

# EBU

OPERATING EUROVISION

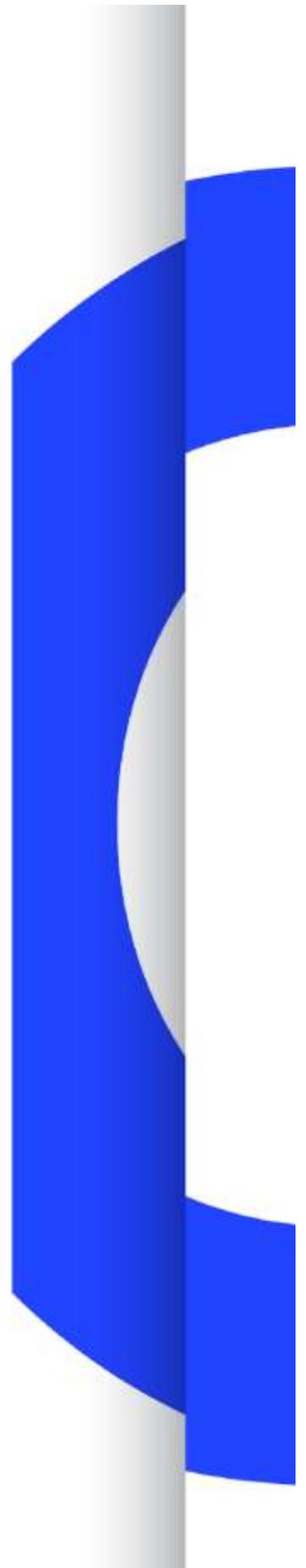
## TECH 3351

### EBU CLASS CONCEPTUAL DATA MODEL (CCDM)

Source: MIM

Version 1.1

Geneva  
January 2016



# EBU

OPERATING EUROVISION AND EURORADIO

TECH 3351

## EBU CLASS CONCEPTUAL DATA MODEL (CCDM)

**Внимание!**

Данный перевод **НЕ** претендует на аутентичность  
и может содержать отдельные неточности.

Оригинал документа на сайте <https://tech.ebu.ch>

## КОНЦЕПТУАЛЬНАЯ МОДЕЛЬ КЛАССОВ ДААННЫХ (CCDM) EBU

Источник: MIM

Версия: 1.1

Женева  
Январь 2016

## Введение

Концептуальная модель данных классов (CCDM) EBU – это онтология, определяющая базовый набор классов и свойств как общий словарь для бизнес-объектов, например, программ, статей и других типов контента, и их отношения в бизнес-процессах медиа предприятий. Примеры – программы на различных этапах создания от заказа до выдачи, связанные с ними права или события публикации и т.д.

CCDM – общая структура, и мы приглашаем пользователей к дальнейшему обогащению модели классами и свойствами, которые наиболее соответствуют их нуждам. Свойства для описания каждого объекта можно найти в EBUCore или определить свои собственные.

Это версия 1.1 “CCDM”.

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.

Спецификация CCDM сочетает некоторые аспекты из существующих моделей и спецификаций в общей структуре. Она построена на нескольких попытках EBU представить вещание как простую логическую модель. В ней использована работа EBU по моделированию метаданных (IP-META и EBUCore) и разработки семантической паутины. Область распространения разработана с учетом максимального преобразования в TV-Anytime и “BBC Programmes Ontology”.

Онтология CCDM представлена в RDF/OWL и диаграммах классов.

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

## Сроки и условия использования

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

© EBU 2016.

**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

|  |           |
|--|-----------|
| Введение .....   | 4         |
| <b>1. Область действия .....</b>                                   | <b>4</b>  |
| 1.1 Обоснование .....  | 4         |
| <b>2. Концептуальная модель данных классов .....</b>               | <b>5</b>  |
| 2.1 Основные принципы .....  | 5         |
| 2.2 Классы, отношения и свойства .....                             | 6         |
| 2.2.1 Юридическая область .....                                    | 6         |
| 2.2.1.1 Asset .....  | 7         |
| 2.2.1.2 Rights .....   | 8         |
| 2.2.1.3 Contract .....   | 8         |
| 2.2.2 Область редактирования .....                                 | 8         |
| 2.2.2.1 EditorialObject .....                                      | 9         |
| 2.2.2.2 TimelineTrack .....  | 11        |
| 2.2.2.3 Location .....   | 11        |
| 2.2.2.4 Event .....  | 11        |
| 2.2.2.5 Agent.....   | 12        |
| 2.2.2.6 Role .....   | 12        |
| 2.2.3 Область производства .....                                   | 12        |
| 2.2.3.1 Resource .....   | 13        |
| 2.2.3.2 MediaResource .....  | 14        |
| 2.2.3.3 Track .....  | 14        |
| 2.2.3.4 Format .....   | 14        |
| 2.2.3.5 Essence .....  | 15        |
| 2.2.3.6 ProductionJob .....  | 16        |
| 2.2.3.7 ProductionDevice .....                                     | 16        |
| 2.2.4 Область распространения .....                                | 16        |
| 2.2.5.1 PublicationEvent .....                                     | 17        |
| 2.2.5.2 Service .....  | 18        |
| <b>3. Принципы реализации / Вопросы и ответы .....</b>             | <b>18</b> |
| 3.1 Общие замечания .....  | 18        |
| 3.2 Примеры, предоставленные SRG SSR, Швейцарская Конфедерация ... | 18        |
| 3.3 Пример, предоставленный TV2, Норвегия .....                    | 24        |
| 3.4 Общая диаграмма классов .....                                  | 25        |
| 3.5 Еще вопросы?.....  | 25        |
| <b>4. Соответствие CCDM .....</b>                                  | <b>26</b> |
| <b>5. Зона скачивания .....</b>                                    | <b>26</b> |
| <b>6. Режим лицензирования .....</b>                               | <b>26</b> |
| <b>7. Обслуживание .....</b>                                       | <b>26</b> |
| <b>8. Полезные ссылки .....</b>                                    | <b>26</b> |
| <b>Приложение А: Онтология EBU CCDM .....</b>                      | <b>27</b> |

Примечание переводчика. В оригинале имеются некоторые неточности:

Нумерация: после п. 2.2.4 идут пп. 2.2.5.1 и 2.2.5.2.

Стр. 8, 9, 11, 15, 17 – в таблицах скопирован текст из других таблиц без замены соответствующих слов на нужные.

## Концептуальная модель данных классов (CCDM) EBU

| Комитет EBU | Первый выпуск | Переработка | Переиздание |
|-------------|---------------|-------------|-------------|
| MIM         | Октябрь 2012  | Январь 2016 |             |

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

### 1. Область действия

Концептуальная модель данных классов (CCDM) EBU – это онтология, определяющая базовый набор классов и свойств как общий словарь для описания программ на различных этапах создания от заказа до выдачи. CCDM – общая структура, и мы приглашаем пользователей к дальнейшему обогащению модели классами и свойствами, которые более соответствуют их нуждам.

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 Интернету, моделирование семантической паутины и связанные открытые данные (LOD).

Эта спецификация – модель классов, онтология, а не спецификация метаданных. Свойства метаданных и типы данных (кроме отношений между классами) являются **показательными**. Пользователи, желающие адаптировать модель CCDM к своим нуждам, приглашаются к описанию классов и собственных расширений CCDM, используя либо свойства из EBU Tech 3293 (набор метаданных EBUCore), либо другие спецификации метаданных (например, TVAnytime или собственные схемы метаданных).

Спецификация CCDM сочетает некоторые аспекты из существующих моделей и спецификаций в общей структуре. Она построена на нескольких попытках EBU представить вещание как простую логическую модель. В ней использована работа EBU по моделированию метаданных (IP-META и EBUCore) и разработки семантической паутины. Область распространения разработана с учетом максимального преобразования в TV-Anytime и “BBC Programmes Ontology”.

Онтология CCDM представлена в RDF/OWL.

#### 1.1 Обоснование

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

- Большого понимания бизнес-моделей и рабочих процессов;
- Оптимизации процессов с более легким и надежным обменом данными;
- Более простого и рационализованного описания классов медиа;
- Более простой реализации архитектур производства, ориентированного на медиа услуги;
- Принятия новых моделей управления информацией, таких как семантическая паутина и связанные данные (обогащение, улучшение поиска и доступности).

CCDM разработана для того, чтобы позволить реализаторам адаптировать имена классов и их отношения в соответствии со своими нуждами. Призываем каждую организацию провести анализ и создать собственную модель, начиная со структуры CCDM как общей базы для сравнения с моделями других пользователей CCDM.

## 2. Концептуальная модель данных классов

### 2.1 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 (см. учебники по RDF и OWL), в определенных проектах также называется «бизнес-объекты» или «понятия», см. также [http://protege.stanford.edu/publications/ontology\\_development/ontology101.pdf](http://protege.stanford.edu/publications/ontology_development/ontology101.pdf). W3C's Media-Ontology (MA-ONT) основана на модели классов CCDM (<http://www.w3.org/ns/ma-ont.rdf>).

- Отношения: связывают классы (например, 'Programme hasContributor Person').
  - Примечание: эквивалент понятия *objectProperties*, используемого в моделировании семантической паутины (см. учебники по RDF и OWL).
- Свойства: определяют внутренние характеристики классов (например, 'bitrate', выраженной как целое число, или 'name', выраженной в виде строки).
  - Примечание: эквивалент понятия *dataProperties*, используемого в моделировании семантической паутины (см. учебники по RDF и OWL).

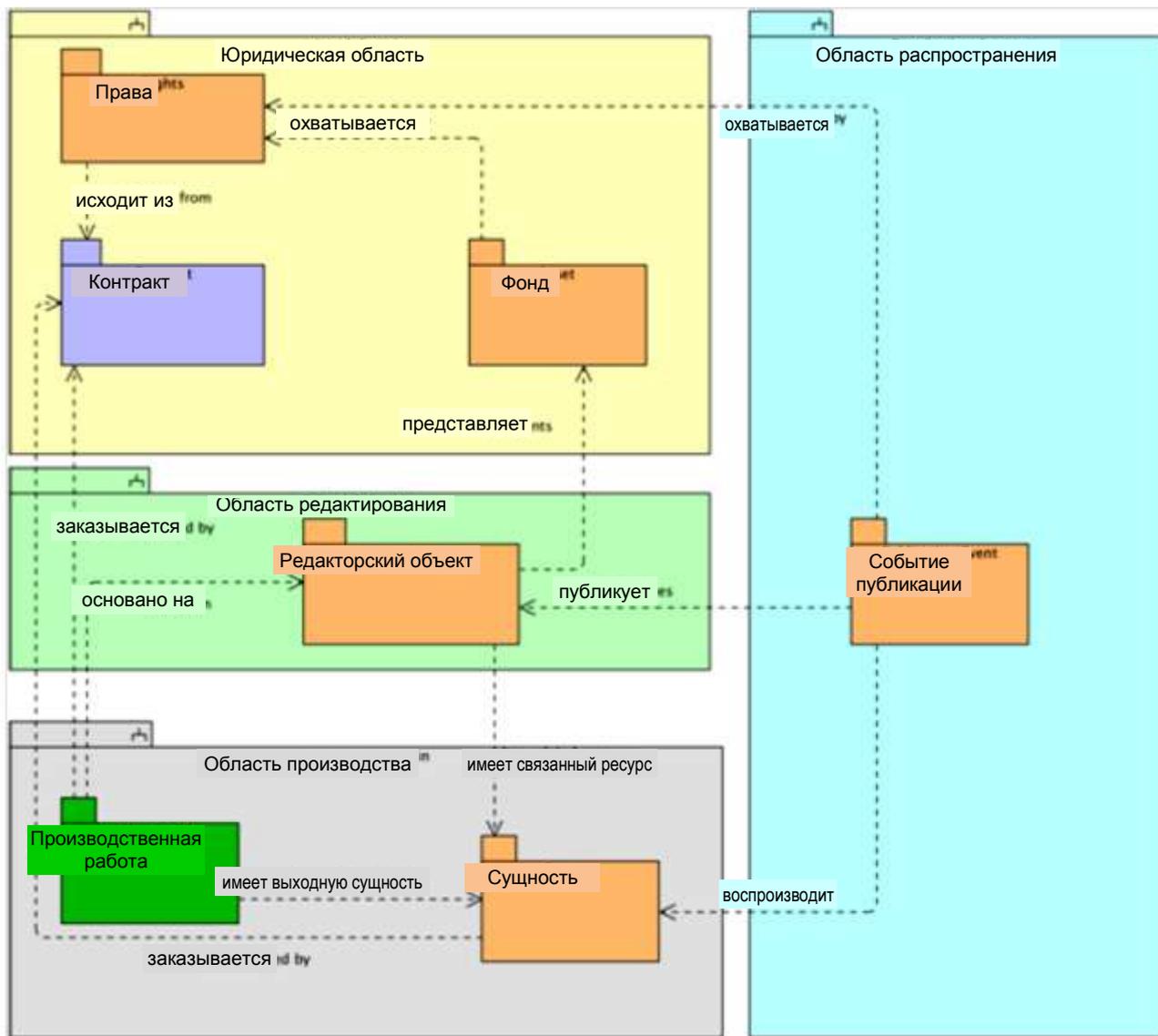


Рис. 1: Области CSDM

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

- **Юридическая область** – где происходит управление заказами, производственными заказами, интеллектуальной собственностью и другими правами, связанными с контентом и его публикациями. После того как идея программы разработана и согласована, заказывается производство, а ресурсы идентифицируются и резервируются через контракт на заказ производственной работы. Центральным классом юридической области – это фонд, который устанавливает отношение с *EditorialObject*, с информацией об интеллектуальной собственности и правах.
- **Область распространения** – куда входит любая форма публикации, воспроизведения или распространения. Центральным классом – *PublicationEvent*, который играет роль сущности, т.е. медиа объект, являющийся результатом производственной работы.
- **Область редактирования** – где происходит управление информацией, связанной с концепцией и контентом. Кроме того, здесь представлены все монтажные решения. *Editorial Object* – центральный класс этой области. Он может группироваться и упорядочиваться на тайм-линии.
- **Область производства** – где производственные заказы реализуются путем приобретения *MediaResource* (например, изготовления объекта путем производственной работы, закупки или поиска материала) согласно производственному плану. *MediaResource*, готовые к публикации, используют класс *Essence* для связи контента с определенной публикацией.

EBU CSDM разработан для того, чтобы позволить пользователям адаптировать имена классов и отношений в соответствии со своими нуждами. Например, класс '*EditorialObject*' может быть типа «программа», «элемент» или «план», но также представлять группу «цикл», «сериал» или «сезон». Определение соответствующих свойств остается за пользователем. Базовый набор свойств предлагается в EBU Tech 3293, EBUCore или в других спецификациях метаданных (например, TV-Anytime или собственных схемах метаданных).

## 2.2 Классы, отношения и свойства

См. Рис. 2, где проиллюстрированы отношения между областями и объектами.

### 2.2.1 Юридическая область

Это область, в которой интеллектуальная собственность, права и контракты управляются и связываются с *MediaResource* и/или *EditorialObject* и косвенно с *PublicationEvent* (включая условия эксплуатации и распространения) для определения *Asset*. Эта область также включает заказ продукции и материала.

Центральный класс юридической области – фонд, который служит соединением между набором прав и редакторским объектом.

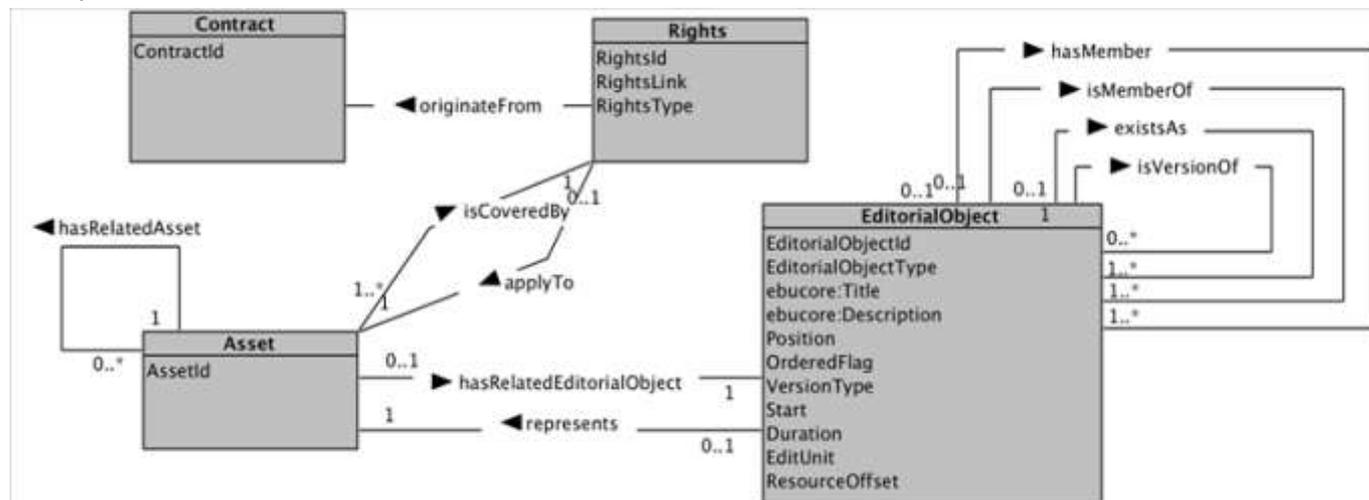


Рис. 3: Фонд

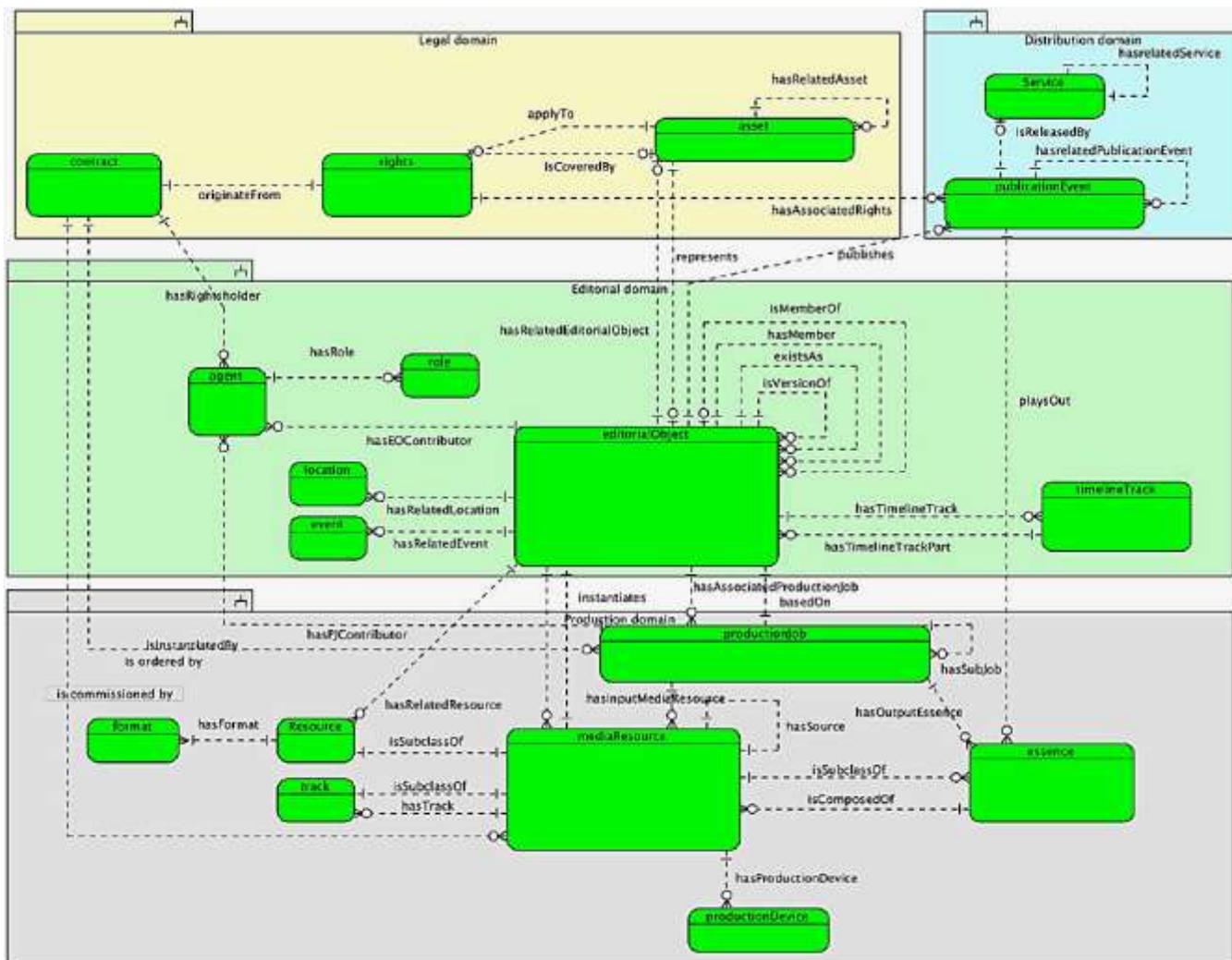


Рис. 2: Области, объекты и отношения

### 2.2.1.1 Asset (Фонд)

#### Определение:

Класс "Asset" – объект, с которым будет связан идентификатор при заказе. Он будет служить центральной эталонной точкой для управления правами, связанного с *EditorialObjects*, *MediaResources* или *Essences* и – косвенно – с *PublicationEvents* (условия распространения и эксплуатации).

Помните, что *MediaResources* или сущности в этой модели всегда будут представлены *EditorialObject*.

#### Пример:

Модель CCDM позволяет связывать Rights с соответствующим *EditorialObject*, представляющим Essence.

| Отношения класса                 |  |
|----------------------------------|--|
| <i>hasRelatedEditorialObject</i> | Указатель на <i>EditorialObject</i> , который фонд связывает со своими правами.  |
| <i>hasRelatedAsset</i>           | Указатель на другой фонд (например, телесериал), с которым связан этот фонд      |
| <i>isCoveredBy</i>               | Указатель на права, связанные с <i>EditorialObject</i>                           |
| <i>И т.д.</i>                    | С фондом могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore. |
| Свойства класса                  |  |
| <i>AssetID</i>                   | Идентификатор, связанный с фондом  |
| <i>И т.д.</i>                    | С фондом могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.         |

### 2.2.1.2 Rights (Права)

#### Определение:

Класс "Right" определяет права, исходящие из контракта. Right связан с *MediaResource* через определение Asset. *RightsHolder* соединяется через Contract.

| Отношения класса     |   |
|----------------------|---|
| <i>applyTo</i>       | Указатель на фонд, т.е. <i>EditorialObject</i> , к которому применяются права.  |
| <i>originateFrom</i> | Указатель на контракт, предоставляющий права.                                   |
| <i>И т.д.</i>        | С Rights могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore |
| Свойства класса      |   |
| <i>RightsID</i>      | Идентификатор, связанный с Rights.  |
| <i>RightsType</i>    | Тип, связанный с Rights, например, лицензионные условия.                        |
| <i>RightsLink</i>    | Ссылка, например, на веб-ресурс, где можно найти правовые условия.              |
| <i>И т.д.</i>        | С Rights могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.        |

### 2.2.1.3 Contract (Контракт)

#### Определение:

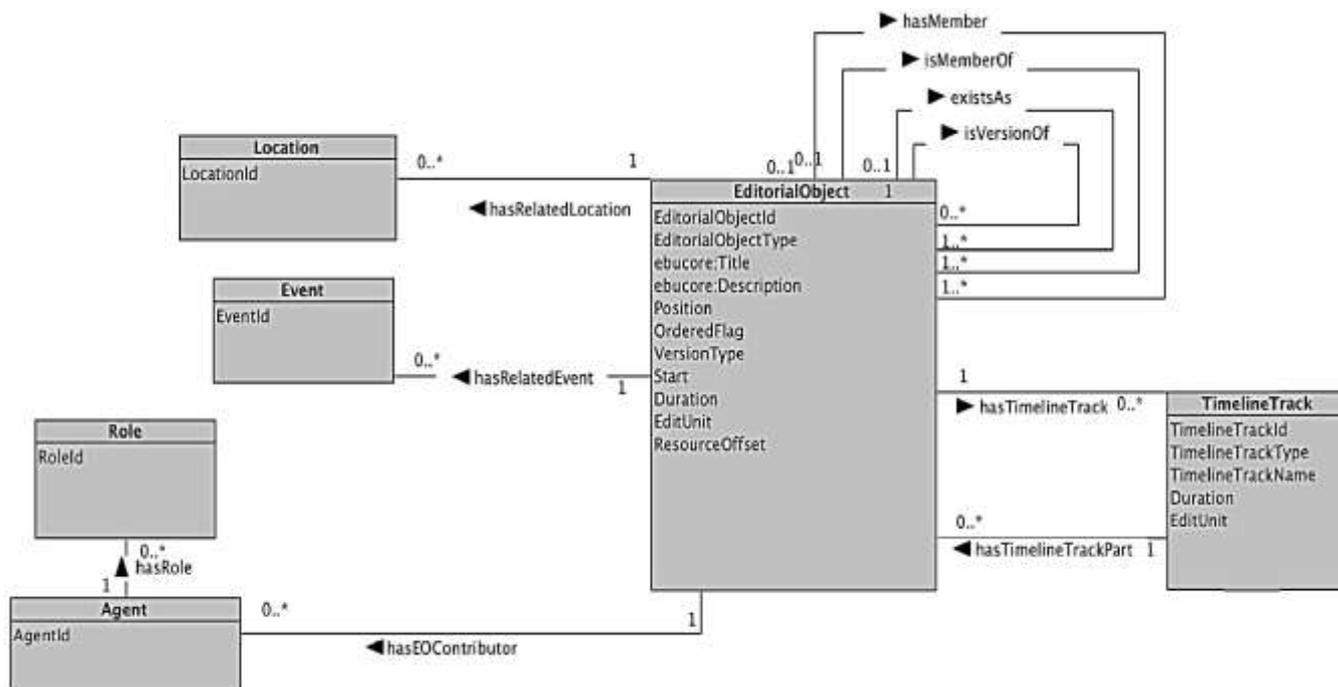
Класс "Contract" представляет любой юридический документ, охватывающий вопросы прав или заказов. Этот объект / класс включает комбинацию производственных заказов и сбыта. Контракт соединяет права с любыми *RightsHolders*. Контракт определяет один или более наборов прав.

| Отношения класса       |   |
|------------------------|---|
| <i>hasRightsHolder</i> | Указатель на Agent, владеющий правами.  |
| <i>Etc.</i>            | С Rights могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore |
| Свойства класса        |   |
| <i>ContractID</i>      | Идентификатор, связанный с Contract   |
| <i>Etc.</i>            | С Rights могут быть связаны другие свойства. See EBU Tech 3293, EBUCore.        |

### 2.2.2 Область редактирования

Область редактирования – это область, в которой определяется заказанная концепция перед изготовлением и распространением. Все метаданные, связанные с идеей программы (например, контент, формат, цель, аудитория, интервал в расписании), связанные с контентом программы (например, титры, темы, вкладки, места, события) и все монтажные решения представлены в соответствующих классах.

Центральный класс области редактирования – *EditorialObject*.

Рис. 4: Классы вокруг *EditorialObject*

### 2.2.2.1 EditorialObject (Редакторский объект)

#### Определение:

Класс “*EditorialObject*” описывает любую идею, любой сюжет и будет использоваться для преобразования концепции в редакторское определение *MediaResource* перед изготовлением (в области производства) и распространением (в области распространения). “*EditorialObject*” – набор описательных метаданных, суммирующий, например, монтажные решения.

“*EditorialObject*” может быть группой.

“*EditorialObject*” также может быть частью другого “*EditorialObject*”, который определяется своим начальным временем и хронометражем.

*EditorialObjects* могут быть упорядочены либо как группы, либо как элементы на тайм-линии.

#### Примеры:

Программа, элемент, план, часть, глава, сегмент и где используются групповые свойства: цикл, сериал, компиляция, коллекция, группа элементов, блок элементов.

Упрощенный вариант применения:

Новостная телепередача состоит из двух новостных элементов. Один элемент содержит последние 10 секунд 1-минутного интервью, взятого из другого источника (т.е. 50" > 60"). Это можно смоделировать следующим образом:

- *NewsBroadcast* связан с *MediaResource* посредством свойства создания экземпляров
- *NewsItems* связан с *NewsBroadcast* посредством *TimelineTrack*.
- *InterviewPart* связан с *NewsItem* с помощью свойства *hasMember*. *Start* и *Duration* – свойства внутри *InterviewPart*, указывающие его присутствие внутри *NewsItem*.
- *InterviewPart* связан со своим источником с помощью свойства *existsAs*.
- *Interview* создает экземпляр *MediaResource*, который в свою очередь доступен из *MediaResource* в *NewsBroadcast* посредством свойства *hasSource*.
- Представление сегментации: *TimelineTracks* предпочтительнее свойств *hasPart*, когда нужен порядок прогона, например, для эфира.

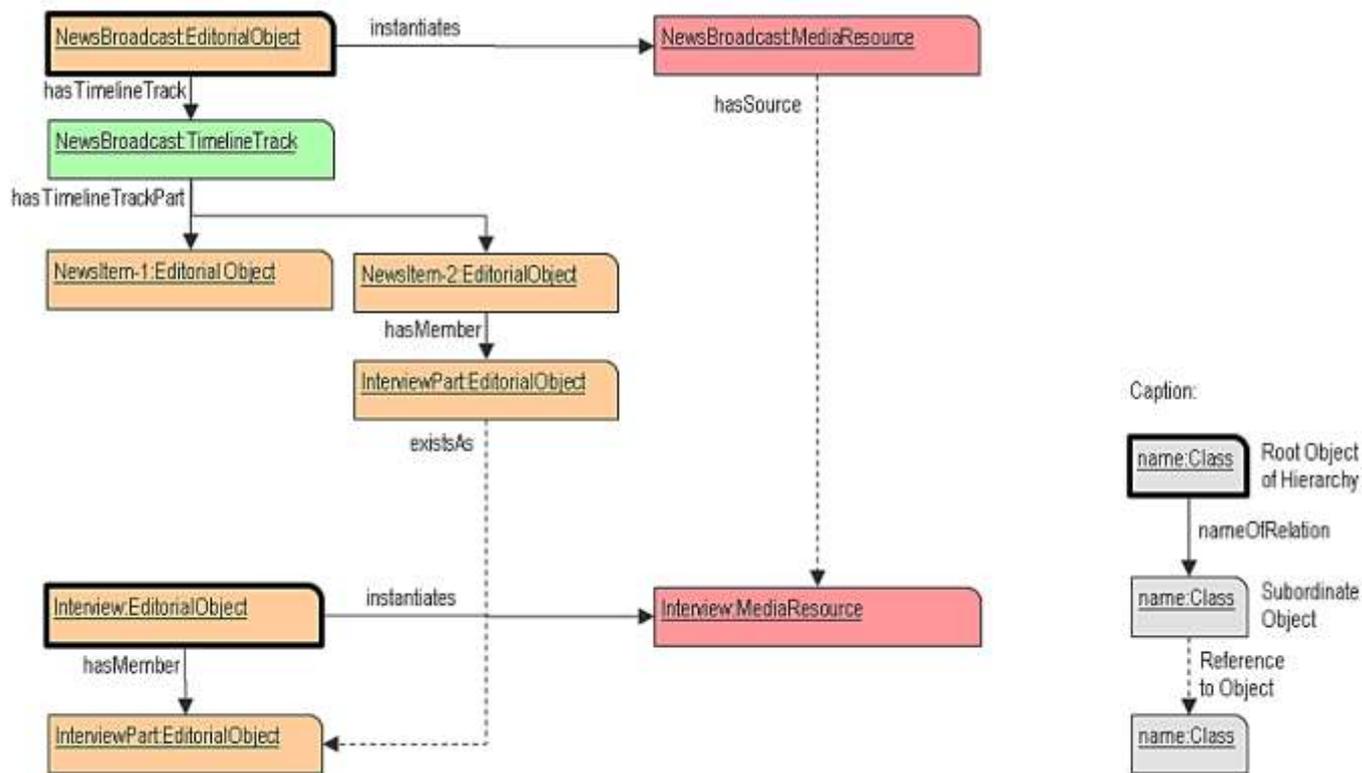


Рис. 5: Иллюстрация варианта применения

| Отношения класса                  |  |
|-----------------------------------|--|
| <i>isMemberOf</i>                 | Список групп, членом которых является <i>EditorialObject</i> .   |
| <i>hasMember</i>                  | Список <i>EditorialObjects</i> , которые содержит <i>EditorialObject</i> и которые не являются частью тайм-линии. Эпизод сериала – пример такого отношения   |
| <i>hasRelatedResource</i>         | Отношение для идентификации ресурса, связанного с <i>EditorialObject</i> ,   |
| <i>isInstantiatedBy</i>           | Отношение для идентификации <i>MediaResource</i> , создающего экземпляр <i>EditorialObject</i>   |
| <i>hasEOContributor</i>           | Агент(ы), внесший вклад в реализацию <i>EditorialObject</i> . Вклад характеризуется по Agent Role. Agent – не-медиа класс, описанный в другом разделе данного документа. Свойство " <i>hasEOContributor</i> " может расширяться под свойствами для более специфических ролей, например, <i>hasEOCreator</i> , <i>hasEODirector</i> . |
| <i>hasRelatedLocation</i>         | Опционально, одно (или более) мест, связанное с <i>EditorialObject</i> , характеризующее типом (например, документальный или художественный).  |
| <i>hasRelatedEvent</i>            | Опционально, одно (или более) событий, связанное с <i>EditorialObject</i> , характеризующее типом (например, спортивное событие / встреча).  |
| <i>represents</i>                 | <i>EditorialObject</i> представляет фонд.  |
| <i>hasAssociatedProductionJob</i> | Производственная работа представляет производственный процесс, в котором <i>EditorialObject</i> реализуется в <i>MediaResource</i> и <i>Essence</i> .  |
| <i>isVersionOf</i>                | Для идентификации <i>EditorialObjects</i> , представляющих альтернативные версии контента.   |
| <i>existsAs</i>                   | Для идентификации <i>EditorialObjects</i> , представляющих альтернативные медиа наборы контента  |
| <i>hasTimelineTrack</i>           | Для связи <i>TimelineTrack</i> , например, <i>RunDown</i> , с <i>EditorialObject</i> , состоящим из других <i>EditorialObjects</i> .   |
| <i>И т.д.</i>                     | С <i>EditorialObject</i> могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.   |
| Свойства класса                   |  |
| <i>EditorialObjectType</i>        | Тип <i>EditorialObject</i> , например, Programme, Item.  |
| <i>EditorialObjectID</i>          | Опционально один (или более) идентификатор, присвоенный <i>EditorialObject</i> .   |
| <i>ebucore:Title</i>              | Основное название, под которым известен <i>EditorialObject</i> . Например, это свойство импортируется из пространства имен EBUCore.  |

|                            |  |
|----------------------------|--|
| <i>ebucore:Description</i> | Опционально одно (или более) описание <i>EditorialObject</i> . Например, это свойство импортируется из пространства имен EBUCore.  |
| <i>Position</i>            | Позиция <i>EditorialObject</i> в <i>EditorialObject</i> типа 'rundown', или в упорядоченной группе   |
| <i>VersionType</i>         | Строка для опциональной идентификации версии <i>EditorialObject</i> , например, удлиненной, сокращенной, с сурдопереводом, со скрытыми субтитрами и т.д.                   |
| <i>Start</i>               | Начальная точка Member, т.е. части, в <i>EditorialObject</i> или в <i>TimelineTrack</i> .  |
| <i>Duration</i>            | Хронометраж Member в <i>EditorialObject</i> или в <i>TimelineTrack</i> .   |
| <i>EditUnit</i>            | Единица, используемая для выражения начала и хронометража.   |
| <i>ResourceOffset</i>      | Начальный сдвиг соответствующего ресурса, если данный ресурс используется не с начала.   |
| <i>OrderedFlag</i>         | Если 'true', то флажок показывает, что члены <i>EditorialObject</i> упорядочены (например, членство обусловлено строгой последовательностью, например, эпизоды в сериале). |
| <i>И т.д.</i>              | С <i>EditorialObject</i> могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.  |

### 2.2.2.2 TimelineTrack (Дорожка тайм-линии)

#### Определение:

“TimelineTrack” используется для определения тайм-линий, т.е. привязанной ко времени последовательности *EditorialObjects* (или части *EditorialObjects*).

| Отношения класса            |   |
|-----------------------------|---|
| <i>HasTimelineTrackPart</i> | Для идентификации частей <i>TimelineTrack</i> , т.е. <i>EditorialObjects</i> с временем начала и хронометражем. |
| <i>И т.д.</i>               |   |
| Свойства класса             |   |
| <i>TimelineTrackID</i>      | Идентификатор, присвоенный <i>TimelineTrack</i> .   |
| <i>TimelineTrackType</i>    | Например, прогон или другие типы, не определенные как подклассы в спецификации                                  |
| <i>Duration</i>             | Хронометраж <i>TimelineTrack</i> в <i>EditorialObject</i> .   |
| <i>EditUnit</i>             | Единица для выражения хронометража.   |
| <i>И т.д.</i>               | С <i>TimelineTrack</i> могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.                   |

### 2.2.2.3 Location (Место)

#### Определение:

“Location” используется для определения мест, например, пространственного охвата сюжета или места записи типа студийного или внестудийного, связанных с *EditorialObjects* (или частью *EditorialObjects*).

| Свойства класса   |  |
|-------------------|--|
| <i>LocationId</i> | Для идентификации места в системе определенных мест  |
| <i>Etc.</i>       | С <i>Location</i> могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore (включая GPS координаты) или <i>GeoNames</i> . |

### 2.2.2.4 Event (Событие)

#### Определение:

“Event” используется для определения события, которое охватывает *EditorialObject*.

#### Примеры:

Олимпийские игры 1994, всеобщие выборы и т.п.

| Свойства класса |   |
|-----------------|---|
| <i>EventId</i>  | Для идентификации события   |
| <i>Etc.</i>     | С <i>EditorialObject</i> могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore. |

### 2.2.2.5 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 |   |
|------------------|---|
| <i>hasRole</i>   | Роль Agent. Роль уточняет " <i>hasContributor</i> ". Альтернативно пользователь может добавить новый класс и соответствующие отношения в качестве вкладов в <i>EditorialObject</i> , например, <i>hasContributorCreator</i> , <i>hasContributorComposer</i> и т.д., который, в свою очередь, будет уточняться в <i>hasRole Role</i> . |
| <i>И т.д.</i>    | С Agent могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.   |
| Свойства класса  |   |
| <i>AgentId</i>   | Идентификатор для Agent   |
| <i>И т.д.</i>    | С Agent могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore.   |

### 2.2.2.6 Role (Роль)

Определение:

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

Пример:

Contact может быть актером.

| Свойства класса |  |
|-----------------|--|
| <i>RoleId</i>   | Идентификатор для роли, предпочтительно из определенного списка ролей  |
| <i>И т.д.</i>   | С Role могут быть связаны другие свойства. См. EBU Tech 3293, EBUCore. |

## 2.2.3 Область производства

Область производства – это область, в которой производственные заказы реализуются путем сбора *MediaResource* (например, изготовления объекта через производственные работы, закупку или поиск материала).

Центральный класс в области производства – *MediaResource* и его подкласс *Essence*.

*MediaResources*, готовые к публикации, используют класс *Essence* для соединения контента с определенной публикацией.

*MediaResource* всегда имеет отношение к *EditorialObject* (области редактирования) описывающему его контент. *Essence* – реализация *MediaResource* в определенном формате, предназначенном для публикации. *Essence* – результат *ProductionJob* и подкласс *MediaResource* и наследует все его свойства, такие как формат, место и *ProductionDevice*.

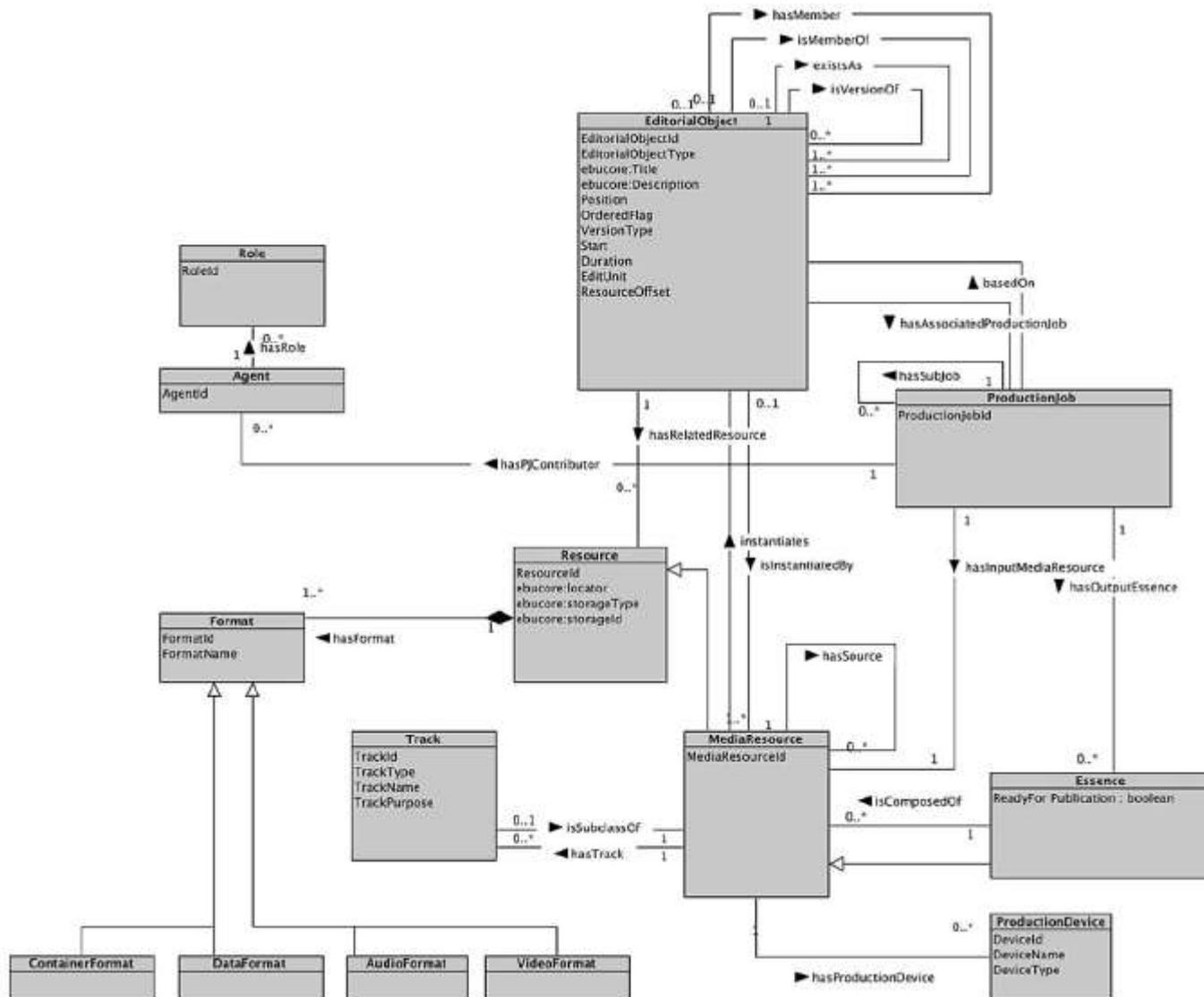


Рис. 6: MediaResource

### 2.2.3.1 Resource (Ресурс)

Определение:

“Resource” – обычный файл, используемый в связи с производством. Он определяется *EditorialObject* (областью редактирования). Он может изготавливаться в одном или более форматах, конкретизируемых в одном или более экземпляров сущности. Имеет индикацию локатора, где его можно найти.

Примеры:

Файл pdf, используемый в поиске, рукопись в хранилище и т.д.

| Отношения класса         |  |
|--------------------------|--|
| <i>hasFormat</i>         | Состав ресурса. Ресурс может существовать в одном или более форматах.  |
| <i>И т.д.</i>            | С <i>MediaResource</i> могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.   |
| Свойства класса          |  |
| <i>Resourceid</i>        | Уникальный идентификатор, например, UUID, UMID, URI и т.д. Может генерироваться или присваиваться в производственном процессе или извлекаться из контента. |
| <i>MediaResourceType</i> | Свободный текст, содержащий дополнительную информацию делового уровня о типе контента или папке.   |
| <i>StorageId</i>         | Идентификатор накопителя   |
| <i>StorageType</i>       | Определение типа / структуры накопителя, в котором хранится Essence.   |
| <i>Locator</i>           | Указывает, где можно найти и получить доступ к определенному <i>Resource</i>   |
| <i>И т.д.</i>            | С <i>MediaResource</i> могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.  |

### 2.2.3.2 MediaResource (Медиа ресурс)

#### Определение:

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

| Отношения класса           |   |
|----------------------------|---|
| <i>hasProductionDevice</i> | Производственное оборудование, ответственное за создание <i>MediaResource</i>   |
| <i>hasSource</i>           | Отношение к <i>MediaResource</i> , служащему источником <i>MediaResource</i> . Например, аналоговая лента, являющаяся источником файла                |
| <i>hasTrack</i>            | Отношение к дорожкам, на которые разделен <i>MediaResource</i> .  |
| <i>instantiates</i>        | Отношение к <i>EditorialObject</i> , который описывает <i>MediaResource</i> .   |
| <i>И т.д.</i>              | С <i>MediaResource</i> могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore.  |
| Свойства класса            |   |
| <i>MediaResourceId</i>     | Уникальный идентификатор, например, UUID, UMID и т.д. Может генерироваться или присваиваться в производственном процессе или извлекаться из контента. |
| <i>И т.д.</i>              | С <i>MediaResource</i> могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.   |

### 2.2.3.3 Track (Дорожка)

#### Определение:

“Track” является частью *MediaResource* и подклассом *MediaResource*. *MediaResource* потенциально состоит из любой комбинации дорожек аудио, видео и данных.

#### Примеры:

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

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

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

| Отношения класса |  |
|------------------|--|
| <i>И т.д.</i>    | С <i>Track</i> могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore. |
| Свойства класса  |  |
| <i>TrackID</i>   | Идентификатор, присвоенный дорожке.  |
| <i>TrackName</i> | Имя, связанное с дорожкой.   |
| <i>И т.д.</i>    | С <i>Track</i> могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.  |

### 2.2.3.4 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 *MediaResource*.

#### Пример:

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

| Часто используемые подклассы |  |
|------------------------------|--|
| <i>subclass</i>              | <i>AudioFormat</i> – подкласс <i>Format</i> , используемый для перечня всех характеристик аудио сигнала. См. например 'audioFormat' в EBU Tech 3293, EBUCore.  |
| <i>subclass</i>              | <i>VideoFormat</i> – подкласс <i>Format</i> , используемый для перечня всех характеристик видео сигнала. См. например 'videoFormat' в EBU Tech 3293, EBUCore.  |
| <i>subclass</i>              | <i>DatFormat</i> – подкласс <i>Format</i> , используемый для перечня всех характеристик сигнала данных.  |
| <i>subclass</i>              | <i>ContainerFormat</i> – подкласс <i>Format</i> , используемый для перечня всех характеристик контейнера. Он обеспечивает информацию о формате контейнера / упаковки в дополнение в информации о кодировании потока в 'channel', (например, mp3, wave, Quicktime, ogg...). См., например 'containerFormat' в EBU Tech 3293, EBUCore. |
| Свойства класса              |  |
| <i>FormatId</i>              | Идентификатор, связанный с форматом.   |
| <i>FormatName</i>            | Имя, связанное с форматом.   |

|        |   |
|--------|---|
| И т.д. | С Format могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore. |
|--------|---|

#### 2.2.3.4.1 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. Этот стандарт определяет модель аудио описаний. |
| Свойства класса  |   |
| И т.д.           | С AudioFormat могут быть связаны другие свойства данных. См. EBU Tech 3293, EBUCore. Этот стандарт определяет модель аудио описаний.  |

#### 2.2.3.4.2 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 данных. См. EBU Tech 3293, EBUCore.  |

#### 2.2.3.4.3 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 данных. См. EBU Tech 3293, EBUCore.  |

#### 2.2.3.4.4 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 данных. См. EBU Tech 3293, EBUCore.  |

#### 2.2.3.5 Essence (Сущность)

##### Определение:

“Essence” – физическое представление *MediaResource* в определенном формате, предназначенном для воспроизведения или публикации. “Essence” является подклассом *MediaResource* и наследует его свойства. “Essence” может быть в форме простого файла или комплексных пакетов (например, передаваемых камерами различных брендов).

##### Примеры:

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

| Отношения класса            |   |
|-----------------------------|---|
| <i>isComposedOf</i>         | Список <i>MediaResources</i> , составляющих Essence.                              |
| И т.д.                      | С Essence могут быть связаны другие отношения класса. См. EBU Tech 3293, EBUCore. |
| Свойства класса             |   |
| <i>ReadyFor Publication</i> | Флажок, который устанавливается, если Essence готова к публикации.                |
| И т.д.                      | С Essence могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.  |

### 2.2.3.6 ProductionJob (Производственная работа)

#### Определение:

“*ProductionJob*” производит *Essence* для публикации, используя *MediaResources* как исходный материал, на основе *EditorialObject*, описывающего процесс в деталях. Заказывается по контракту.

| Отношения класса             |   |
|------------------------------|---|
| <i>basedOn</i>               | Отношение с <i>EditorialObject</i> , который производится данной работой                        |
| <i>hasSubJob</i>             | Отношение с распределением <i>ProductionJob</i> , т.е. отдельная задача в рабочем процессе.     |
| <i>hasInputMediaResource</i> | Список <i>MediaResources</i> , используемых для составления <i>Essence</i> .                    |
| <i>hasOutputEssence</i>      | Отношение с сущностью, являющейся результатом работы.   |
| <i>hasPJContributor</i>      | Информация о бригаде и т.д.   |
| <i>isOrdered</i>             | Отношение с контрактом, заказывающим <i>ProductionJob</i>                                       |
| <i>Etc.</i>                  | С <i>Essence</i> могут быть связаны многие другие отношения класса. См. EBU Tech 3293, EBUCore. |
| Свойства класса              |   |
| <i>ProductionJobId</i>       | Идентификатор для <i>ProductionJob</i>  |
| <i>Etc.</i>                  | С <i>Essence</i> могут быть связаны многие другие свойства. См. EBU Tech 3293, EBUCore.         |

### 2.2.3.7 ProductionDevice (Производственное оборудование)

#### Определение:

“*ProductionDevice*” – устройство, используемое в течение *ProductionJob*.

#### Пример:

Пример *ProductionDevice* – безленточный камкордер.

| Отношения класса  |  |
|-------------------|--|
| <i>И т.д.</i>     | С <i>ProductionDevice</i> могут быть связаны другие отношения класса.  |
| Свойства класса   |  |
| <i>DeviceId</i>   | Идентификатор, связанный с <i>ProductionDevice</i> .   |
| <i>DeviceType</i> | Тип <i>ProductionDevice</i> , например, камкордер.   |
| <i>DeviceName</i> | Имя <i>ProductionDevice</i> .  |
| <i>И т.д.</i>     | С <i>ProductionDevice</i> могут быть связаны многие другие свойства. Примеры дополнительных свойств камкордера можно найти в EBU Tech 3349 (Acquisition Metadata). |

### 2.2.4 Область распространения

Область распространения охватывает все формы публикации, воспроизведения или распространения.

Центральный класс – *PublicationEvent*, который воспроизводит сущность, т.е. медиа объект, являющийся результатом производственного задания.

Другие классы можно добавлять в соответствии с нуждами воспроизведения или распространения.

*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. События по заказу также могут быть связаны с вещательными и запланированными событиями.
- Онлайн событие, т.е. контент, доступный для загрузки / пользования в веб-хранилище (например, на веб-сайте)

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

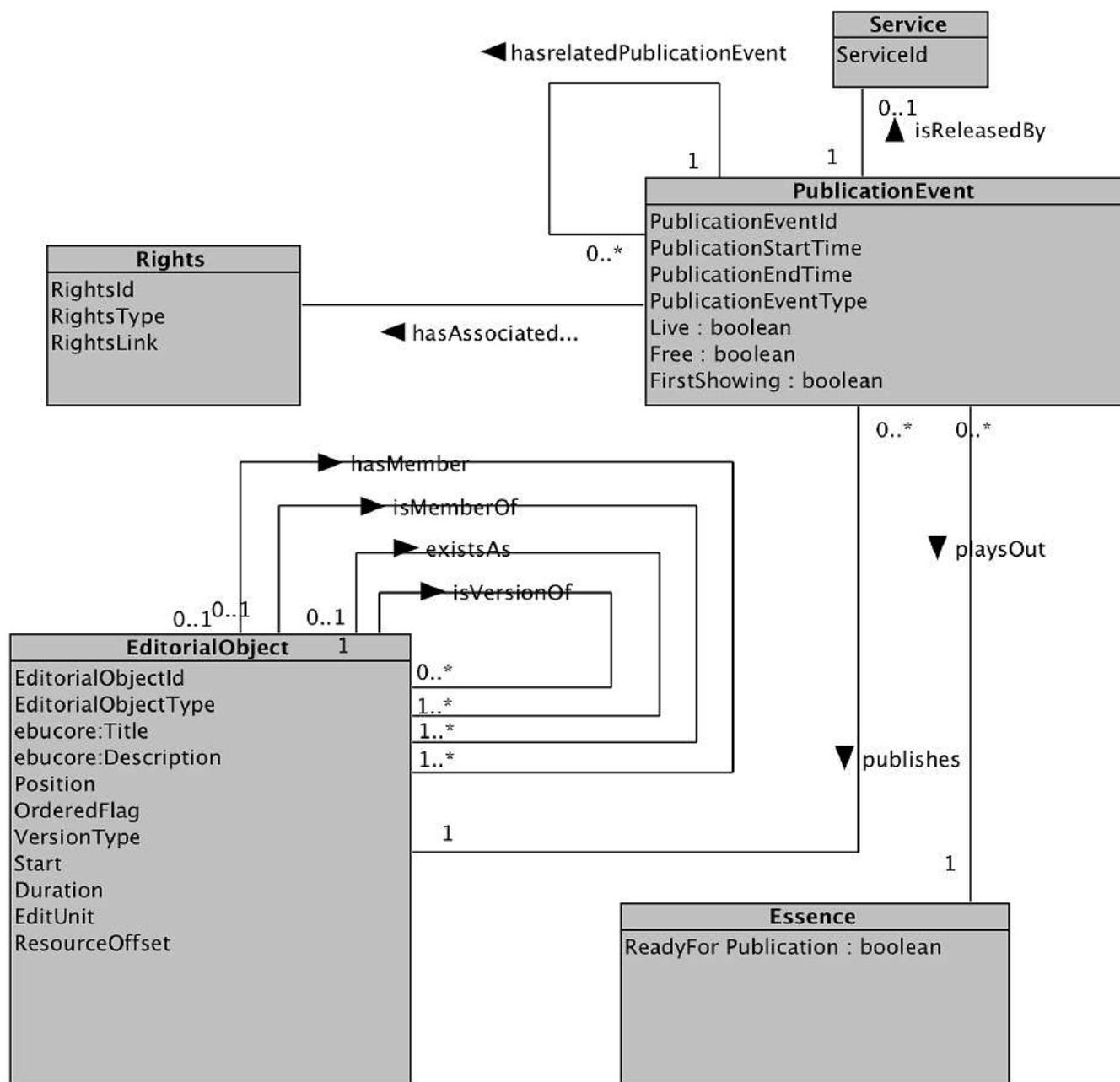


Рис. 6: Событие публикации

### 2.2.5.1 PublicationEvent (Событие публикации)

Определение:

Публикация *EditorialObject* для потребления пользователями путем обеспечения доступа к сущности как представлению ее *MediaResource*.

Пример:

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.

| Отношения класса                  |  |
|-----------------------------------|--|
| <i>publishes</i>                  | Отношение с <i>EditorialObject</i> , представляющим сюжет для публикации.  |
| <i>playsOut</i>                   | Для упорядоченной публикации привязанной ко времени последовательности <i>MediaResource</i> / <i>Essence</i> как <i>TimelineTrack</i> в <i>EditorialObject</i> .                                 |
| <i>hasAssociatedRights</i>        | Для идентификации прав, напрямую связанных с <i>PublicationEvent</i> , в дополнение к предполагаемым правам, связанным с <i>EditorialObjects</i> , <i>MediaResources</i> и/или <i>Essences</i> . |
| <i>hasrelatedPublicationEvent</i> | Для установления связи между двумя <i>PublicationEvents</i> (например, связь заказного события, включаемого из события вещания).   |

|                               |  |
|-------------------------------|--|
| <i>isReleasedBy</i><br>И т.д. | Канал или платформа услуг, выпускающая контент.<br>С <i>PublicationEvent</i> могут быть связаны другие отношения класса. См., например, ETSI TS 102 822 (TV-Anytime) или BBC Programme Ontology.               |
| <b>Свойства класса</b>        |  |
| <i>PublicationEventid</i>     | Идентификатор, связанный с <i>PublicationEvent</i> .   |
| <i>PublicationStartTime</i>   | Время, когда по расписанию должна начинаться программа или когда контент делается доступным или может потребляться.  |
| <i>PublicationEndTime</i>     | Время, когда программа по расписанию должна заканчиваться или после которого контент недоступен или не предназначен для потребления.   |
| <i>PublicationEventType</i>   | Тип события публикации, например, публикация в сети или прогон на радио  |
| <i>Live</i>                   | При установке флажок указывает, что контент должен быть маркирован как «прямой».   |
| <i>Free</i>                   | При установке флажок указывает, что контент доступен или может потребляться без подписки.  |
| <i>FirstShowing</i>           | При установке флажок указывает, что контент доступен в данном <i>PublicationChannel</i> . Это просто индикация, коллекция <i>PublicationEvents</i> одной <i>Essence</i> расскажет реальную историю публикации. |
| И т.д.                        | Для определения <i>PublicationEvent</i> можно использовать много других свойств. См., например, ETSI TS 102 822 (TV-Anytime) или BBC Programme Ontology.   |

### 2.2.5.2 Service (Услуга)

Определение:

“Service” – это канал или платформа публикации, выпускающие контент для данной аудитории.

|                                    |   |
|------------------------------------|---|
| <b>Отношения класса</b>            |   |
| <i>hasRelatedService</i><br>И т.д. | Отношение с какой-либо связанной службой публикации.<br>С <i>PublicationEvent</i> могут быть связаны другие отношения класса. См., например, ETSI TS 102 822 (TV-Anytime) |
| <b>Свойства класса</b>             |   |
| <i>ServiceId</i><br>И т.д.         | Идентификатор, связанный с услугой.<br>Для определения <i>PublicationEvent</i> могут использоваться многие другие свойства. См., например, ETSI TS 102 822 (TV-Anytime)   |

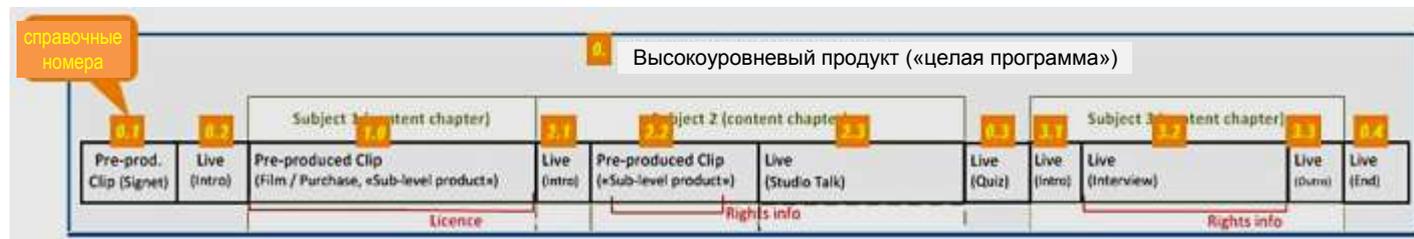
## 3. Принципы реализации / Вопросы и ответы

### 3.1 Общие замечания

Этот раздел содержит примеры текущих реализаций EBU CCDM и призван дать пользователям советы и пояснения, чтобы помочь им в реализации EBU CCDM в будущих версиях спецификации.

### 3.2 Примеры, предоставленные SRG SSR, Швейцарская Конфедерация

Ниже приведен пример программы под названием «идеальная программа»:



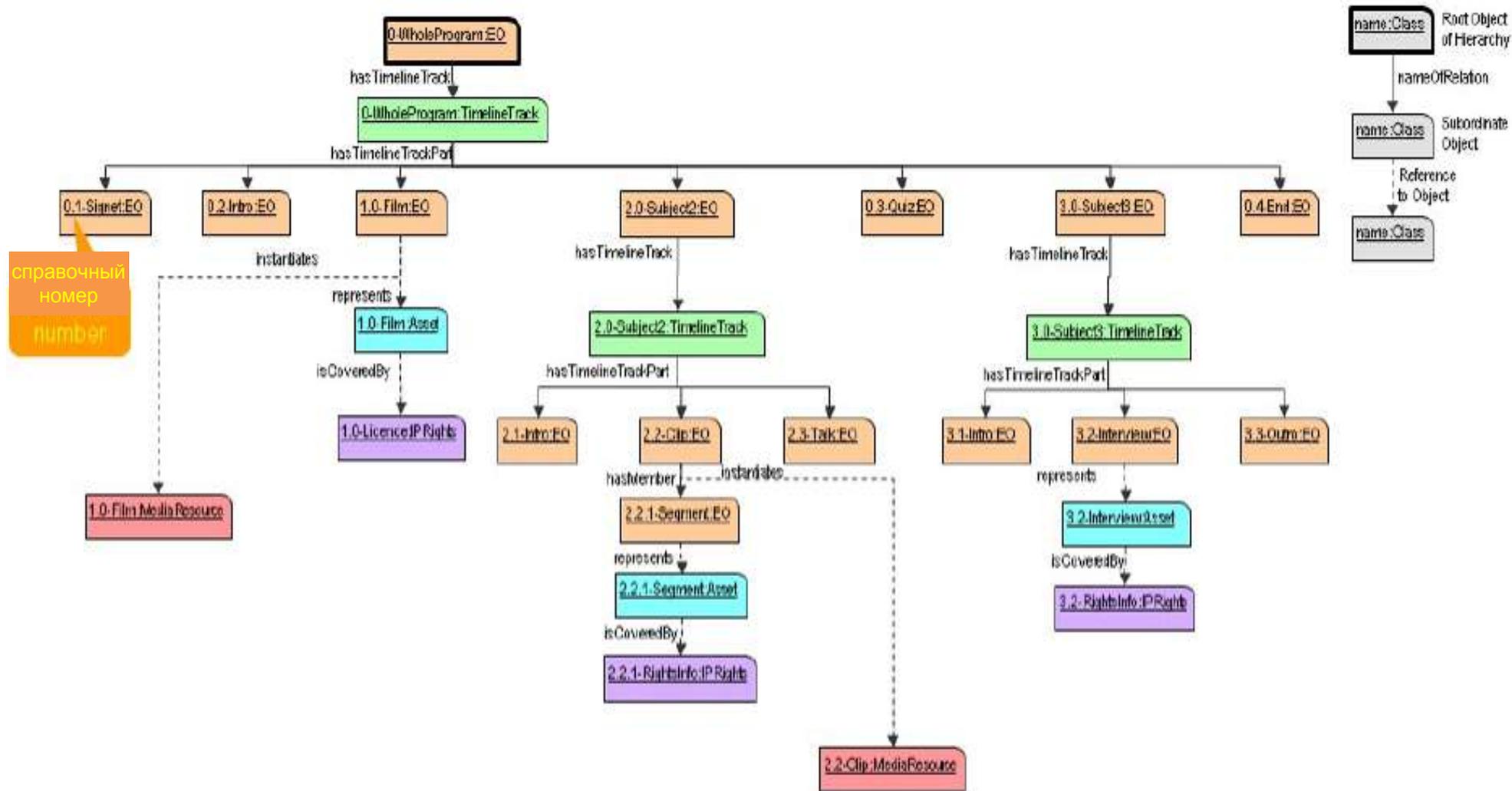
Теперь этот пример будет представлен с помощью CCDM. Представление зависит от точки зрения, которая прекрасно преобразуется в области, описанные в настоящем документе. Кроме того, следующие примеры предполагают разные сценарии публикации, такие как «прямой эфир» или «повтор». Некоторые примеры содержат объекты, прямо не представленные в схеме «идеальной программы», например, *ProductionDevices* Cam1 и Mic1.

Все эти предположения были сделаны лишь для демонстрации возможностей моделирования с CCDM.

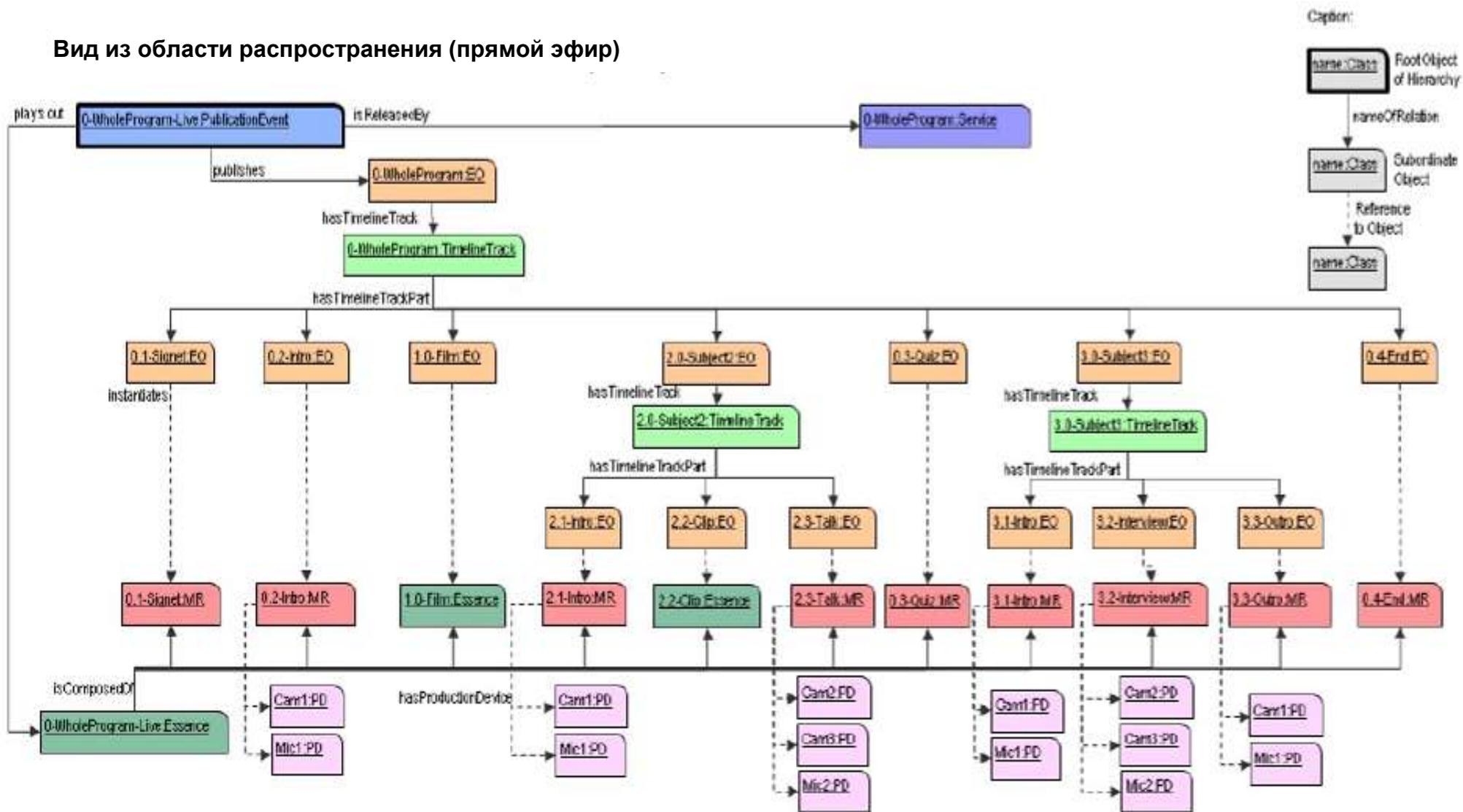
Съемы объектов представляют такую же иерархическую структуру, как в документе XML. Для подчеркивания иерархии необходимо ввести в иерархии «ссылки» (представленные пунктирными стрелками) помимо простых отношений объектов (представленных сплошными стрелками).

Следующие рисунки иллюстрируют, как моделировать «идеальную программу» с EBU CCDM 1.1.

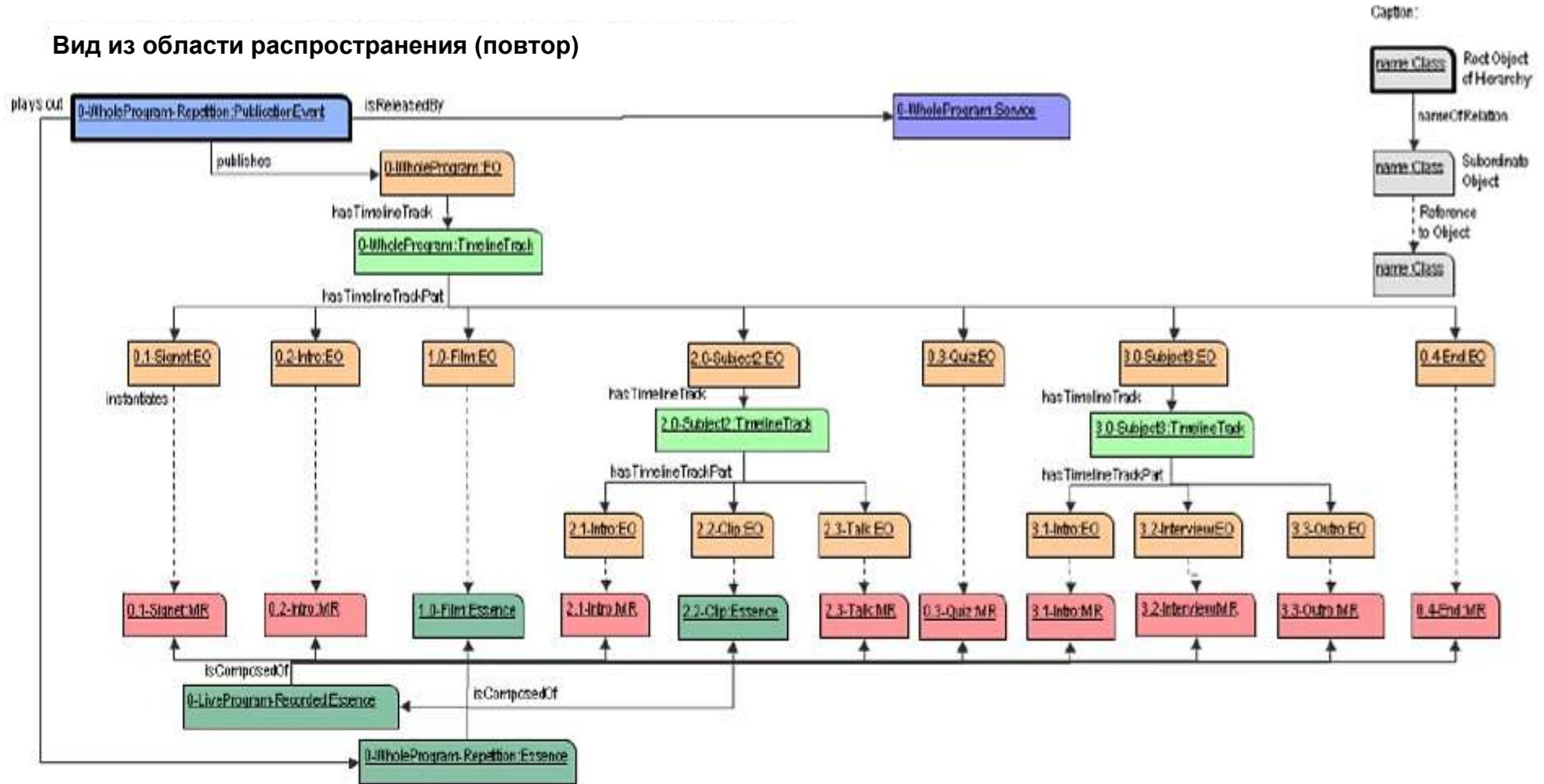
Вид из области редактирования



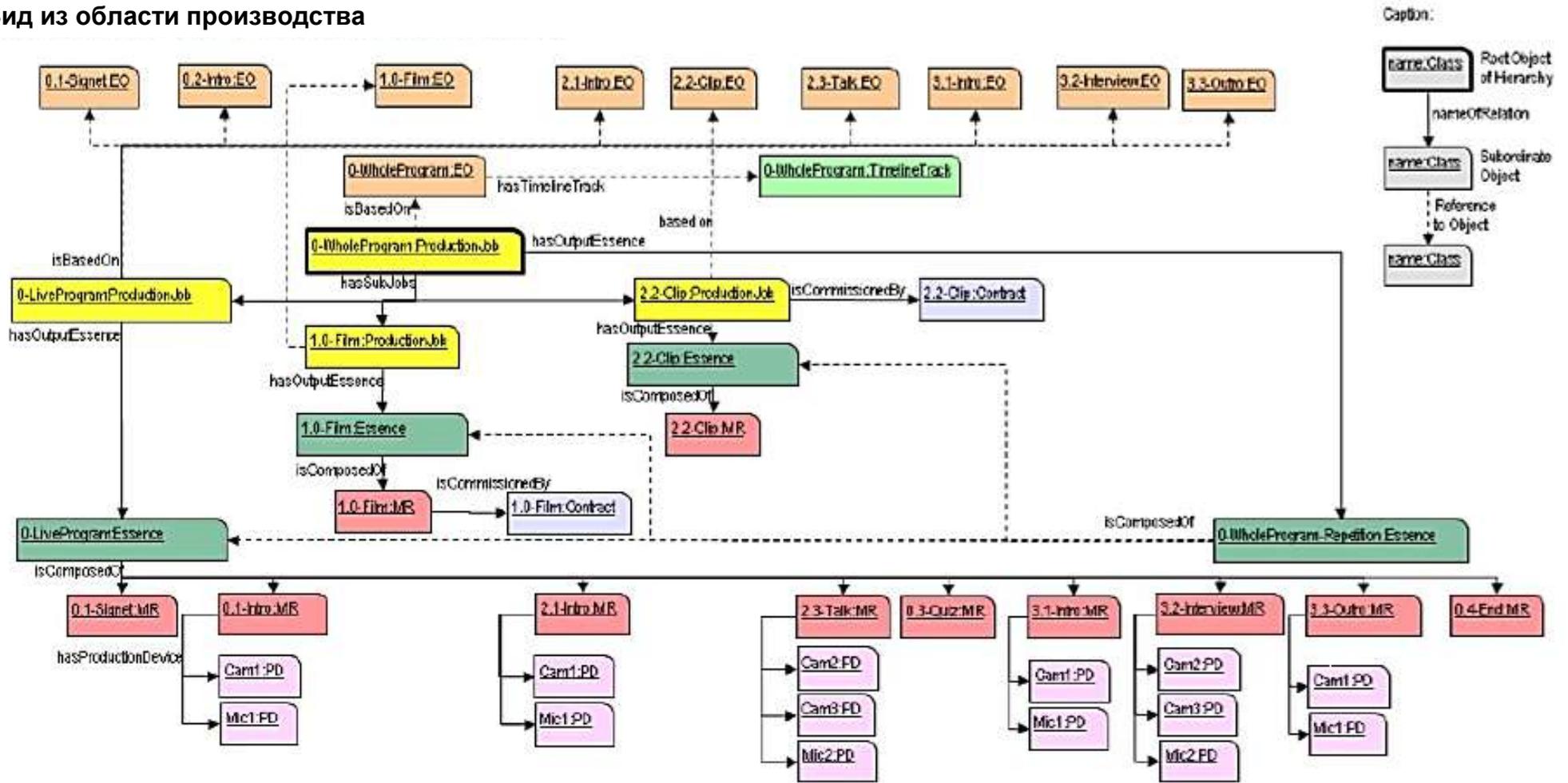
Вид из области распространения (прямой эфир)



Вид из области распространения (повтор)



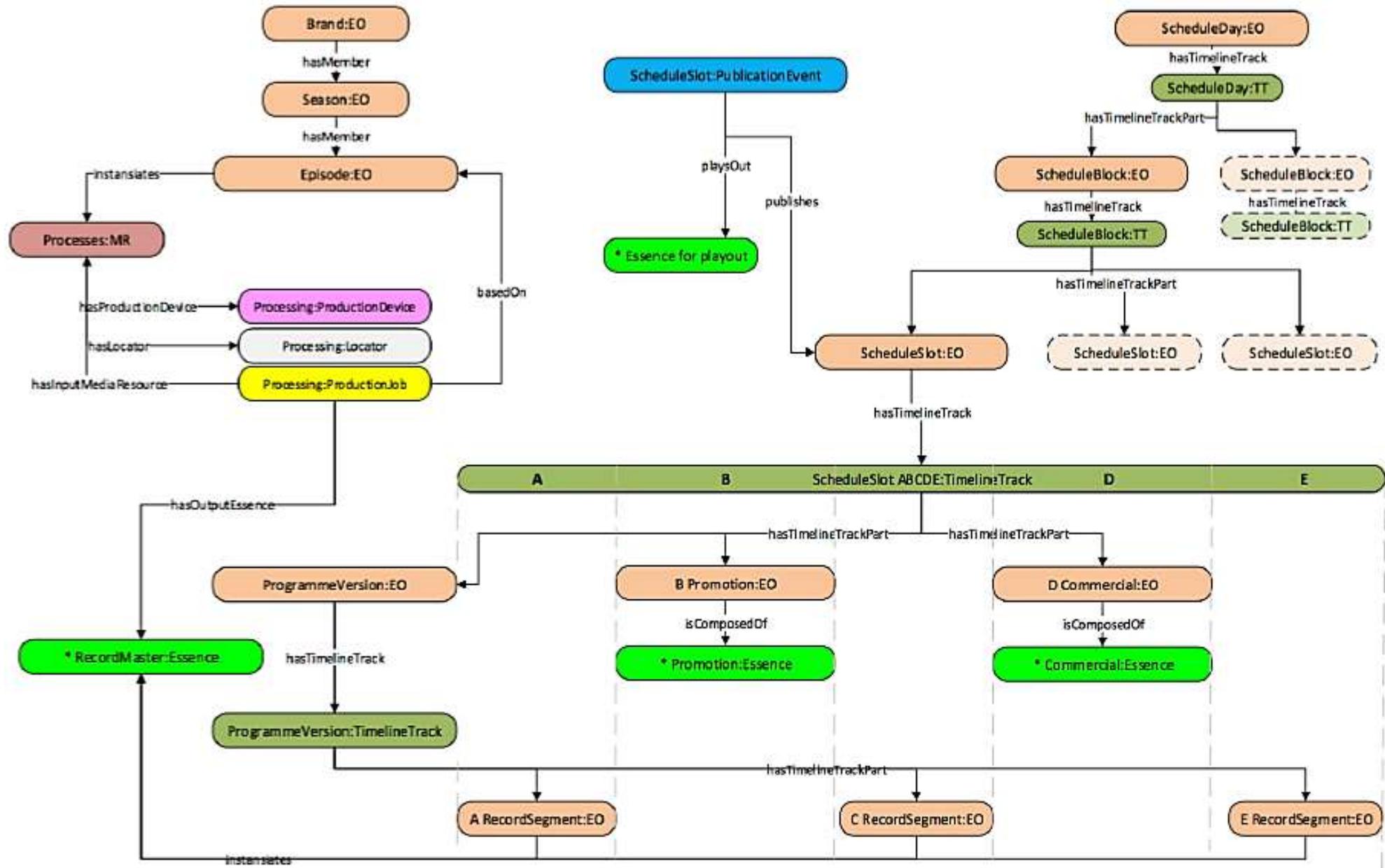
Вид из области производства



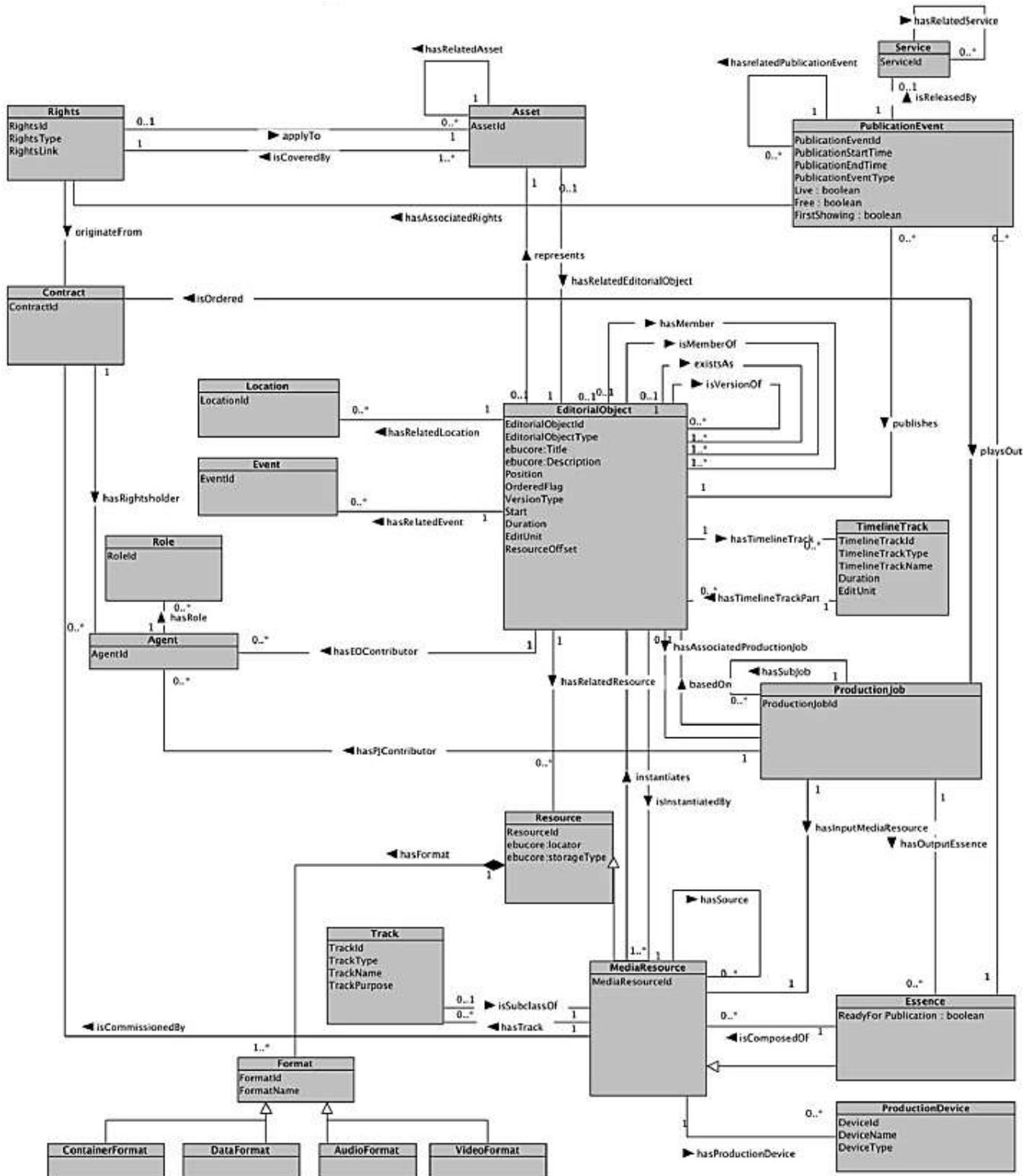


### 3.3 Пример, предоставленный TV2, Норвегия

#### Проект TV2 Norway GPO: расписание программы и рекламы



### 3.4 Общая диаграмма классов



### 3.5 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 с Вашего согласия ответы будут использованы для обогащения данного раздела в будущей версии спецификации.

## 4. 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 совместимые реализации должны содержать, расширять, но не заменять. Дополнительная информация об онтологии CCDM содержится в **Приложении А**.

## 5. Зона скачивания

| Имя файла и место   | Описание         |
|---|------------------|
| <a href="https://www.ebu.ch/metadata/ontologies/ebuccdm/">https://www.ebu.ch/metadata/ontologies/ebuccdm/</a><br><a href="https://www.ebu.ch/metadata/ontologies/ebuccdm/">https://www.ebu.ch/metadata/ontologies/ebuccdm/</a><br><a href="https://www.ebu.ch/metadata/ontologies/ebuccdm/">https://www.ebu.ch/metadata/ontologies/ebuccdm/</a> | Документация RDF |
| <a href="https://www.ebu.ch/metadata/ontologies/ebuccdm/ebuccdm.rdf">https://www.ebu.ch/metadata/ontologies/ebuccdm/ebuccdm.rdf</a>   | Файл RDF / XML   |

## 6. 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 произведение, *ремикшировать* – адаптировать произведение, в т.ч. в собственном пространстве имен при следующих условиях:

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

## 7. Обслуживание

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

## 8. Полезные ссылки

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

EBUCore (<http://tech.ebu.ch/publications/tech3293>)

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

TV-Anytime (<http://www.etsi.org>, скачивание стандарта в серии TS 102 822)

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

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

W3C- Resource Description Framework (<http://www.w3.org/TR/rdf-primer/>)

W3C - Web Ontology Language (<http://www.w3.org/TR/owl2-primer/>)

## Приложение А: Онтология EBU CCDM

Опорная программная реализация CCDM дана в RDF/OWL.

Ссылка для скачивания дана в §5 («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))