EBU

OPERATING EUROVISION

TECH 3351

EBU CLASS CONCEPTUAL DATA MODEL (CCDM)

Source: MIM

Version 1.1

Geneva
January 2016

* Page intentionally left blank. This document is paginated for two sided printing

Introduction

'The EBU Class Conceptual Data Model (CCDM) is an ontology defining a basic set of classes and properties as a common vocabulary to describe business objects, e.g. programmes, articles and other types of content, and their relations in the business processes of media enterprises. Examples are programmes in their different phases of creation from commissioning to delivery, their associated rights or publication events, etc.

CCDM is a common framework and users are invited to, and should, further enrich the model with classes and properties fitting their needs more specifically. Properties for describing each of the objects can be found in EBUCore, or you are welcome to define your own.

This is version 1.1 of the “CCDM”.

The CCDM has been purposefully designed as a minimum and flexible set of classes for a wide range of broadcasting applications, including archives, exchange and media service oriented production, semantic web and linked data.

The CCDM specification combines several aspects from existing models and specifications into a common framework. It has been built over several EBU attempts to represents broadcasting as a simple logical model. It has benefited from EBU work in metadata modelling (P-META and EBUCore) and semantic web developments. The distribution part has been designed to seek maximum mapping to TV-Anytime and the "BBC Programmes Ontology".

The CCDM ontology is represented in RDF/OWL and associated class diagrams.

More information on EBU metadata activities is provided on the EBU TECHNICAL website (**http://tech.ebu.ch/metadata**).

Terms and Conditions of Use

Page intentionally left blank. This document is paginated for two sided printing

Contents

Page intentionally left blank. This document is paginated for two sided printing

EBU Class Conceptual Data Model (EBU CCDM)

EBU Committee	*First Issued*	*Revised*	*Re-issued*
MIM	October 2012	January 2016	

Keywords: Class, Model, Metadata, Business, Object, Radio, Television, Production, SOA, Semantic Web, Linked Data, Internet, Web Publishing.

1. Scope

The EBU Class Conceptual Data Model (CCDM) is an ontology defining a basic set of Classes and properties as a common vocabulary to describe business objects in their different phases of creation from commissioning to delivery. CCDM is a common framework and users are welcome to further enrich the model with Classes and properties fitting their needs more specifically.

The CCDM has deliberately been designed as a minimum and flexible set of classes for a wide range of applications including but not restricted to archives, exchanges, media service oriented production, broadcasting, Internet delivery, Semantic Web modelling and Linked Open Data (LOD).

This specification is a class model, an ontology, and not a metadata specification. Metadata properties and datatypes (other than the relationships between Classes) are **indicative**. Users willing to adapt the CCDM model to their needs are invited to describe CCDM classes and custom extensions either using properties from EBU Tech 3293 (EBUCore metadata set) or other metadata specifications (e.g. TV-Anytime or in-house metadata schemes).

The CCDM specification is combining several aspects from existing models and specifications into a common framework. It has been built over several EBU attempts to represents broadcasting as a simple logical model. It has benefited from EBU work in metadata modelling (P-META and EBUCore) and semantic web developments. The distribution part has specifically been designed to seek maximum mapping to TV-Anytime and the "BBC Programmes ontology".

The CCDM ontology is represented in RDF/OWL.

1.1 Rationale

It is vital for content providers and broadcasters to have a well-defined class model. This is a necessary step towards:

- Greater understanding of the business models and workflows;
- Process optimisation with easier and more reliable data exchange;
- A simpler and rationalised description of Media Classes;
- The easier implementation of media service oriented production architectures;
- The adoption of new information management models such as Semantic Web and Linked Data (enrichment, improved searching and ubiquity).

The CCDM has been designed to let implementers adapt the names of the Classes and their Relationships to their respective modelling needs. Each organisation is encouraged to make its proper analysis and to create its own model starting from the CCDM framework as a common basis for comparison with models from other CCDM implementers.

2. Class Conceptual Data Model

2.1 Main principles

The EBU CCDM is composed of:

- Classes: directly related (e.g. a programme, a part, a clip, a track) or associated (e.g. a person, a location) to media.
  - Note: equivalent to the notion of class used in semantic web modelling (see RDF and OWL Primers), also referred to as 'Business Objects' or 'concepts' in certain projects, see also http://protege.stanford.edu/publications/ontology_development/ontology101.pdf . W3C's Media-Ontogy (MA-ONT) is based on the CCDM class model (http://www.w3.org/ns/ma-ont.rdf).
- Relationships: linking Classes (e.g. 'Programme hasContributor Person')
  - Note: equivalent to the notion of *objectProperties* used in semantic web modelling (see RDF and OWL Primers)
- Properties: defining intrinsic characteristics of Classes (e.g. 'bitrate' expressed as an integer or a person 'name' expressed as a string)
  - Note: equivalent to the notion of *dataProperties* used in semantic web modelling (see RDF and OWL Primers)

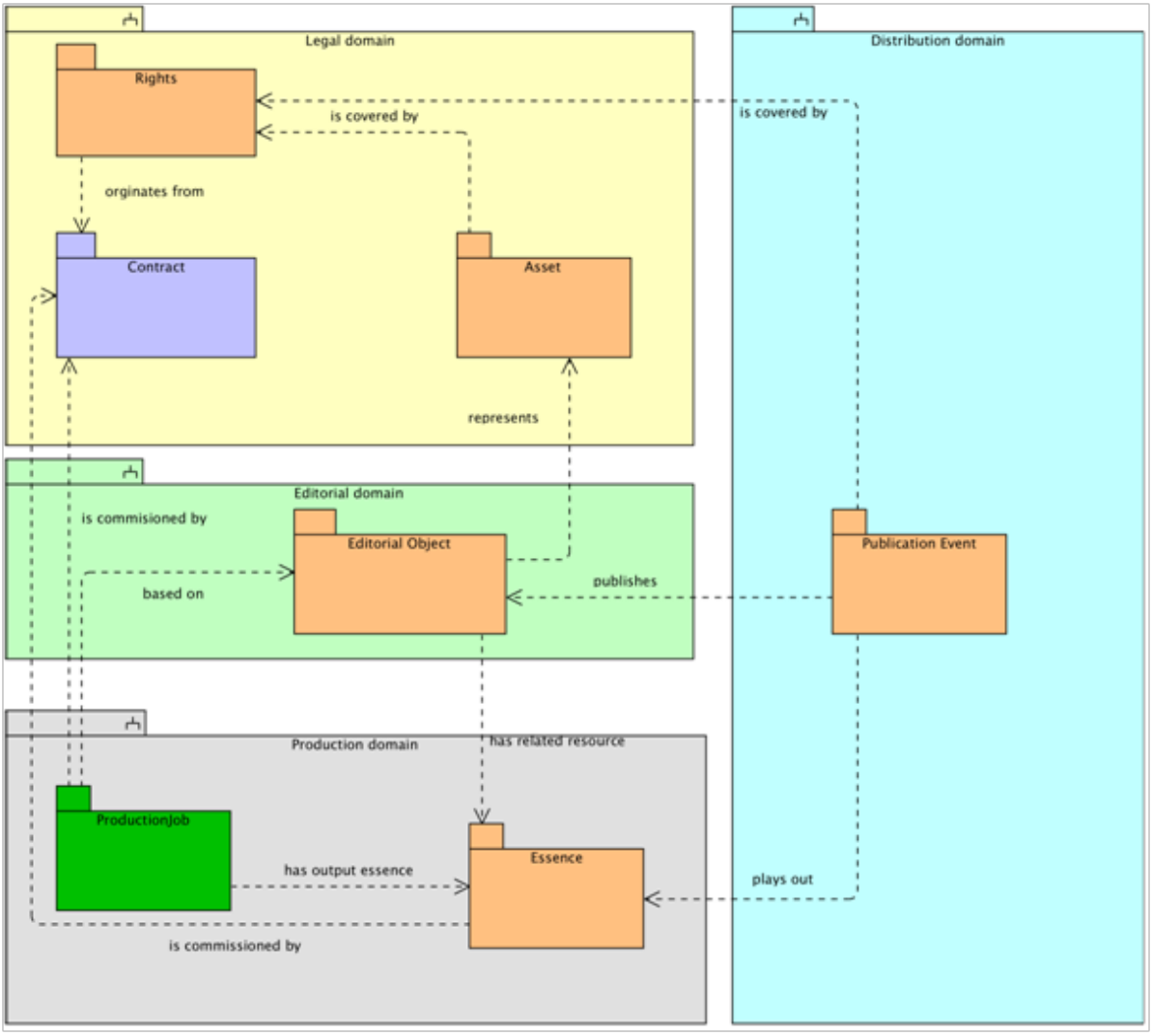


Figure 1: CCDM domains

As shown in **Figure 1**, the model is defined around four main domains:

- **Legal Domain** is where commissioning, production orders, intellectual property and other rights associated to content and its manifestations are being managed. Once the idea of a programme has been developed and agreed, its production is commissioned and resources are identified and reserved through a Contract commissioning a Production Job. The central Class of the Legal Domain is the Asset, which establishes the association of an *EditorialObject*, Intellectual Property and Rights related information.
- **Distribution Domain** is where any form of publishing, play-out or distribution is covered. The central Class is the *PublicationEvent* that plays out an Essence, i.e. the media object that was the result of the Production job.
- **Editorial Domain** is where concept related and content related information is being managed. Furthermore, all editing decisions are represented here. The Editorial Object is the central class of the domain. It can be grouped and it can be ordered on a timeline.
- **Production Domain** is where production orders are realised through the acquisition of *MediaResource* (e.g. manufacturing an object through the Production Job, purchase or retrieval of material) according to the production plan. *MediaResources* ready for publication use the Essence Class for connecting the content to a certain publication.

The EBU CCDM has been designed to let users adapt the names of Classes and relationships to their respective modelling needs. For example a Class '*EditorialObject*' can be of type 'programme', 'item' or shot, but it can also represent a group 'series', 'serial' or 'season'. The definition of appropriate Properties is left to the user. A core set of properties is proposed in EBU Tech 3293, EBUCore, or in other metadata specifications (e.g. TV-Anytime or in-house metadata schemes).

2.2 Classes, Relationships and Properties

See **Figure 2**, overleaf, which illustrates the relationships between domains and objects.

2.2.1 Legal domain

It is the domain in which intellectual property, rights and contracts are being managed and associated to a *MediaResource* and / or an *EditorialObject*, and by inference to a *PublicationEvent* (inc. exploitation and distribution conditions), to define an Asset. The domain also covers and commissioning of productions and material.

The central Class of the Legal Domain is the Asset that acts like a conjunction between a set of rights and an Editorial Object.

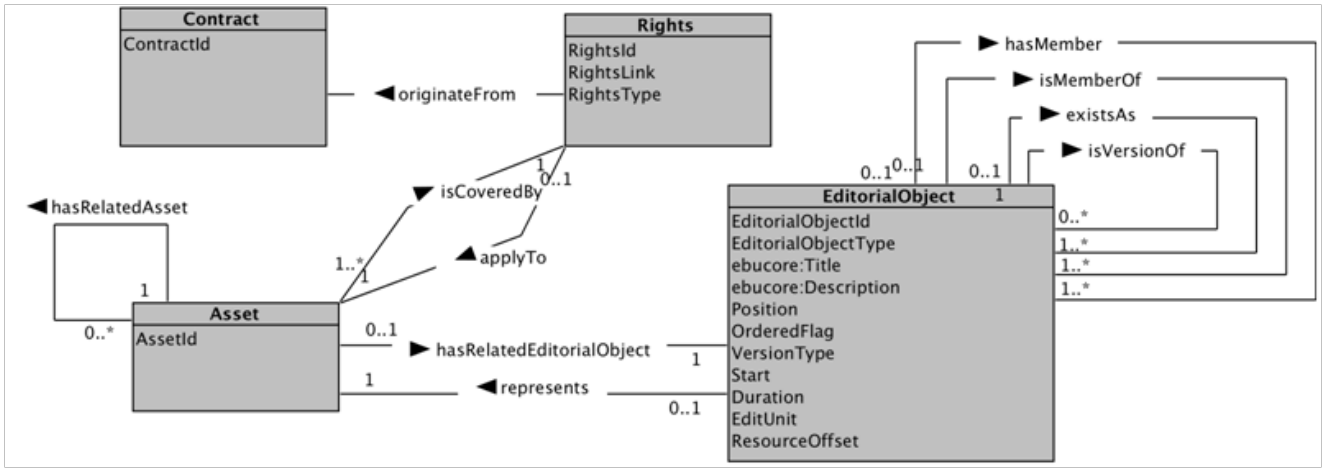


Figure 3: The Asset

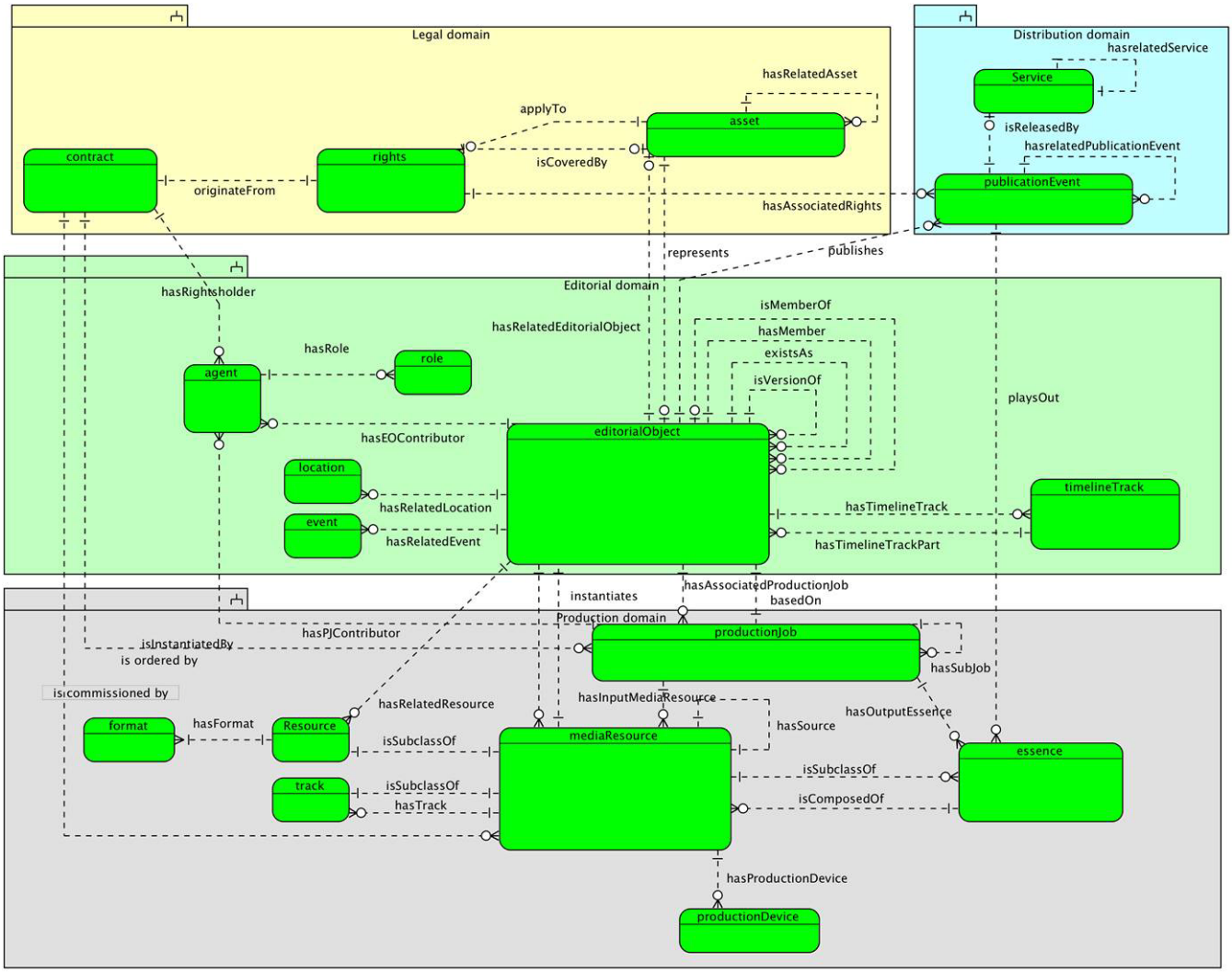


Figure 2: Domains, objects and relations

2.2.1.1 Asset

Definition:

The Class "Asset" is an object to which an identifier will be associated at commissioning. It will serve as a central reference point to manage rights associated to *EditorialObjects*, *MediaResources* or Essences, and - by inference - *PublicationEvents* (distribution and exploitation conditions).

Remember that the *MediaResources* or Essences will in this model always be represented by an *EditorialObject*.

Example:

The CCDM model allows the association of Rights to a related *EditorialObject* that represents Essence.

Class Relationships	
hasRelated*EditorialObject*	A pointer to the *EditorialObject* that the Asset links to its Rights
hasRelatedAsset	A pointer to another asset (e.g. a TV Series) that the Asset links to
isCoveredBy	A pointer to the Rights associated to the *EditorialObject*
Etc.	Other Class Relationships can be associated to an Asset. See EBU Tech 3293, EBUCore.
Class Properties	
AssetID	An Identifier associated with the Asset
Etc.	Other properties can be associated to an Asset. See EBU Tech 3293, EBUCore.

2.2.1.2 Rights

Definition:

The Class "Right" defines rights that originate from a contract. The Right is associated to *MediaResource* through the definition of an Asset. The *RightsHolder* is connected by the Contract.

Class Relationships	
applyTo	A pointer to the Asset, which in turn has *EditorialObject*, to which the Rights apply.
orginateFrom	A pointer to the contract granting the rights
Etc.	Other Class Relationships can be associated to Rights. See EBU Tech 3293, EBUCore
Class Properties	
RightsID	An Identifier associated with the Rights.
RightsType	A type associated to Rights e.g. licensing terms.
RightsLink	A link to e.g. a web resource where the rights terms can be found.
Etc.	Other properties can be associated to Rights. See EBU Tech 3293, EBUCore.

2.2.1.3 Contract

Definition:

The Class "Contract" represents any legal document covering right - or commissioning issues. This object/class covers the production order and sales order combined. The contract connects the rights to any *RightsHolders*. A contract defines one of more set of rights.

Class Relationships	
hasRightsHolder	A pointer to an Agent holding the Rights.
Etc.	Other Class relationships can be associated to Rights. See EBU Tech 3293, EBUCore
Class properties	
ContractID	An Identifier associated with the Contract
Etc.	Other properties can be associated to Rights. See EBU Tech 3293, EBUCore.

2.2.2 Editorial Domain

The Editorial Domain is the domain within which a commissioned concept is defined before fabrication and distribution. All metadata related to the idea of a programme (e.g. content, format, purpose, audience, schedule window), related to the content of the programme (e.g. titles, subjects, contributors, locations, events) and all editing decisions are represented in the respective classes.

The central Class in the Editorial Domain is the *EditorialObject*.

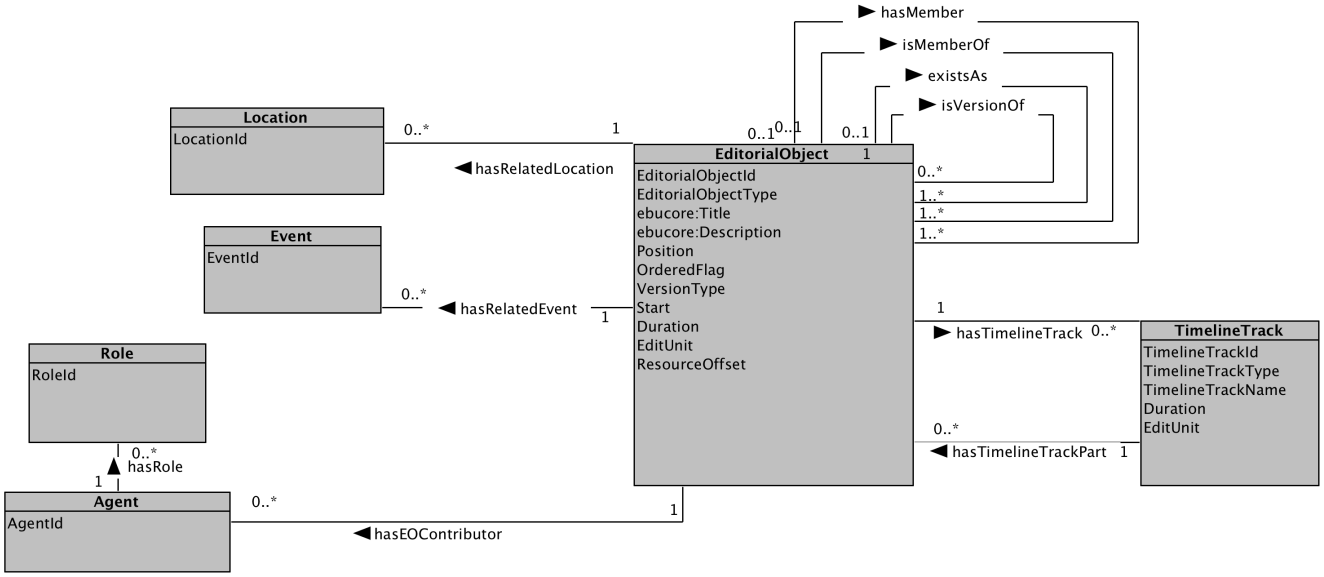


Figure 4: Classes around the *EditorialObject*

2.2.2.1 EditorialObject

Definition:

The Class "*EditorialObject*" describes any idea, any story and will be used to transform a concept into an editorial definition of a *MediaResource* before fabrication (Production Domain) and Distribution (Distribution Domain). An "*EditorialObject*" is a set of descriptive metadata summarising e.g. editing decisions.

An "*EditorialObject"* can be a group.

An "*EditorialObject*" can also be a part of another "*EditorialObject*", which is defined by its start time and duration.

EditorialObjects can be ordered either as groups or as items on a timeline.

Examples:

Programme, item, shot, part, chapter, segment, and where the group properties are in use: series, serial, compilation, collection, item group, item block

A simplified use-case:

A TV news broadcast consists of two news items. One news item contains the last ten seconds of a one minute long interview taken from another source (i.e. 50'' > 60''). This could be modelled as follows:

- The *NewsBroadcast* is linked to a *MediaResource* using the instantiates-property
- The *NewsItems* are linked to the *NewsBroadcast* using a *TimelineTrack*.

- The *InterviewPart* is linked to the *NewsItem* using the *hasMember*-property. Start and Duration are properties within the *InterviewPart* indicating its appearance within the *NewsItem2.*
- The *InterviewPart* is linked to its original source using the *existsAs*-property
- The Interview instantiates a *MediaResource*, which in turn is linked from the *MediaResource* of the *NewsBroadcast* using the *hasSource*-property
- Representation of segmentation: *TimelineTracks* are preferred over *hasPart*-properties, when a rundown is needed, e.g. for playout.

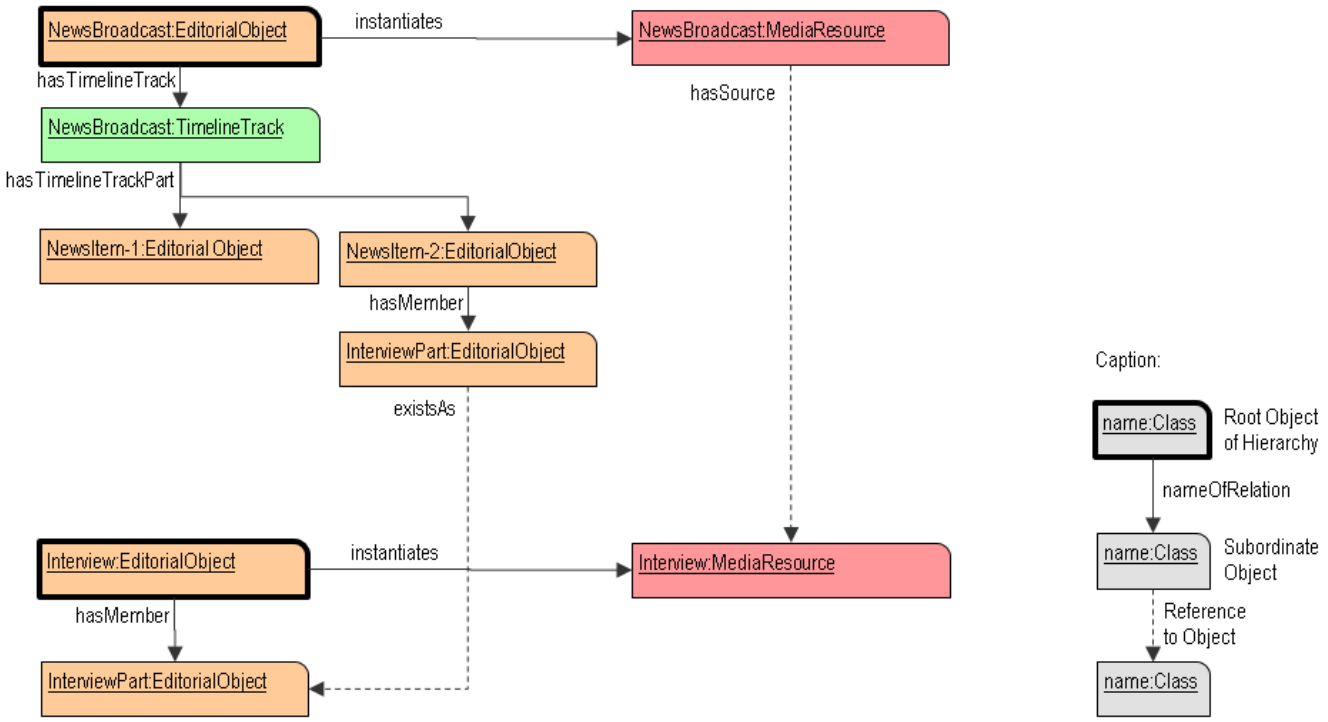


Figure 5: Illustration of use-case

Class Relationships	
isMemberOf	A list of Groups that the *EditorialObject* is a member of.
hasMember	A list of *EditorialObjects* that the *EditorialObject* contains that is not a part of a timeline. Series-episode is a example of such a relationship
hasRelatedResource	A relationship to identify a Resource that are related to the *EditorialObject*
isInstantiatedBy	A relationship to identify the *MediaResource* that instatiates the *EditorialObject*
hasEOContributor	The Agent(s) having contributed to the realisation of the *EditorialObject*. The contribution is characterised by the Agent Role. Agent is a non-media Class described in another section of this document. The "*hasEOContributor*" property can be extended with subproperties for different more specific roles, such as *hasEOCreator*, *hasEODirector*.
hasRelatedLocation	Optionally, one (or more) Location related to the *EditorialObject* characterised by its type (e.g. shooting or fictional).
hasRelatedEvent	Optionally, one (or more) Event related to the *EditorialObject* characterised by its type (e.g. sport event / meeting).
represents	An *EditorialObject* represents an Asset.
hasAssociatedProductionJob	A productionJob represents a production process through which an *EditorialObject* is being instantiated into a *MediaResource* and Essence.
isVersionOf	To identify *EditorialObjects* presenting alternative version of the content.
existsAs	To identify *EditorialObjects* representing alternative mediasets of the content
hasTimelineTrack	To associate a *TimelineTrack*, e.g. a *RunDown*, with an *EditorialObject* itself constituted of other *EditorialObjects*.
Etc.	Other Class Relationships can be associated to an *EditorialObject*. See EBU Tech 3293, EBUCore.
Class Properties	
EditorialObjectType	The type of *EditorialObject* e.g. Programme, Item.
EditorialObjectID	Optionally one (or more) identifier attributed to the *EditorialObject*.
ebucore:Title	The main Title by which of the *EditorialObject* is known. As an example, this property is imported from the EBUCore namespace.
ebucore:Description	Optionally one (or more) description of the *EditorialObject*. As an example, this property is imported from the EBUCore namespace.
Position	The position of the *EditorialObject* in an *EditorialObject* of type 'rundown', or in an ordered Group
VersionType	A string to optionally identify the version of the *EditorialObject* such as lengthened, shortened, signed, closed-captioned, etc.
Start	The starting point of the Member, i.e. the part, in an *EditorialObject* or in a *TimelineTrack*.
Duration	The duration of the Member in an *EditorialObject* or in a TimelineTrack.
EditUnit	The unit used to express start and duration.

ResourceOffset	The start offset of the related resource, used if the related resource is not used from its start.
OrderedFlag	If 'true', a flag which indicates that the members of the *EditorialObject* are ordered (e.g. membership is subject to a strict sequence such as episodes in a series).
Etc.	Many other Properties can be associated to an *EditorialObject*. See EBU Tech 3293, EBUCore.

2.2.2.2 TimelineTrack

Definition:

A "TimelineTrack" is used to define timelines, i.e. a time related sequence of *EditorialObjects* (or Part of *EditorialObjects*).

Class Relationships	
HasTimelineTrackPart	To identify the Parts of a *TimelineTrack*. I. e. *EditorialObjects* with a start time and duration.
Etc.	Many other Properties can be associated to an *EditorialObject*. See EBU Tech 3293, EBUCore.
Class properties	
TimelineTrackID	The identifier attributed to a *TimelineTrack*.
TimelineTrackType	E.g. rundown or other types not defined as subclass in the specification
Duration	The duration of the *TimelineTrack* in the *EditorialObject*.
EditUnit	The unit used to express the duration.
Etc.	Many other Properties can be associated to an *EditorialObject*. See EBU Tech 3293, EBUCore.

2.2.2.3 Location

Definition:

A "Location" is used to define the locations, e.g. spatial coverage of the story or recording locations like studios or in the field, associated with the *EditorialObjects* (or Part of *EditorialObjects*).

Class properties	
LocationId	To identify the location in a system of defined locations
Etc.	Many other Properties can be associated to a Location. See EBU Tech 3293, EBUCore (incl. GPS coordinates) or *GeoNames*.

2.2.2.4 Event

Definition:

An "Event" is used to define the event that the *EditorialObject* covers.

Examples:

Olympic Games 1994, General election, etc.

Class properties	
EventId	To identify the event
Etc.	Many other Properties can be associated to an *EditorialObject*. See EBU Tech 3293, EBUCore.

2.2.2.5 Agent

Definition:

An "Agent" is either a Contact/Person or Organisation to which is associated a Role corresponding to the contribution the "Agent" brings to the realisation of a *MediaResource* or *EditorialObject*.

Examples:

Examples of the Agent Role are 'producer' or 'cameraman' or 'actor'.

Class Relationships	
hasRole	The role of the Agent. Role refines "*hasContributor*". Alternatively, a user can decide to add new Class and associated Relationships as contributions to an *EditorialObject* e.g. *hasContributorCreator*, *hasContributorComposer* etc, which in turn will be refined with *hasRole* Role.
Etc.	Other Class Relationships can be associated to an Agent. See EBU Tech 3293, EBUCore.
Class Properties	
AgentId	An identifier for the Agent
Etc.	Other Class Properties can be associated to an Agent. See EBU Tech 3293, EBUCore.

2.2.2.6 Role

Definition:

The "Role" played by an Agent. A "Role" will be identified e.g. by a Concept from a SKOS Classification Scheme. "Role" is therefore to be considered as a Class, i.e. a subClass of SKOS Concept.

Example:

A Contact may be an actor.

Class Properties	
RoleId	Identifier for the role, preferable from a defined list of roles
Etc.	Other Class Properties can be associated to a Role. See EBU Tech 3293, EBUCore.

2.2.3 Production Domain

The Production Domain is the domain, within which production orders are realised through the acquisition of *MediaResource* (e.g. manufacturing an object through a Production Job, purchase or retrieval of material).

The central Class in the Production Domain is the *MediaResource* and its subclass the Essence.

MediaResources ready for publication use the Essence Class for connecting the content to a certain publication.

A *MediaResource* has always a relation to an *EditorialObject* (Editorial Domain) describing its content. The Essence is a manifestation of a *MediaResource* in a particular Format that are destined for publication. The Essence is the result of a *ProductionJob* and is a subclass of *MediaResource* and inherits the all of its properties such as Format, Location and *ProductionDevice*.

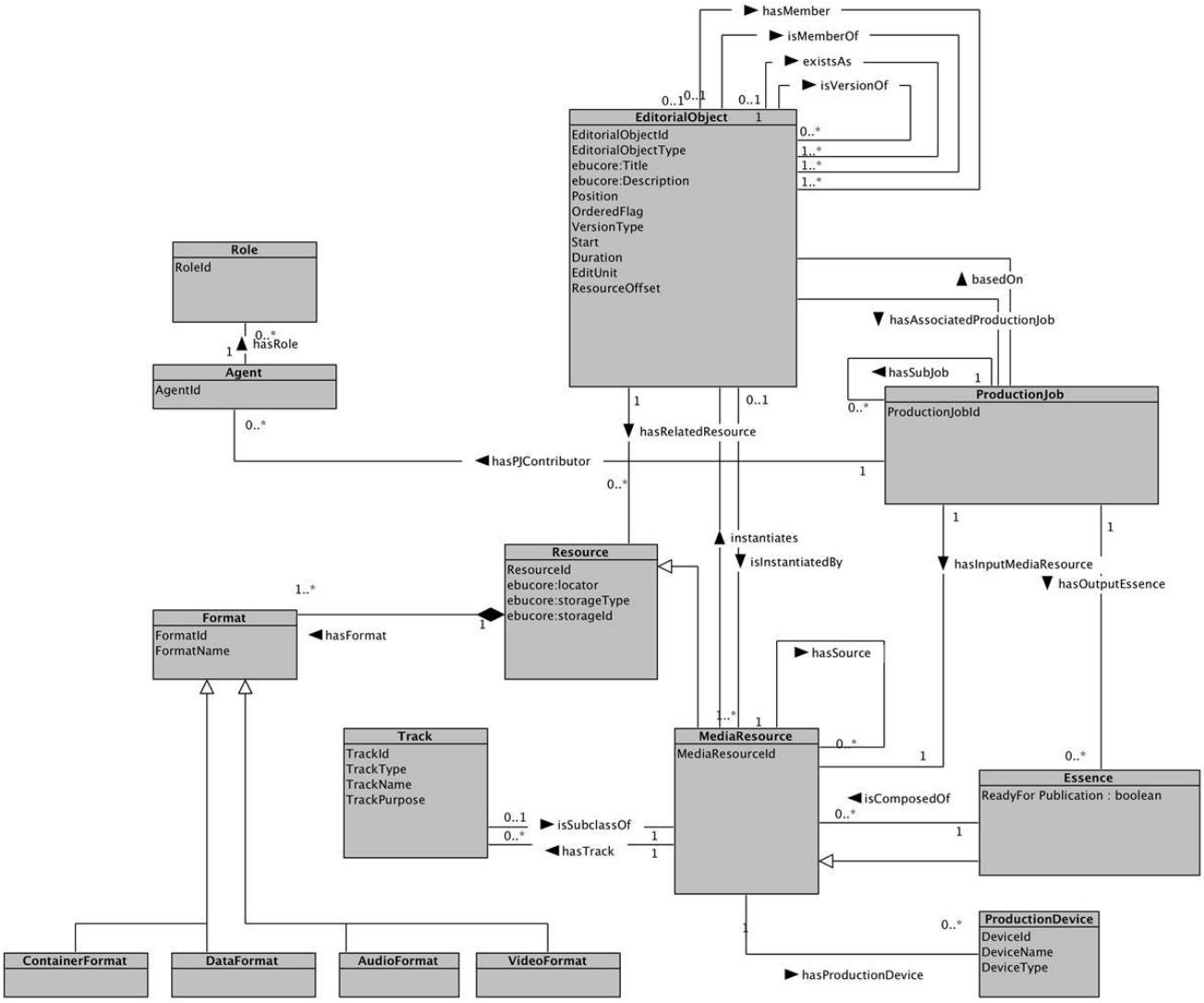


Figure 6: *MediaResource*

2.2.3.1 Resource

Definition:

"Resource" is a generic file used in relation to a production. It is defined by an *EditorialObject* (Editorial Domain). It has a locator indication where the resource can be retrieved.

Examples:

A pdf file used as part of the research, a manuscript stored in a repository etc.

Class relations	
hasFormat	The composition of a Resource. Resource can exist in one or more formats.
Etc.	Other Class Relationships can be associated to a *MediaResource*. See EBU Tech 3293, EBUCore.
Class Properties	
ResourceId	Unique Identifier e.g. a UUID, UMID, URI etc. It can be generated or assigned by the business process or it can be extracted from the content.
StorageId	The identifier of the storage
StorageType	A definition of the type / structure of storage where the Essence is stored.
Locator	This indicates where a particular *Resource* can be found and accessed
Etc.	Many other Properties can be associated to a *MediaResource*. See EBU Tech 3293, EBUCore.

2.2.3.2 MediaResource

Definition:

"*MediaResource*" is commissioned for production. It is defined by an *EditorialObject* (Editorial Domain). It can be fabricated in one or more Formats, in turn instantiated in one or more instances of Essence. It can be represented by one or more Essence e.g. in a particular Format for distribution on a specific delivery media. The *MediaResource* is a subclass of Resource.

Many properties can be found in the format part of EBUCore for describing the technical metadata of the *MediaResource*

Class relations	
hasProductionDevice	The production device responsible for the creation of the *MediaResource*
hasSource	The relation to a *MediaResource* acting as a source of the *MediaResource*. E.g. a analogue tape that is the source of a file
hasTrack	The relation to the tracks that the *MediaResource* are divided into.
instantiates	Relation to the *EditorialObject* that describes the *MediaResource*.
Etc.	Other Class Relationships can be associated to a *MediaResource*. See EBU Tech 3293, EBUCore.
Class Properties	
MediaResourceId	Unique Identifier e.g. a UUID, UMID etc. It can be generated or assigned by the business process or it can be extracted from the content.
Etc.	Many other Properties can be associated to a *MediaResource*. See EBU Tech 3293, EBUCore.

2.2.3.3 Track

Definition:

A "Track" is part of a *MediaResource*, and is a subclass of *MediaResource*. A *MediaResource* is potentially composed of any combination of audio, video and data Tracks.

Examples:

Examples of video Tracks are different camera angles or an additional signing Track.

Examples of audio Tracks are stereo pairs, multichannel audio e.g. surround, international sound, etc.

Examples of data tracks: ancillary data, captioning, etc.

Class Relationships	
Etc.	Other Class Relationships can be associated to a Track. See EBU Tech 3293, EBUCore.
Class properties	
TrackID	The identifier attributed to a Track.
TrackName	A name associated to a Track.
Etc.	Many other Properties can be associated to a Track. See EBU Tech 3293, EBUCore.

2.2.3.4 Format

Definition:

"Format" is a structure of technical metadata. A "Format" can be defined as the composition of audio, video and or data components and the description of their respective formats. The *ContainerFormat* defines the file / package structure of the *MediaResource*.

Example:

A Format for an Audio *MediaResource* will define the audio encoding format, the sampling frequency, etc.

Often used subclasses	
subclass	*AudioFormat* is a sub-Class of Format, used to list all the characteristics of the audio signal. See e.g. 'audioFormat' in EBU Tech 3293, EBUCore for more information.
subclass	*VideoFormat* is a sub-Class of Format, used to list all the characteristics of the video signal. See e.g. 'videoFormat' in EBU Tech 3293, EBUCore for more information.
subclass	*DataFormat* is a sub-Class of Format, used to list all the characteristics of the data signal.
subclass	*ContainerFormat* is a sub-Class of Format, used to list all the characteristics of the container. It provides information on the container / wrapper format in complement to the stream encoding information provided in 'channel', (e.g. mp3, wave, Quicktime, ogg). See, e.g., 'containerFormat' in EBU Tech 3293, EBUCore for more information.
Class Properties	
FormatId	An identifier associated to the format.
FormatName	A name associated to the format.
Etc.	Many other Properties can be associated to a Format. See EBU Tech 3293, EBUCore.

2.2.3.4.1 AudioFormat

Definition:

A class to provide definitions about the "*AudioFormat*" (e.g. encoding format, sampling rate).

Class relations	
Etc.	Other Class Relationships can be associated to an Audio Format. See EBU Tech 3293, EBUCore. This standard defines the Audio Definition Model
Class Properties	
Etc.	Other data properties can be associated to an Audio Format. See EBU Tech 3293, EBUCore. This standard defines the Audio Definition Model

2.2.3.4.2 VideoFormat

Definition:

A class to provide definitions about the "*VideoFormat*" (e.g. encoding format, frame rate).

Class relations	
Etc.	Other Class Relationships can be associated to a *VideoFormat*. See EBU Tech 3293, EBUCore.
Class Properties	
Etc.	Other data properties can be associated to a *VideoFormat*. See EBU Tech 3293, EBUCore.

2.2.3.4.3 DataFormat

Definition:

A class to provide definitions about the "*DataFormat*" (e.g. captioning format).

Class relations	
Etc.	Other Class Relationships can be associated to a *DataFormat*. See EBU Tech 3293, EBUCore.
Class Properties	
Etc.	Other data properties can be associated to a *DataFormat*. See EBU Tech 3293, EBUCore.

2.2.3.4.4 ContainerFormat

Definition:

A class to provide definitions about the "*ContainerFormat*" (e.g. container type).

Class relations	
Etc.	Other Class Relationships can be associated to a *ContainerFormat*. See EBU Tech 3293, EBUCore.
Class Properties	
Etc.	Other data properties can be associated to a *ContainerFormat*. See EBU Tech 3293, EBUCore.

2.2.3.5 Essence

Definition:

The "Essence" is a physical representation of a *MediaResource* in a particular Format destined for play-out or publishing. "Essence is a subclass of a *MediaResource* and inherits the *MediaResource* properties. "Essence" can be available in a form of a simple file or complex packages (e.g. as delivered by cameras of different brands).

Examples:

An AAC file is an example of audio essence. A P2 file structure (audio, video clip, voice, icon, proxy directories) is an example of package.

Class relations	
isComposedOf	A list of *MediaResources* that composes the Essence.
Etc.	Other Class Relationships can be associated to an Essence. See EBU Tech 3293, EBUCore.
Class Properties	
ReadyFor Publication	A flag that is set if the Essence is ready for publication.
Etc.	Many other Properties can be associated to an Essence. See EBU Tech 3293, EBUCore.

2.2.3.6 ProductionJob

Definition:

The "*ProductionJob*" produces an Essence for publishing using *MediaResources* as inputs, based on an *EditorialObject* describing the process in detail. It is ordered by a contract.

Class relations	
basedOn	Relation to the *EditorialObject* that is produced by the job
hasSubJob	Relation to a breakdown of the *ProductionJob*, i.e. a separate task of a workflow.
hasInputMediaResource	A list of *MediaResources* that are used for composing the Essence.
hasOutputEssence	Relation to the essence that is the result of the job.
hasPJContributor	Information about crew etc
isOrdered	Relation to the Contract that orders the *ProductionJob*
Etc.	Other Class Relationships can be associated to an Essence. See EBU Tech 3293, EBUCore.
Class Properties	
ProductionJobId	Identifier for the *ProductionJob*
Etc.	Many other Properties can be associated to an Essence. See EBU Tech 3293, EBUCore.

2.2.3.7 ProductionDevice

Definition:

A "*ProductionDevice*" is a device used during the *ProductionJob*.

Example:

An example of a *ProductionDevice* is a tapeless camcorder.

Class Relationships	
Etc.	Other Class Relationships can be associated to a *ProductionDevice*.
Class Properties	
DeviceId	An identifier associated to a *ProductionDevice*.
DeviceType	The type of the *ProductionDevice* e.g. a camcorder.
DeviceName	The name of the *ProductionDevice*.
Etc.	Many other Class Properties can be associated with a *ProductionDevice*. Examples of additional properties for a camcorder can be found in EBU Tech 3349 (Acquisition Metadata).

2.2.4 Distribution Domain

The Distribution Domain covers any form of publishing, play-out or distribution.

The central Class is the *PublicationEvent* that plays out an Essence, i.e. the media object that was the result of the Production job.

Other classes can be added to suit a specific need in play-out or distribution.

A *PublicationEvent* can be, for example:

- A broadcast event, i.e. an isolated event such as for last minutes news reports, etc. This content can be available via over the air broadcast or streaming.
- A scheduled event, i.e. each event being identified in a particular timeslot. This content can be available via over the air broadcast or streaming.
- An on-demand event, i.e. content is made available for immediate viewing or for download. It generally has a certain window of time availability. Catch-up TV is considered as an on-demand event. On-demand events can also be linked to broadcast and schedule events.
- An on-line event, i.e. content is made available for download/fruition on some web repository (e.g. on a web site)

According to the type of *PublicationEvent*, *MediaResource* is available in different Formats instantiated in Essence files or packages.

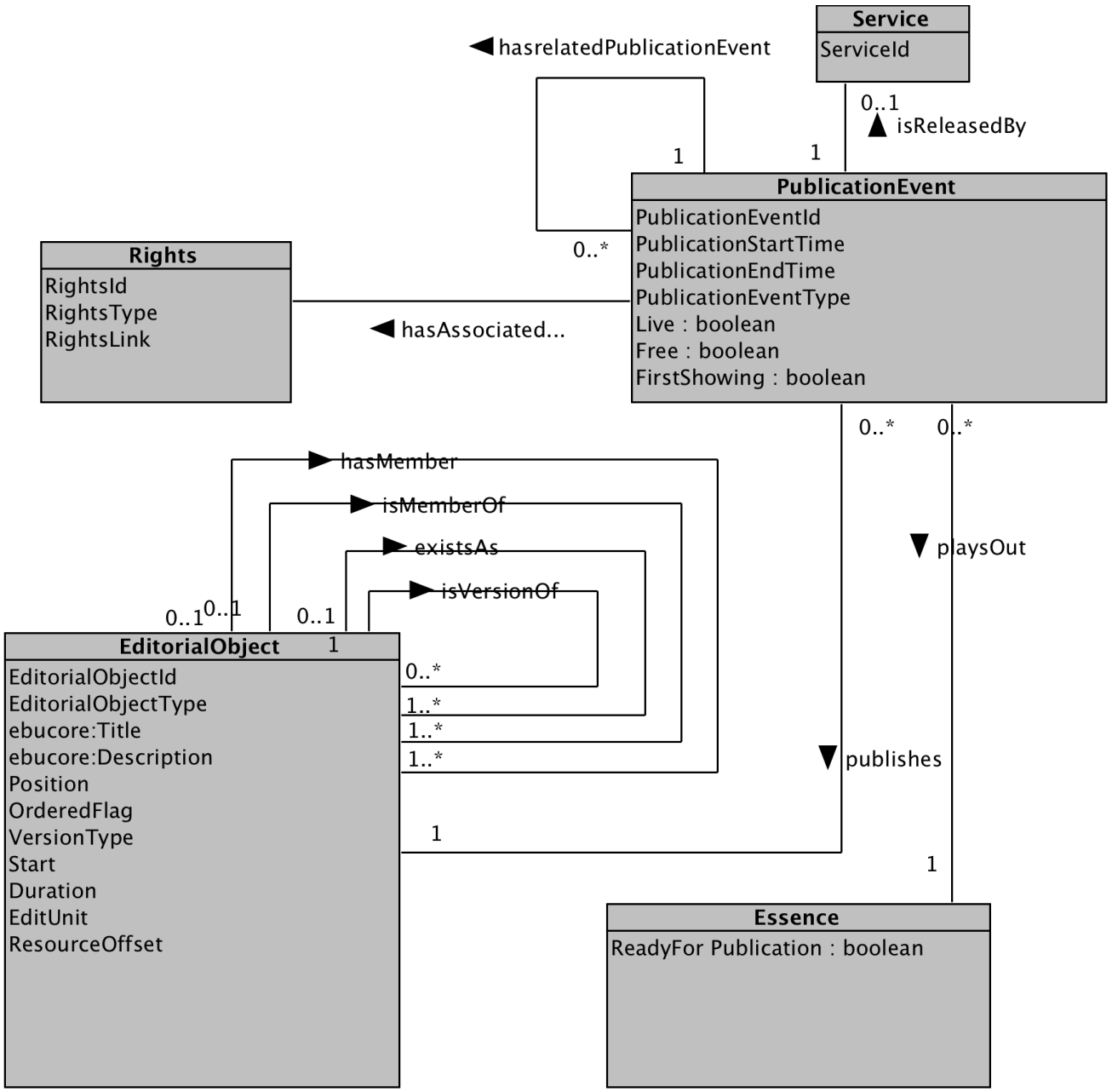


Figure 6 Publication Event

2.2.5.1 PublicationEvent

Definition:

Publishing an *EditorialObject* for user consumption by making available an Essence as a reprentation of its *MediaResource*.

Example:

An Event that is, for example, a scheduled Event i.e. a time slot in a schedule associated with a *PublicationChannel*. An Event can also be a broadcast Event not in a preliminary schedule, such as a live special news report. An Event can also be a streaming Event or a VoD publication Event.

Class Relationships	
publishes	A relation to an *EditorialObject* representing the story that will be published.
playsOut	To allow the ordered publication of a time related sequence of *MediaResource* / Essence as a *TimelineTrack* of an *EditorialObject*.
hasAssociatedRights	To identify the Rights directly associated with a *PublicationEvent* in addition to inferred rights associated with the related *EditorialObjects*, *MediaResources* and/or Essences.
hasrelatedPublicationEvent	To establish a link between two *PublicationEvents* (e.g. linking an on-demand event triggered from a broadcast event.
isReleasedBy	The channel or service platform that releases the content
Etc.	Other Class Relationships can be associated to a *PublicationEvent*. See e.g. ETSI TS 102 822 (TV-Anytime) or the BBC Programme Ontology.
Class Properties	
PublicationEventId	An identifier associated with the *PublicationEvent*.
PublicationStartTime	The time at which the programme is scheduled to start or when content is made available / can be accessed or consumed.
PublicationEndTime	The time at which the programme is scheduled to end or after which content is no longer available / accessible / consumable.
PublicationEventType	The type of the publication event, e.g. publishing on web or play-out on radio
Live	If set, a flag to indicate that the content should be marked as "Live".
Free	If set, a flag to indicate that content can be accessed / consumed without subscription.
FirstShowing	If set, a flag to indicate that this is the first time that this content is available on this *PublicationChannel*. This is just a indication, the collection of the *PublicationEvents* one Essence have will tell the real publishing history.
Etc.	Many other properties can be used to define a *PublicationEvent*. See e.g. ETSI TS 102 822 (TV-Anytime) or the BBC Programme Ontology.

2.2.5.2 Service

Definition:

A service is a channel or publishing platform that releases the content to a given audience.

Class Relationships	
hasRelatedService	Relation to some related publishing service.
Etc.	Other Class Relationships can be associated to a *PublicationEvent*. See e.g. ETSI TS 102 822 (TV-Anytime)
Class Properties	
ServiceId	An identifier associated with the Service.
Etc.	Many other properties can be used to define a *PublicationEvent*. See e.g. ETSI TS 102 822 (TV-Anytime)

3. Implementation Guidelines / Questions & Answers

3.1 General remarks

This section provides examples from current implementers of the EBU CCDM and is intended to provide advice and clarification for users to help them in implementing the EBU CCDM in future versions of the specification.

3.2 Examples provided by SRG SSR, Swiss Confederation

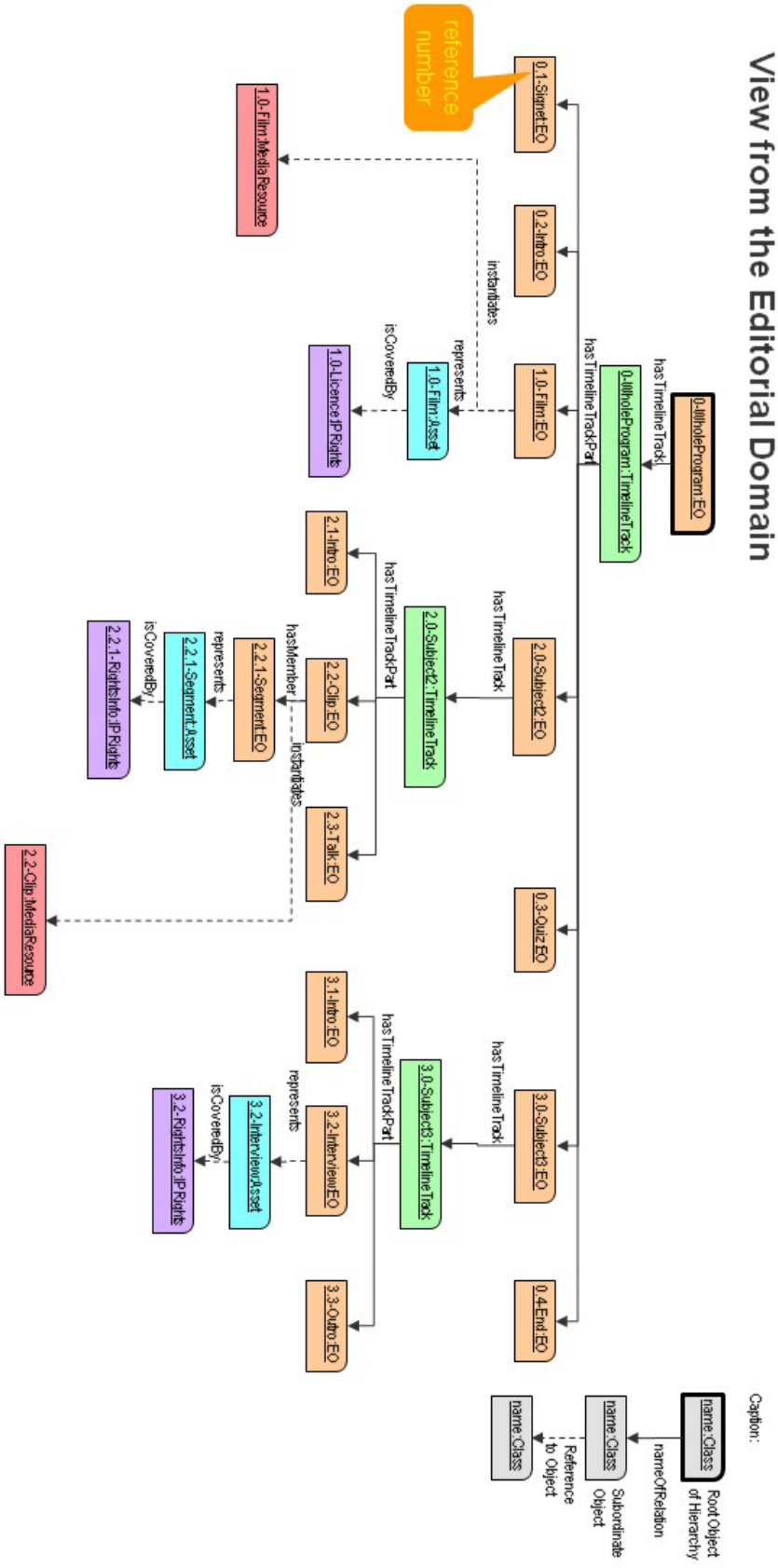
An example of a programme, called "ideal programme", is shown below:

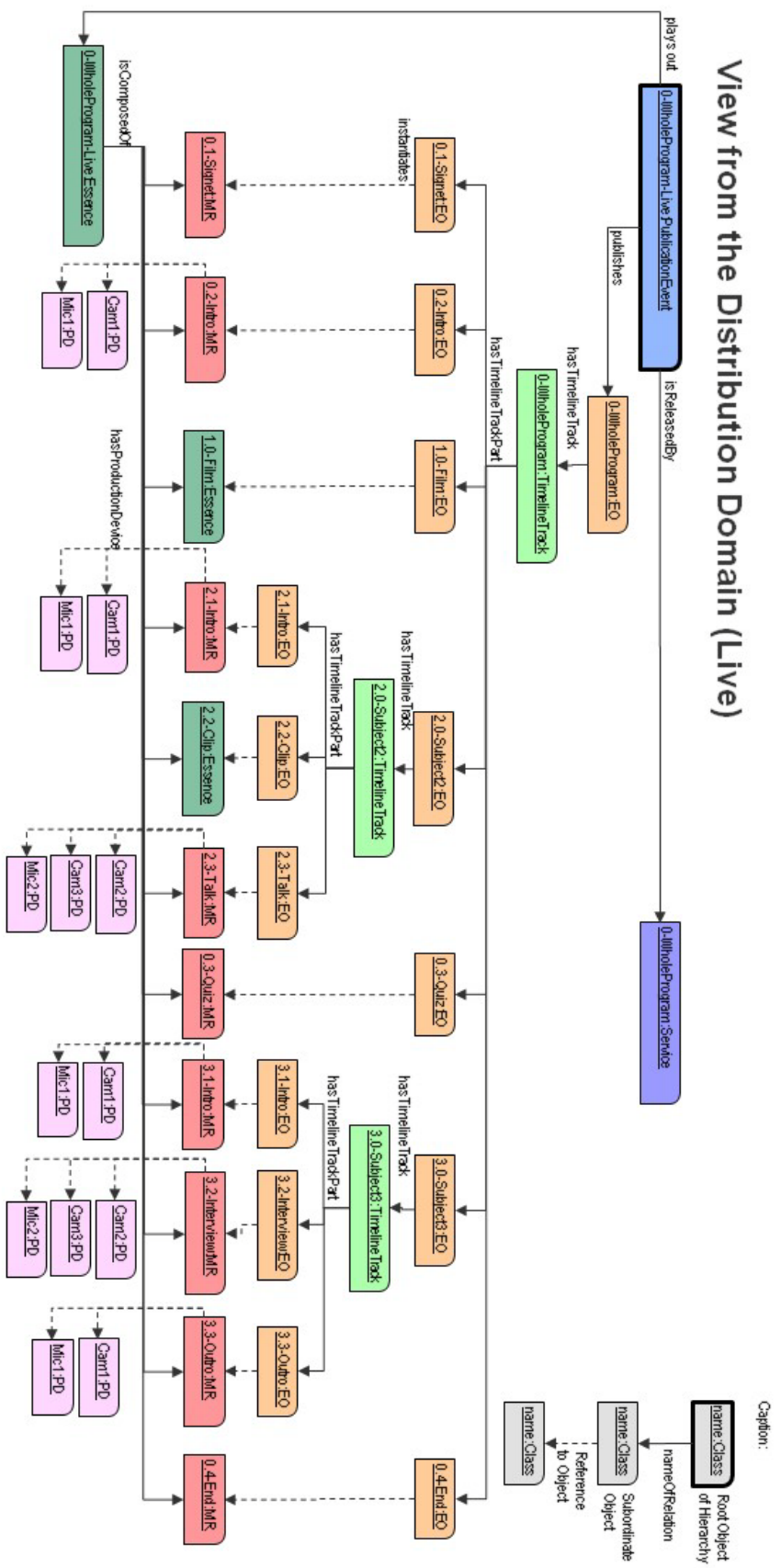
This example will now be represented using CCDM. The representation depends on the viewpoint, which maps nicely to the domains described in this document. Also, the following examples assume different Publication scenarios, such as "Live" or "Repetition". Some examples contain objects that are not directly represented in the graph of the "ideal programme", for example, the *ProductionDevices* Cam1 and Mic1.

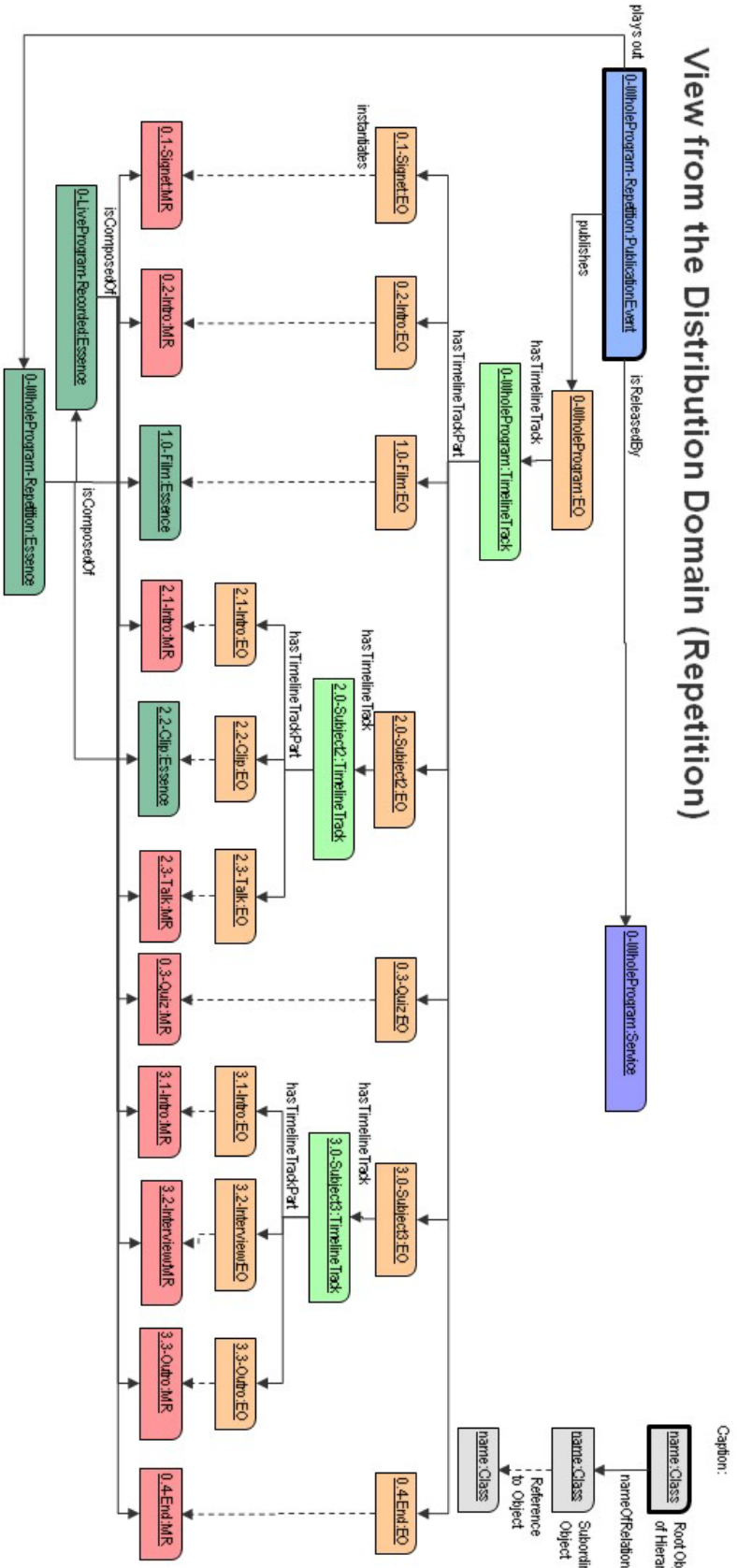
All of these assumptions were made only to show the possibilities of modelling with CCDM.

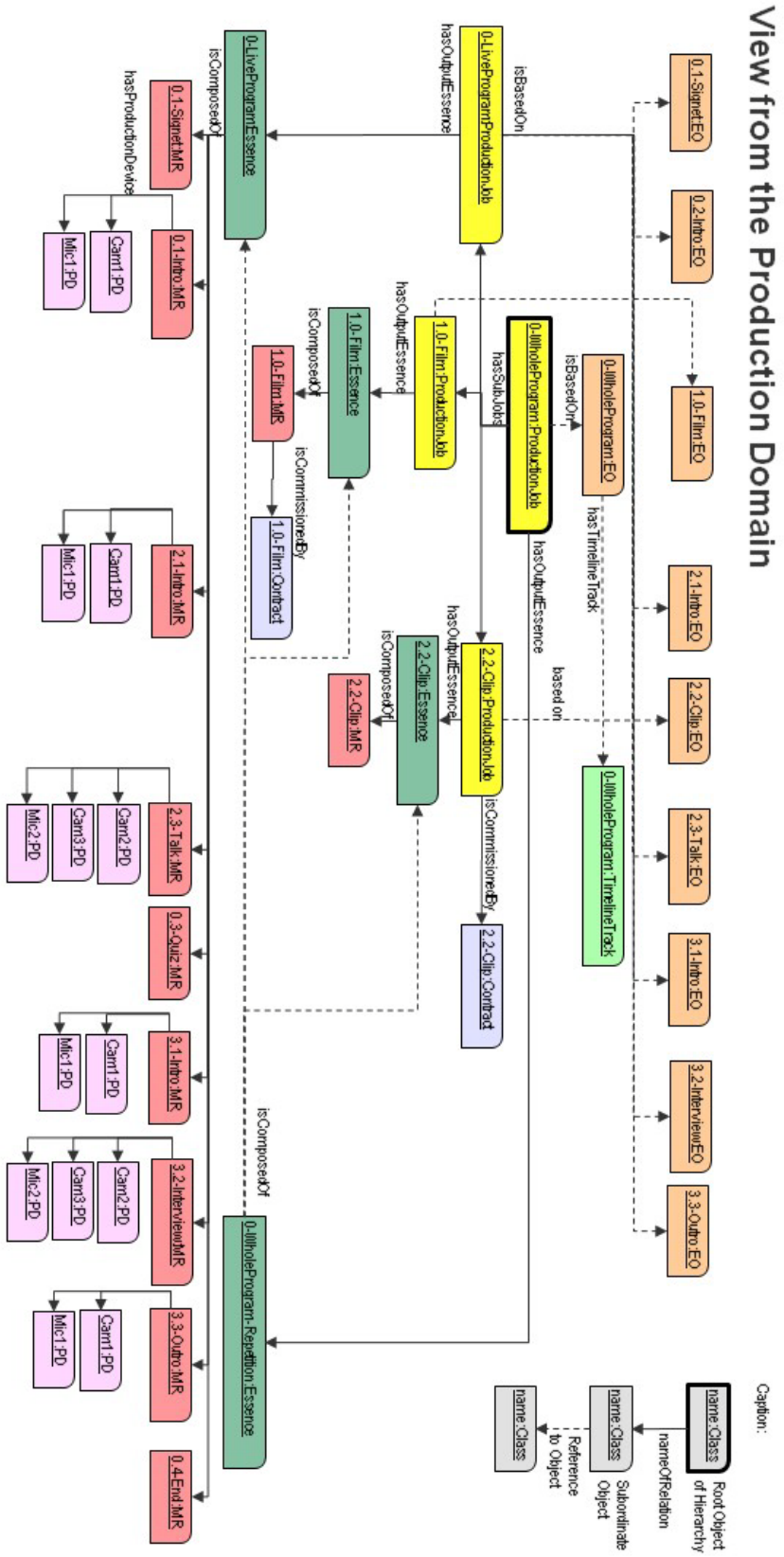
The object graphs represent a hierarchical structure, such as that found in an XML document. To emphasise the hierarchy it is necessary to introduce "references" (represented as dashed arrows) besides the pure object relation (represented as full arrows) in the hierarchy.

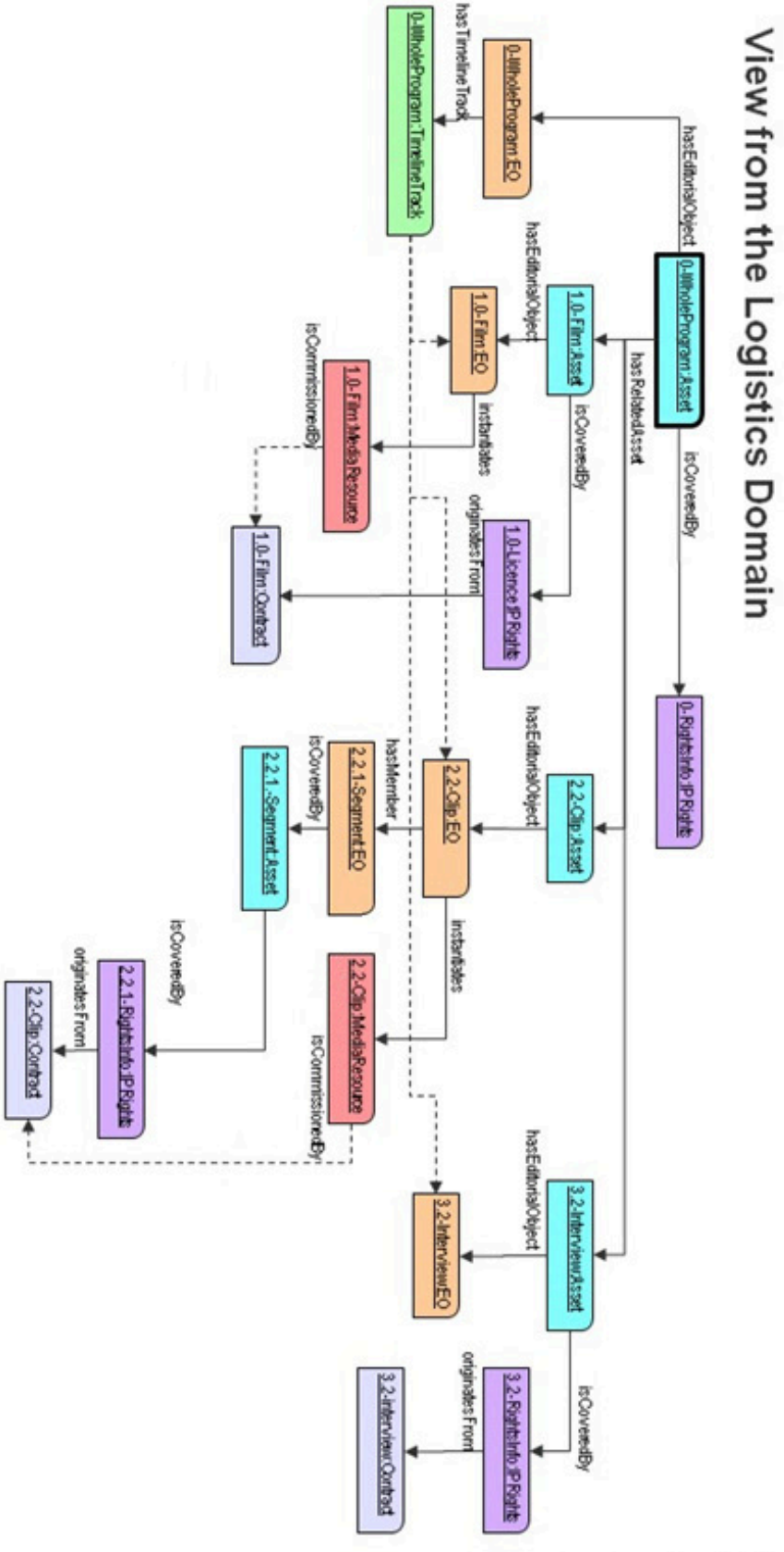
The following diagrams illustrate how to model the "ideal programme" with EBU CCDM 1.1.


View from the Editorial Domain
0.0WholeProgram:EQ
hasTimelineTrack
0.0WholeProgram:TimelineTrack
hasTimelineTrackPart
reference number
0.1-Signet:EQ
0.2-Intro:EQ
1.0-Film:EQ
instantiates
represents
1.0-Film:Asset
isCoveredBy
1.0-Licence:IPRights
1.0-Film:MediaResource
2.1-Intro:EQ
hasTimelineTrack
2.0-Subject2:EQ
2.0-Subject2:TimelineTrack
hasTimelineTrackPart
hasMember
2.2-Clip:EQ
2.2.1-Segment:EQ
represents
2.2.1-Segment:Asset
isCoveredBy
2.2.1-RightsInfo:IPRights
instantiates
2.3-Talk:EQ
2.2-Clip:MediaResource
0.3-Quiz:EQ
hasTimelineTrack
3.0-Subject3:EQ
3.0-Subject3:TimelineTrack
hasTimelineTrackPart
3.1-Intro:EQ
3.2-Interview:EQ
represents
3.2-Interview:Asset
isCoveredBy
3.2-RightsInfo:IPRights
3.3-Outro:EQ
0.4-End:EQ
Caption:
name:Class
Root Object of Hierarchy
nameOfRelation
name:Class
Subordinate Object
Reference to Object
name:Class

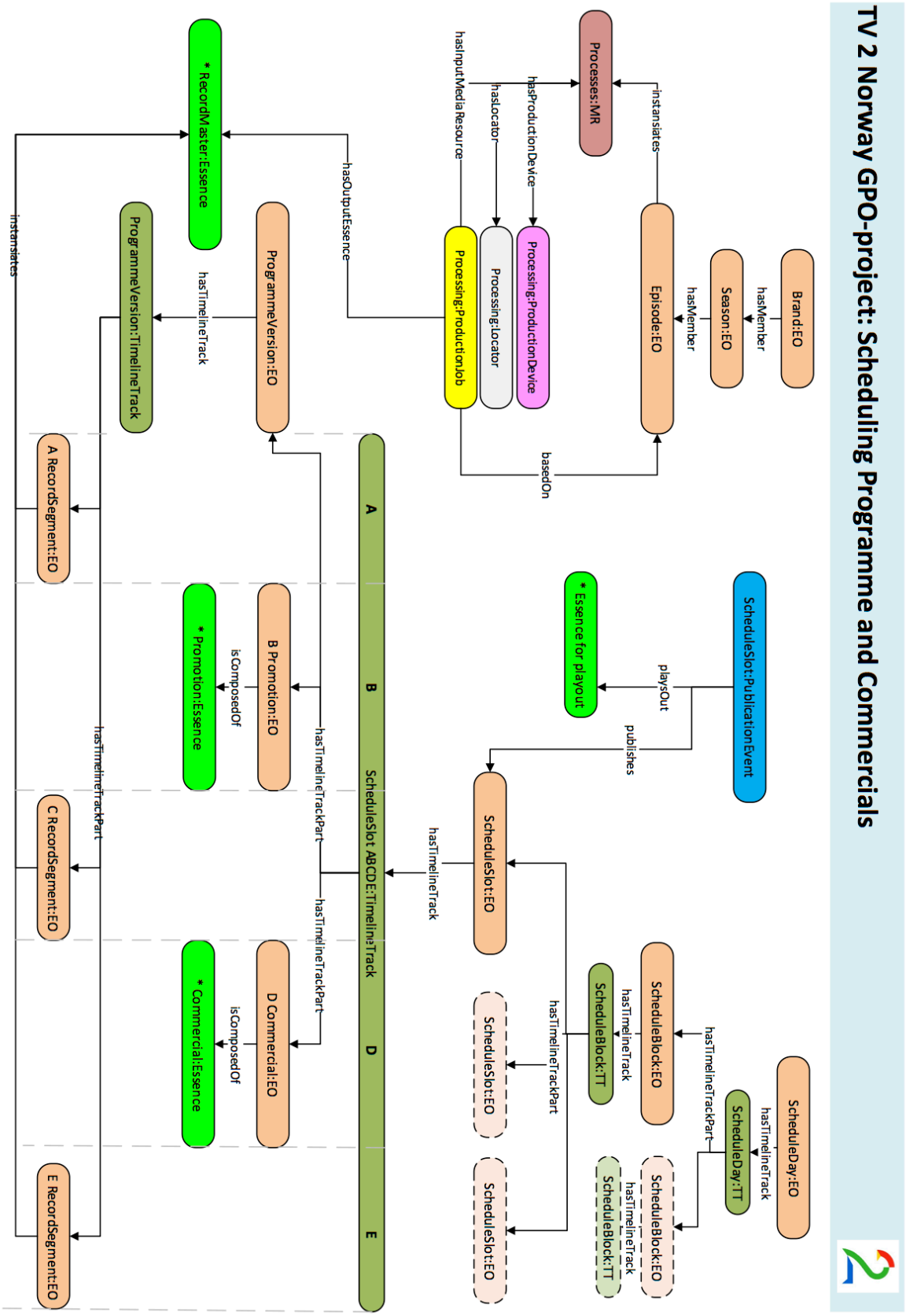

View from the Distribution Domain (Live)
plays out
0-WholeProgram-LivePublicationEvent
isReleasedBy
publishes
0-WholeProgram:Service
0-WholeProgram:EO
hasTimelineTrack
0-WholeProgram:TimelineTrack
hasTimelineTrackPart
instantiates
isComposedOf
0-WholeProgram-Live:Essence
0.1-Signet:MR
0.1-Signet:EO
0.2-Intro:MR
0.2-Intro:EO
Mic1:PD
Cam1:PD
1.0-Film:Essence
1.0-Film:EO
hasProductionDevice
2.1-Intro:MR
2.1-Intro:EO
2.2-Clip:Essence
2.2-Clip:EO
hasTimelineTrack
2.0-Subject2:TimelineTrack
2.0-Subject2:EO
hasTimelineTrackPart
Mic2:PD
Cam3:PD
Cam2:PD
2.3-Talk:MR
2.3-Talk:EO
0.3-Quiz:MR
0.3-Quiz:EO
3.1-Intro:MR
3.1-Intro:EO
3.0-Subject3:TimelineTrack
3.0-Subject3:EO
3.2-Interview:MR
3.2-Interview:EO
3.3-Outro:MR
3.3-Outro:EO
0.4-End:MR
0.4-End:EO
Caption:
name:Class
Root Object of Hierarchy
nameOfRelation
Subordinate Object
Reference to Object


View from the Distribution Domain (Repetition)
plays out
0-WholeProgram-Repetition:PublicationEvent
publishes
isReleasedBy
0-WholeProgram:EQ
0-WholeProgram:Service
hasTimelineTrack
0-WholeProgram:TimelineTrack
hasTimelineTrackPart
instantiates
0-1-Signet:EQ
0-2-Intro:EQ
1-0-Film:EQ
2-0-Subject2:EQ
2-1-Intro:EQ
2-2-Clip:EQ
2-3-Talk:EQ
2-0-Subject2:TimelineTrack
3-0-Subject3:EQ
3-1-Intro:EQ
3-2-Interview:EQ
3-3-Outro:EQ
3-0-Subject3:TimelineTrack
0-3-Quiz:EQ
0-4-End:EQ
0-1-Signet:MR
0-2-Intro:MR
1-0-Film:Essence
2-1-Intro:MR
2-2-Clip:Essence
2-3-Talk:MR
0-3-Quiz:MR
3-1-Intro:MR
3-2-Interview:MR
3-3-Outro:MR
0-4-End:MR
0-LiveProgram-RecordedEssence
isComposedOf
0-WholeProgram-Repetition:Essence
isComposedOf
Caption:
name:Class
Root Object of Hierarchy
nameOfRelation
name:Class
Subordinate Object
Reference to Object
name:Class

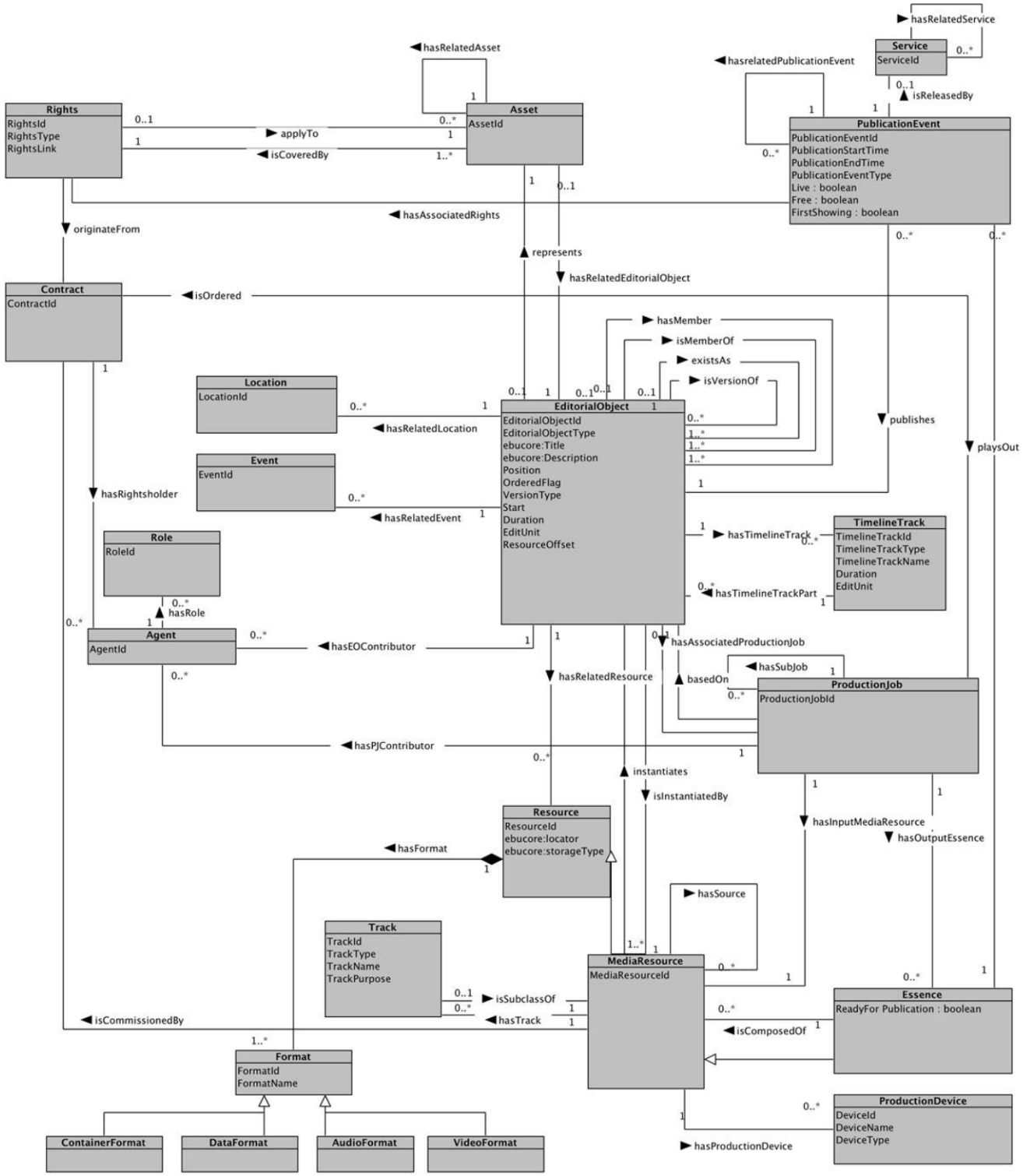

View from the Production Domain
0.1-Signet.EO
0.2-Intro.EO
1.0-Film.EO
2.1-Intro.EO
2.2-Clip.EO
2.3-Talk.EO
3.1-Intro.EO
3.2-Interview.EO
3.3-Outro.EO
0-WholeProgram.EO
0-WholeProgram.TimelineTrack
0-LiveProgramProduction.Job
1.0-Film.Production.Job
0-WholeProgram.Production.Job
2.2-Clip.Production.Job
0-LiveProgramEssence
1.0-Film.Essence
2.2-Clip.Essence
0-WholeProgram.Repetition.Essence
0.1-Signet.MR
0.1-Intro.MR
1.0-Film.MR
2.1-Intro.MR
2.2-Clip.MR
2.3-Talk.MR
0.3-Quiz.MR
3.1-Intro.MR
3.2-Interview.MR
3.3-Outro.MR
0.4-End.MR
Mic1:PD
Cam1:PD
Mic2:PD
Cam2:PD
Cam3:PD
1.0-Film.Contract
2.2-Clip.Contract
hasOutputEssence
isBasedOn
hasSubJobs
isComposedOf
isCommissionedBy
hasProductionDevice
hasTimelineTrack
based on
Caption:
name:Class
Root Object of Hierarchy
nameOfRelation
Subordinate Object
Reference to Object


View from the Logistics Domain
0-WholeProgram Asset
0-WholeProgram EQ
0-WholeProgram TimelineTrack
0..RightsInfo IPRights
1.0-Film Asset
1.0-Film EQ
1.0-Film Media Resource
1.0-Film Contract
1.0-Licence IPRights
2.2-Clip Asset
2.2-Clip EQ
2.2-Clip Media Resource
2.2-Clip Contract
2.2.1-Segment Asset
2.2.1-Segment EQ
2.2.1-RightsInfo IPRights
3.2-InterviewAsset
3.2-InterviewEQ
3.2-RightsInfo IPRights
3.2-InterviewContract
hasEditorialObject
hasTimelineTrack
hasRelatedAsset
isCoveredBy
instantiates
originates From
isCommissionedBy
hasMember
Caption:
name:Class
Root Object of Hierarchy
nameOfRelation
Subordinate Object
Reference to Object

3.3 Example provided by TV2, Norway



3.4 The total class diagram



3.5 More questions?

If you have questions on how to use or implement the EBU CCDM, please forward your queries to metadata@ebu.ch . You will receive personalised advice, and answers will enrich this section of a future version the specification, with your permission.

4. CCDM Compliance

The CCDM is an open framework allowing each user to adapt it to his own needs. As such, the EBU CCDM is flexible and adaptable in nature.

The CCDM ontology is provided as reference software implementation in RDF/OWL. It is available from the "Download Zone". This file contains the minimum set of classes, hierarchies of classes, *objectProperties* and *dataProperties* that compliant implementations should contain, extend, but not replace. More information of the CCDM ontology is provided in **Annex A**.

5. Download Zone

Filename and location	Description
https://www.ebu.ch/metadata/ontologies/ebuccdm https://www.ebu.ch/metadata/ontologies/ebuccdm/ https://www.ebu.ch/metadata/ontologies/ebuccdm#	RDF documentation
https://www.ebu.ch/metadata/ontologies/ebuccdm/ebuccdm.rdf	RDF / XML file

6. Licensing regime

The EBU CCDM is governed by Creative Commons' Attribution-NonCommercial-ShareAlike3.0 Unported (CC BY-NC-SA 3.0)

You are free: to *Share*—to copy, distribute and transmit the work, to *Remix* — to adapt the work, including under your own namespace under the following conditions:

	Attribution - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
	Noncommercial - You may not use this work for commercial purposes. *Note: this may be used in commercial products but cannot be sold as a specific feature*.
	Share Alike - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

7. Maintenance

The EBU CCDM specification is maintained by the EBU and suggestions for corrections or additions can be made by mailing to (metadata@ebu.ch).

8. Useful links

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 , 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/)

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

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

Page intentionally left blank. This document is paginated for two sided printing

Annex A: EBU CCDM ontology

The reference software implementation of the CCDM is provided in RDF/OWL.

A link for download is provided in § 5, "Download Zone", of this specification.

There is a variety of options for parsing and editing RDF/OWL documents and ontologies:

- Files with an 'owl' extension can be opened with text processors such as Wordpad;
- Microsoft Notepad can be used;
- More specialised software can be used:
  - Protégé (http://protege.stanford.edu/download/download.html) (recommended for beginners)
  - NeonToolkit (http://neon-toolkit.org/wiki/Main_Page)
  - TopBraid Composer, free edition (http://www.topquadrant.com/products/TB_Composer.html)