента. Расположение аудио дорожек определено в EBU R 123.

Примеры:

Примеры видео дорожек – различные углы камеры или дополнительная подписная дорожка.

Примеры аудио дорожек - стереопары, многоканальный звук, например, объемный, международный звук и т.д.

Примеры дорожек данных: вспомогательные данные, субтитры и т.д.

Отношения класса	
И т.д.	С Track могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
TrackID	Идентификатор, присвоенный Track.
TrackType	Тип Track, например, аудио, видео, данные.
TrackName	Имя, связанное с Track.
TrackPurpose	Назначение дорожки, например, угол просмотра, дублирование, субтитры, signing.
И т.д.	С Track могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.3 Область производства

Область производства представляет логическую модель или композицию MediaResource, созданного / приобретенного / модифицированного в производстве в одном или более форматах, и ссылки на соответствующую сущность.

Центральный класс в области производства – MediaResource.

MediaResource определяется EditorialObject (область редактирования) и может защищаться IPRights (область логистики). Сущность является отдельным представлением MediaResource и конкретизирует его в определенном формате. Сущность является подклассом MediaResource и наследует свойство hasFormat.

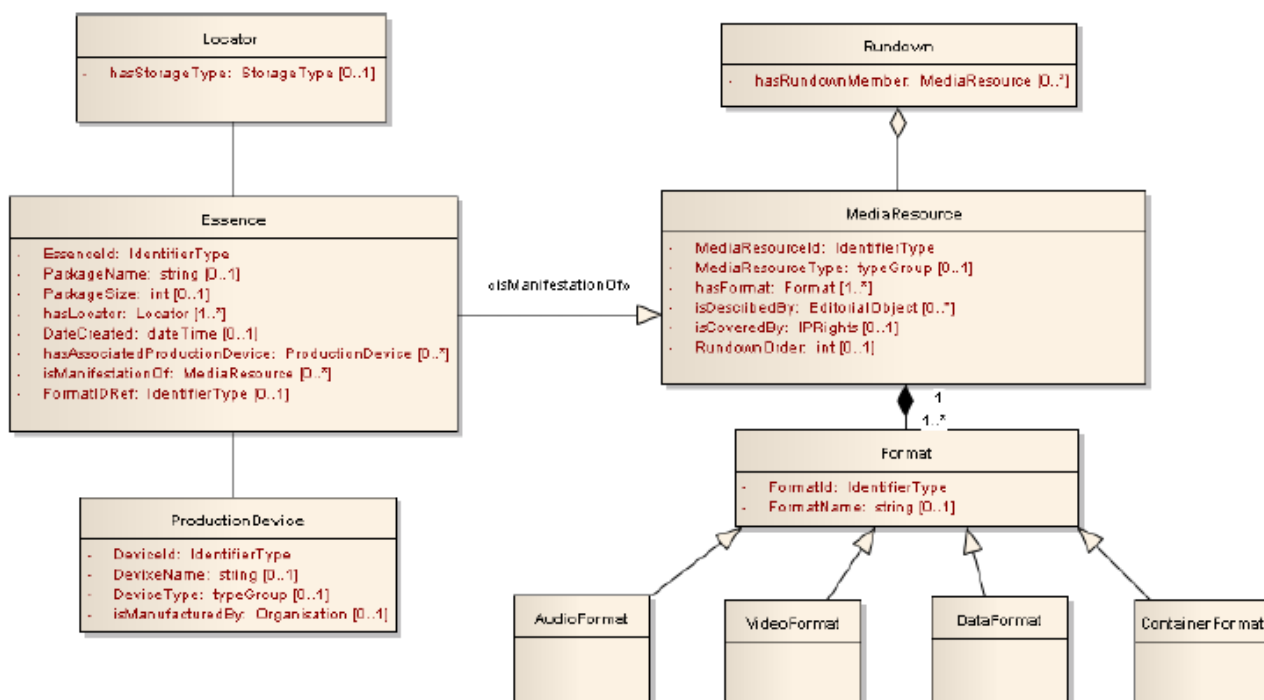


Рис. 6: MediaResource

### 3.2.3.1 MediaResource (Медиа ресурс)

#### Определение:

“MediaResource” заказывается для производства. Он определяется EditorialObject (область редактирования). Он может изготавливаться в одном или более форматах, конкретизируемых в одном или более экземпляров сущности. Он может быть представлен одной или более сущностями, например, в определенном формате для распространения определенными средствами передачи.

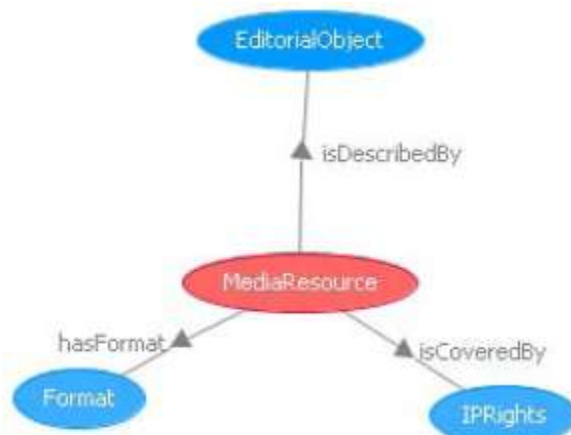


Рис. 7: График отношений MediaResource

Отношения класса	
hasFormat	Состав MediaResource. MediaResource может существовать в одном или более форматах.
isCoveredBy	MediaResource защищается IPRights.
isDescribedBy	MediaResource определяется EditorialObject.
И т.д.	С MediaResource могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
MediaResourceId	Уникальный идентификатор, например, UUID, UMID и т.д. Может генерироваться или присваиваться в производственном процессе или извлекаться из контента.
MediaResourceType	Свободный текст, содержащий дополнительную информацию делового уровня о типе контента или папке.
RunDownOrder	Порядковый номер, по которому MediaResource должен воспроизводиться по расписанию прогона.
И т.д.	С MediaResource могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.3.2 Rundown (Прогон)

#### Определение:

“Rundown” – группа MediaResource / Essence для публикации по порядку в расписании как одно PublicationEvent.

Отношения класса	
hasRundownMember	Для идентификации членов списка Rundown.
И т.д.	С Rundown могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
И т.д.	С MediaResource могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.3.3 Format (Формат)

#### Определение:

“Format” – структура технических метаданных. “Format” может определяться как композиция аудио, видео и / или информационных компонентов и описание их форматов. ContainerFormat определяет структуру файлов / пакетов сущности.

Пример:

Формат для аудио сущности будет определять формат аудио кодирования, частоту дискретизации и т.д.

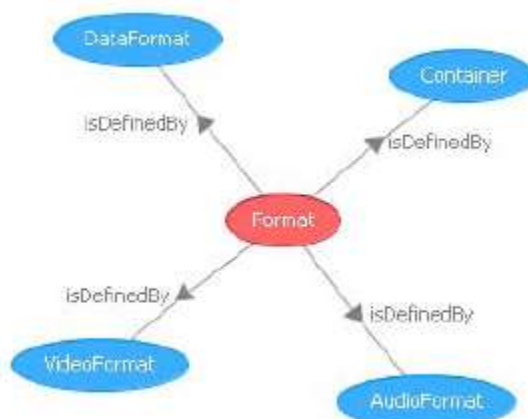


Рис. 8: График отношений Format

Отношения класса	
И т.д.	С Format могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
FormatId	Идентификатор, связанный с форматом.
FormatName	Имя, присвоенное формату.
isDefinedBy	Указатель на класс AudioFormat, используемый для перечня всех характеристик аудио сигнала. Подробнее см. 'audioFormat' в EBU Tech 3293, EBUCore.
isDefinedBy	Указатель на класс VideoFormat, используемый для перечня всех характеристик видео сигнала. Подробнее см. 'videoFormat' в EBU Tech 3293, EBUCore.
isDefinedBy	Указатель на класс DataFormat, используемый для перечня всех характеристик сигнала данных.
isDefinedBy	Указатель на класс ContainerFormat, используемый для перечня всех характеристик контейнера. Содержит информацию о формате контейнера / упаковки в дополнение к информации о кодировании потока в 'channel' (например, mp3, wave, Quicktime, Ogg...). Подробнее см. 'containerFormat' в EBU Tech 3293, EBUCore.
И т.д.	С Format могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.3.3.1 AudioFormat (Аудио формат)

Определение:

Класс для определений "AudioFormat" (например, формат кодирования, частота дискретизации).

Отношения класса	
И т.д.	С AudioFormat могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
И т.д.	С AudioFormat могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.3.3.2 VideoFormat (Видео формат)

Определение:

Класс для определений "VideoFormat" (например, формат кодирования, частота кадров).

Отношения класса	
И т.д.	С VideoFormat могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
И т.д.	С VideoFormat могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.3.3.3 DataFormat (Формат данных)

Определение:

Класс для определений "DataFormat" (например, формат субтитров).

Отношения класса	
И т.д.	С DataFormat могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
И т.д.	С DataFormat могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

**3.2.3.3.4 ContainerFormat (Формат контейнера)**Определение:

Класс для определений "ContainerFormat" (например, тип контейнера).

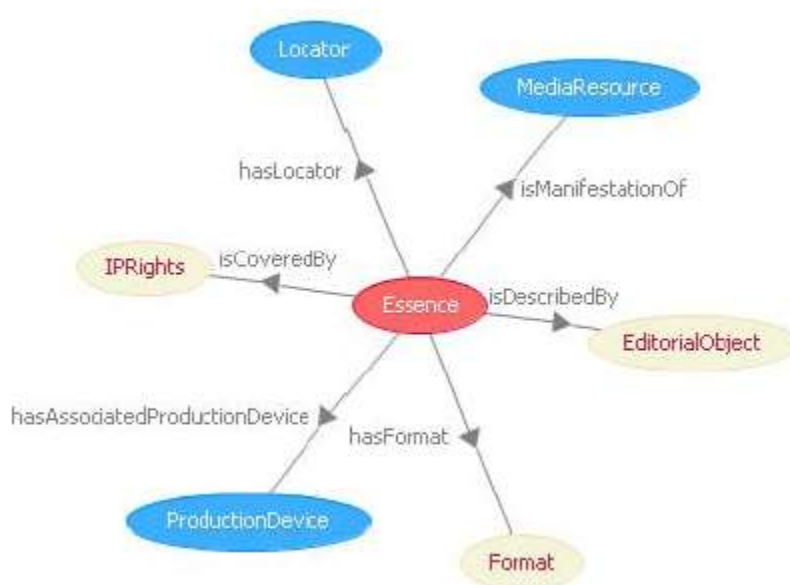
Отношения класса	
И т.д.	С ContainerFormat могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
И т.д.	С ContainerFormat могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

**3.2.3.4 Essence (Сущность)**Определение:

"Essence" – физическое представление MediaResource в определенном формате. "Essence" может быть в форме простого файла или сложных пакетов (например, передаваемых камерам различных брендов).

Примеры:

Файл AAC – пример аудио сущности. Файловая структура P2 (аудио, видеоклип, речь, иконка, прокси директории) – пример пакета.



**Рис. 9: График отношений Essence**

На Рис. 10 показаны все свойства, унаследованные от класса MediaResource. Обратите внимание на "hasFormat", позволяющий напрямую связывать Format с Essence.

Отношения класса	
has Associated ProductionDevice	Список ProductionDevices, используемых для производства всей или части сущности, представляющей MediaResource.
isManifestationOf	Сущность представляет MediaResource.
hasLocator	Ссылка на местоположение сущности.
И т.д.	С Essence могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Предполагаемые отношения класса	
isDescribedBy	Как подкласс MediaResource, Essence также описывается EditorialObject
hasFormat	Опциональная ссылка на FormatId, идентифицирующий формат, в котором реализована Essence.
isCoveredBy	Свойство для идентификации IPRights, связанных с Essence.
Свойства класса	
EssenceId	Идентификатор, связанный с фондом, например, UMID.
PackageName	Имя, связанное с простым файлом или сложным пакетом.

PackageSize	Размер файла или всех компонентов сложного пакета, в байтах. Хотя размер опционален, агенты должны указывать эту информацию, если она известна. Спецификации некоторых профилей могут требовать наличие этой информации.
DateCreated	Дата и время создания контента.
И т.д.	С Essence могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.3.5 ProductionDevice (Производственное оборудование)

Определение:

“ProductionDevice” – устройство, используемое в процессе создания Essence.

Примеры:

Типичный пример ProductionDevice – безленточный камкордер.

Отношения класса	
isManufacturedBy	Указатель на Organisation, изготовившую ProductionDevice.
Etc.	С ProductionDevice могут быть связаны другие отношения класса.
Свойства класса	
DeviceId	Идентификатор, связанный с ProductionDevice.
DeviceType	Тип ProductionDevice, например, камкордер.
DeviceName	Имя ProductionDevice.
И т.д.	С ProductionDevice могут быть связаны многие другие свойства. Примеры дополнительных свойств камкордера можно найти в EBU Tech 3349 (Acquisition Metadata).

### 3.2.3.6 Locator (Локатор)

Определение:

“Locator” указывает, где находится и доступна определенная сущность.

Примеры:

Адрес веб-сервера или ftp.

Отношения класса	
hasStorageType	Определение типа / структуры накопителя, где хранится сущность.
И т.д.	С Locator могут быть связаны другие отношения класса.
Свойства класса	
И т.д.	С Locator могут быть связаны другие свойства.

### 3.2.3.7 StorageType (Тип накопителя)

Определение:

“StorageType” – тип накопителя, на который указывает Locator. “StorageType” будет идентифицироваться, например, понятием из схемы классификации SKOS. Поэтому “StorageType” следует считать классом, т.е. подклассом понятия SKOS.

Примеры:

Contact может быть Distributor.

Отношения класса	
И т.д.	С StorageType могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
И т.д.	С StorageType могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

## 3.2.4 Другие классы

Следующий раздел вводит несколько дополнительных не-медиа классов, принадлежащих EBU CCDM.

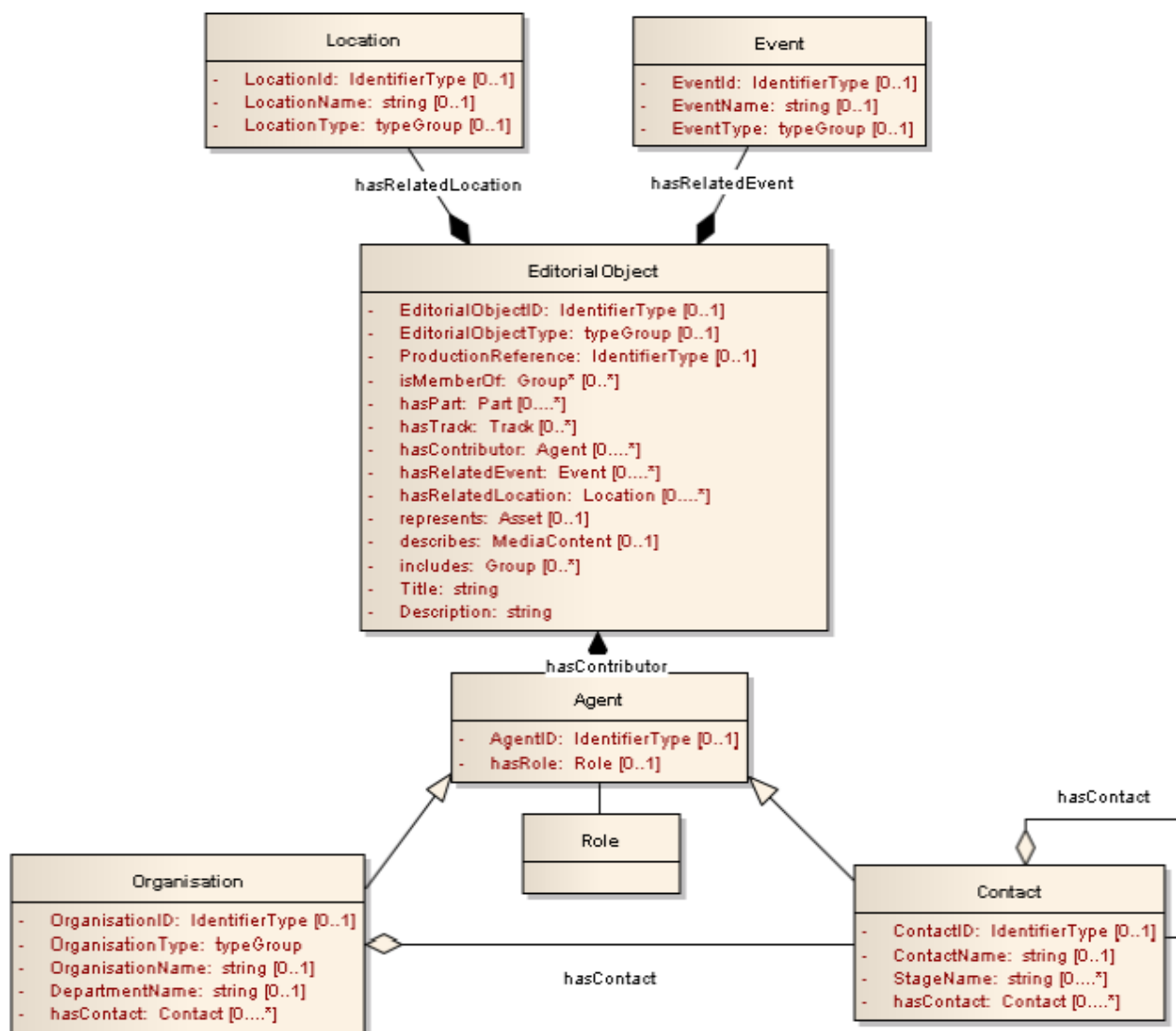


Рис. 10: Другие классы, связанные с медиа ресурсом

### 3.2.4.1 Agent (Агент)

Определение:

“Agent” – либо Contact/Person, либо Organisation, с которой связана роль, соответствующая вкладу “Agent” в реализацию MediaResource или EditorialObject.

Примеры:

Примеры Agent Role – «продюсер», «оператор» или «актер».

Отношения класса	
hasRole	Роль Agent. Роль определяет общую концепцию "hasContributor". Альтернативно, пользователь может добавить новый класс и соответствующие отношения в качестве вкладов в EditorialObject, например, hasContributor Creator (подкласс Contact), который, в свою очередь, будет уточняться в hasRole Role.
hasContact	Для определения контактов (лиц), связанных с Organisation или другим Contact (лицом).
И т.д.	С Agent могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
И т.д.	С Agent могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.4.2 Contact (Контакт)

Определение:

“Contact” – физическое лицо. Организация имеет пункты “Contact”. “Contact” может иметь связанные контакты.

Примеры:

Примеры контактов – сотрудники, ассистенты и т.д.

Пример контакта – пресс-атташе актера.

Отношения класса	
hasContact	Contact, т.е. лицо, может иметь один или более связанных контактов (например, ассистент или родственник).
И т.д.	С Contact могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
ContactID	Идентификатор, присвоенный Contact.
ContactName	Имя, присвоенное Contact.
StageName	Имя, присвоенное Contact в контексте определенной деятельности, например, имя персонажа или актера в художественном фильме.
И т.д.	С Contact могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

**3.2.4.3 Organisation (Организация)**Определение:

“Organisation” – любая компания, участвующая в реализации MediaResource или EditorialObject.

Примеры:

EBU – пример Organisation.

Отношения класса	
hasContact	Organisation может иметь один или более связанных Contact (например, пресс-контакт).
И т.д.	С Organisation могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
OrganisationID	Идентификатор, присвоенный Organisation.
OrganisationName	Имя, присвоенное Organisation.
DepartmentName	Название определенного департамента внутри Organisation.
И т.д.	С Organisation могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

**3.2.4.4 Event (Событие)**Определение:

Любое «событие», связанное с MediaResource или EditorialObject.

Примеры:

Event – например, историческое или спортивное событие.

Отношения класса	
И т.д.	С Event могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
EventID	Идентификатор, присвоенный Event.
EventType	Тип Event, например, название спортивного события или период времени.
EventName	Имя, присвоенное Event.
И т.д.	С Event могут быть связаны другие свойства класса. См. EBU Tech 3293, EBUCore.

**3.2.4.5 Location (Место)**Определение:

Любое «место», связанное с контентом.

Примеры:

Вымышленное место, изображенное в MediaResource или EditorialObject.

Отношения класса	
И т.д.	С Location могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.



Свойства класса	
LocationID	Идентификатор, присвоенный Location.
LocationType	Тип Location, например, название места (например, Женева) или другая информация о месте, например, «место съемки» и т.д.
LocationName	Имя, присвоенное Location.
И т.д.	С Location могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.4.6 Role (Роль)

#### Определение:

“Role”, исполняемая Agent. “Role” будет идентифицироваться, например, понятием из схемы классификации SKOS. Поэтому “Role” следует считать классом, т.е. подклассом понятия SKOS.

#### Примеры:

Contact может быть актер.

Отношения класса	
И т.д.	С Role могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.
Свойства класса	
И т.д.	С Role могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.

### 3.2.5 Область распространения

Этот раздел вводит классы, связанные с распространением контента.

Центральный класс области распространения – PublicationEvent. Чтобы модель была простой и открытой, было решено сфокусироваться на описании PublicationEvents.

PublicationEvent может быть, например:

- Событием вещания, т.е. изолированным событием, таким как выпуск последний известий и т.п. Этот контент может передаваться через эфир или потоком.
- Запланированное событие, т.е. каждое событие, идентифицированное в определенный интервал времени. Этот контент может передаваться через эфир или потоком.
- Событие по заказу, т.е. контент, доступный для немедленного просмотра или загрузки. Доступ обычно имеет определенный интервал времени. Catch-up TV считается событием по заказу.
- Онлайн событие, т.е. контент, доступный для загрузки / пользования в веб-хранилище (например, на веб-сайте)

Каждое событие из любой вышеуказанной категории предлагается через услуги, управляемые организациями (поставщиками услуг). PublicationChannel – общий термин, включающий все типы средств распространения, включая вещание, онлайн потоки, по заказу и т.д.

В соответствии с типом PublicationEvent MediaResource имеется в разных форматах, конкретизированных в файлах или пакетах сущности.

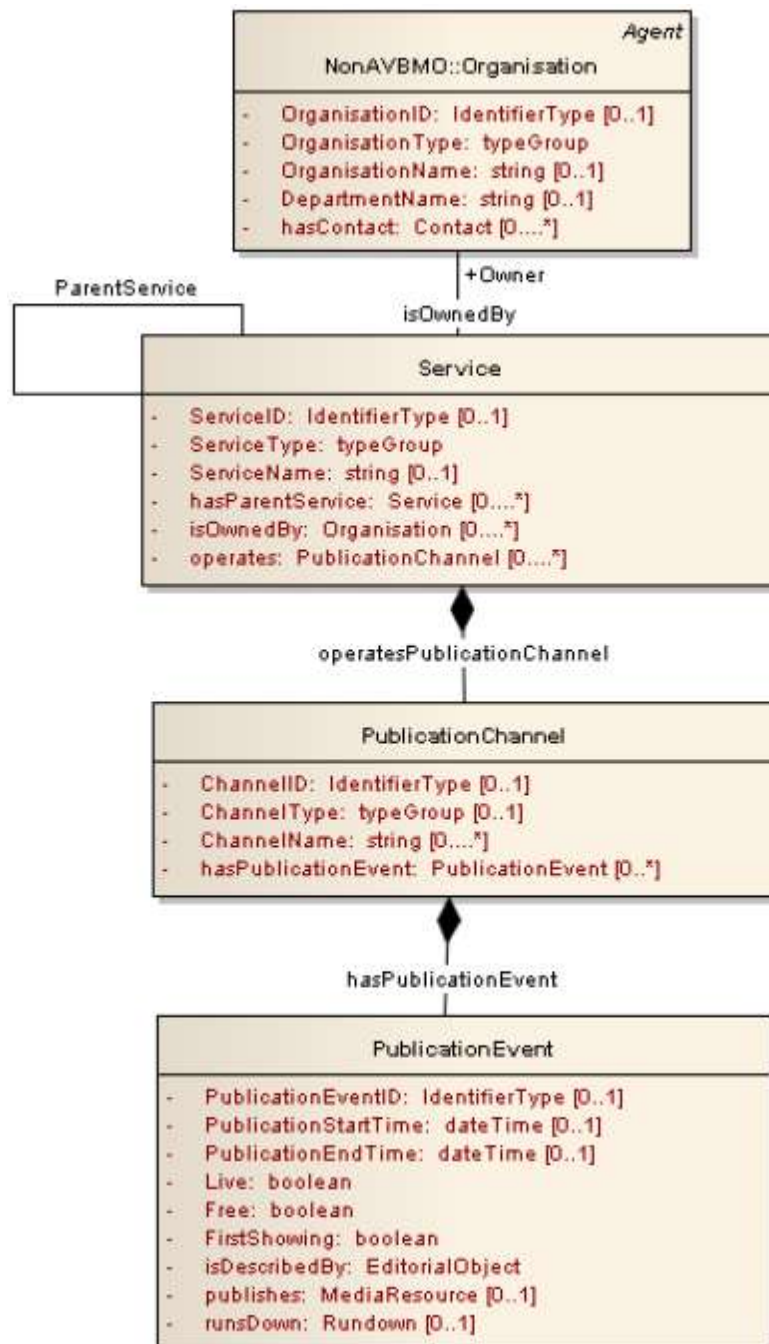


Рис. 11: PublicationEvent

### 3.2.4.1 Service (Услуга)

Определение:

“Service” – коллекция выходов, содержащих общий материал для различных PublicationChannels.

“Service” может быть связана с Brand.

Примеры:

Связанные выходы услуг – телепередача, интернет-потoki или Internet VoD / catch-up TV и т.д.

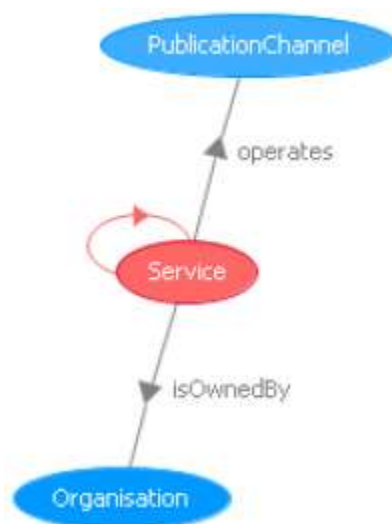


Рис. 12: График отношений Service

Отношения класса	
isOwnedBy	Организация, владеющая / эксплуатирующая Service.
hasParentService	Головная услуга, к которой принадлежит Service.
operates	PublicationChannel, который использует эта услуга.
И т.д.	С Service могут быть связаны другие отношения класса. См., например, ETSI TS 102 822 (TV-Anytime) или BBC Programme Ontology.
Свойства класса	
ServiceId	Идентификатор, связанный с Service.
ServiceType	Тип, связанный с Service, например, вещание, онлайн поток, по заказу.
ServiceName	Имя, связанное с Service.
И т.д.	Для определения Service можно использовать много других свойств. См., например, ETSI TS 102 822 (TV-Anytime) или BBC Programme Ontology.

### 3.2.4.2 PublicationChannel (Канал публикации)

#### Определение:

“PublicationChannel” – физический канал, в котором публикуется сущность, связанная с MediaResource. Один выход или услуга могут быть связаны со множеством PublicationChannels.

#### Примеры:

Канал вещания, сервер интернет-поток и т.д.

Отношения класса	
hasPublicationEvent	PublicationEvents, опубликованные через PublicationChannel.
И т.д.	С Location могут быть связаны другие отношения класса. См., например, ETSI TS 102 822 TV-Anytime) или BBC Programme Ontology.
Свойства класса	
ChannelId	Идентификатор, связанный с PublicationChannel.
ChannelType	Тип, связанный с PublicationChannel (например, вещание или онлайн поток), также называется выходом публикации.
ChannelName	Имя, связанное с PublicationChannel.
И т.д.	Для определения Service можно использовать много других свойств. См., например, ETSI TS 102 822 (TV-Anytime) или BBC Programme Ontology.

### 3.2.4.3 PublicationEvent (Событие публикации)

#### Определение:

Обеспечение сущности в виде представления MediaResource для потребления пользователями называется “PublicationEvent”.

#### Примеры:

Event может быть запланированным событием, т.е. временным окном в расписании, связанным с PublicationChannel. Event также может быть незапланированным событием вещания, например, специальным репортажем. Event также может быть потоковым событием или событием публикации VoD.



Рис. 13: График отношений PublicationEvent

Отношения класса	
publishes	PublicationEvent – публикация MediaResource в формате, соответствующем PublicationChannel, конкретизированному в файле или пакете сущности.
runsDown	Для упорядоченной публикации привязанной ко времени серии медиа ресурсов / сущности как PublicationEvent.
И т.д.	С PublicationEvent могут быть связаны другие отношения класса. См., например, ETSI TS 102 822 (TV-Anytime) или BBC Programme Ontology.
Свойства класса	
PublicationEventid	Идентификатор, связанный с PublicationEvent.
PublicationStartTime	Время, когда по расписанию должна начинаться программа или когда контент делается доступным или может потребляться.
PublicationEndTime	Время, когда программа по расписанию должна заканчиваться или после которого контент недоступен или не предназначен для потребления.
Live	При установке флажок указывает, что контент публикуется в прямом эфире.
Free	При установке флажок указывает, что контент доступен или может потребляться без подписки.
FirstShowing	При установке флажок указывает, что контент доступен в данном PublicationChannel в первый раз.
И т.д.	Для определения PublicationEvent можно использовать много других свойств. См., например, ETSI TS 102 822 (TV-Anytime) или BBC Programme Ontology.

## 4. Принципы реализации / Вопросы и ответы

### 4.1 Общие замечания

Этот раздел содержит советы и пояснения для пользователей в реализации EBU CCDM в различных контекстах, например, в производстве, включая архивы, или для приложений семантической паутины или связанных открытых данных.

### 4.2 Как приспособить мою модель?

Приложение А.1 содержит ссылку на шаблон, помогающий пользователям разработать собственную совместимую с CCDM модель, использующую CCDM.

### 4.3 Еще вопросы?

Если у Вас возникнут вопросы по использованию или реализации EBU CCDM, передайте их на [metadata@ebu.ch](mailto:metadata@ebu.ch). Вы получите персональную консультацию, и по Вашему согласию ответы будут использованы для обогащения данного раздела в будущей версии спецификации.

## 5. Соответствие CCDM

CCDM – открытая структура, позволяющая каждому пользователю адаптировать ее под свои нужды. EBU CCDM имеет гибкий и приспособляемый характер.

Онтология CCDM представлена как опорная программная реализация в RDF/OWL. Она доступна в «Зоне скачивания». Этот файл содержит минимальный набор классов, иерархий классов, objectProperties и dataProperties, которые реализации должны содержать, расширять, но не заменять. Дополнительная информация об онтологии CCDM содержится в **Приложении А**.




## 6. Зона скачивания

Имя файла и место	Описание	Содержание
	Онтология CCDM, выраженная в RDF/OWL	CCDM_Core.owl
	Справка CCDM, пользовательский шаблон и образец рабочей книги Excel (.xlsx)	CCDM_templates.xlsx

## 7. Режим лицензирования

EBU CCDM управляется Creative Commons' Attribution-NonCommercial-ShareAlike3.0 Unported (CC BY-NC-SA 3.0)

Вы можете: *делиться* [копировать, распространять и передавать объект], *ремикшировать* [адаптировать объект, включая собственное пространство имен] при следующих условиях:

	<b>Атрибуция</b> – Вы должны производить атрибуцию в манере, определенной автором или лицензиаром (но не в манере, предполагающей, что они рекомендуют Вас или Ваше использование данного объекта).
	<b>Некоммерчески</b> – Вы не вправе использовать данный объект в коммерческих целях. (Примечание: он может использоваться в коммерческих продуктах, но не продаваться как особая характеристика.)
	<b>В равных долях</b> – Если Вы изменяете, преобразуете или строите на базе данного объекта, Вы можете распространять результирующий объект только по той же или аналогичной лицензии.

## 8. Обслуживание

Спецификация EBU CCDM поддерживается EBU, и предложения по корректировке или дополнениям можно направлять по почте (metadata@ebu.ch).

## 9. Полезные ссылки

EBU Metadata (<http://tech.ebu.ch/metadata/>)

EBUCore ([http://tech.ebu.ch/docs/tech/tech3293v1\\_3.pdf](http://tech.ebu.ch/docs/tech/tech3293v1_3.pdf))

BBC Programmes Ontology (<http://www.bbc.co.uk/ontologies/programmes/2009-09-07.shtml>)

TV-Anytime (<http://www.etsi.org>, Standard download in the TS 102 822 series)

EBU-AWMA FIMS (<http://wiki.amwa.tv/ebu>)

W3C – SKOS (<http://www.w3.org/2004/02/skos/>)

## Приложение А: Шаблон преобразования EBU CCDM и пример

### А.1 Введение

Ожидается, что пользователи CCDM могут использовать собственные дополнительные классы и свойства.

К спецификации прилагается рабочая книга Excel. Она содержит три таблицы Excel:

- Первая таблица представляет структуру модели CCDM с классами, ObjectProperties и DataProperties. Она должна использоваться как опора и не модифицироваться.
- Вторая таблица – шаблон для добавления классов и свойств. Инструкции даны в следующем разделе данного Приложения.
- Третья таблица содержит пример, иллюстрирующий использование механизма расширения.

Пример показывает, как преобразовать собственную «модель» в EBU CCDM и что можно сделать, чтобы эта модель расширяла общую структуру CCDM в соответствующей манере.

### А.2 Инструкции по использованию шаблона

Шаблон состоит из 16 столбцов согласно следующей таблице.

Название столбца	Определение
CCDM Class	Опорные классы CCDM
CCDM SubClass	Опорные подклассы CCDM
User Custom Class	Место для пользовательских классов, определяемых как классы CCDM или новые собственные корневые классы
Upper Class	Пользователи должны указать, каким классом порожден их собственный класс.
Class Instance	Место для экземпляров CCDM и пользовательских классов.
CCDM Object Property	Опорные свойства объектов CCDM
User Custom Object Property	Место для пользовательских свойств объектов
Upper Object Property	Пользователи должны указать, каким свойством объекта порождено их собственное свойство объекта, если используются суб-свойства.
Object Property Range	Место для определения класса, к которому применяется свойство.
Object Property Range Instance	Место для экземпляров CCDM и пользовательских Object Properties Range Classes
User Custom Data Property	Место для пользовательских свойств данных.
Upper Data Property	Пользователи должны указать, каким свойством объекта порождено их собственное свойство объекта, если используются суб-свойства.
Data Property Range / Datatype	Место для определения типов данных свойства данных
Data Property Instance	Экземпляр свойства данных
Note	Место для пользовательских примечаний.

#### Важные примечания:

- Подклассы наследуют свойства из верхних классов (например, классов CCDM);
- Можно добавлять собственные классы;
- Также можно добавлять собственные свойства объектов и данных, но рекомендуется добавление классов и использование предопределенных свойств CCDM;
- ID – неотъемлемые свойства классов (определяемые их URI);
- Типы (например, EditorialObject, Organisation, Part или типы MediaResource) должны быть реализованы через определение подклассов (например, Manufacturer определяется как подкласс Organisation);
- Термины, которые можно взять из схем классификации, например, жанр или роль, следует считать классами. Если такие классы выражены в SKOS, они являются подклассами SKOS Concept Class. Если идентификатор термина (например, понятие SKOS) отсутствует, можно использовать дополнительные свойства, связанные с классом, через a blank node.

## Приложение В: Онтология EBU CCDM

Опорная программная реализация CCDM дана в RDF/OWL.

Ссылка для скачивания дана в §6 («Зона скачивания») данной спецификации.

Имеется ряд вариантов для синтаксического анализа и редактирования документов и онтологий RDF/OWL:

- Файлы с использованием расширения 'owl' можно открыть текстовыми редакторами типа Wordpad;
- Можно также использовать Notepad;
- Можно использовать специальные программы:
  - Protégé (<http://protege.stanford.edu/download/download.html>) (рекомендуется для новичков)
  - NeonToolkit ([http://neon-toolkit.org/wiki/Main\\_Page](http://neon-toolkit.org/wiki/Main_Page))
  - TopBraid Composer, free edition ([http://www.topquadrant.com/products/TB\\_Composer.html](http://www.topquadrant.com/products/TB_Composer.html))