WP 2 CAM Metadata Framework
D2.1 CAM Core Metamodel definition

Action	Name	Date	Approbation
Contributors	Damien Alliez, Samir Amir, Ioan Marius Bilasco, Marija Bjeković, Patrick Blandin, Chabane Djeraba, Muriel Foulonneau, Juhani Laitakari, Jean Martinet, Eduardo Martínez Graciá, Daniel Pakkala, Matti Raty, Mika Rautiainen, Diego Esteban Rodriguez, Slim Turki, Jiehan Zhou	31/10/2008	
Approved by			
Approved by			

History

Date	Version & Status	Author	Modifications
14/12/2007	Version 0.10	J. Laitakari	Initial proposal
08/02/2008	Version 0.11	P. Blandin	Format and submission
05/06/2008	Version 0.12	S. Turki	New template
16/07/2008	Version 0.13	M. Bjeković	Progress on metamodel structure (CAM Elements aspect)
22/07/2008	Version 0.14	M.Bjeković	Evolution of metamodel structure (CAM Elements aspect) based on feedback from Eduardo Martinez Garciá
23/07/2008	Version v0.15	M.Bjeković	Introduction of modified Table 2, Table 3, initally part of D1.3
01/08/2008	Version 0.16	S. Turki, M. Bjeković	Update of content. Novel organisation of model behaviour section.
16/08/2008	Version 0.17	S. Turki, M. Bjeković	Update of content.
19|08|2008	Version 0.18	S.Turki, M.Bjeković, M.Foulonneau	Document structure update Progress on CAM Core structural and behavior aspect Introduction and modelling approach
27|08|2008	Version 0.19	J. Laitakari, S. Amir, S. Turki, M. Foulonneau	Section Overview of the metamodel Modification of the section on CAM Elements, addition of operations and relations
02|09|2008	Version 1.0	J. Laitakari, S. Amir, S. Turki, M. Foulonneau	Modification of operations Modification of model representations Minor edits
16/10/2008	Version 1.1	M. Foulonneau, J. Laitakari, M. Bjeković, M.Rautiainen, M.Bilasco, S.Turki, S. Amir, J.Martinet, E. Martínez Graciá, M. Raty,	Reorganisation of section 4.2 – new subsections Model evolution following the discussions and PCC meeting in Luxembourg (versioning included)
17/10/2008	Version 1.3	M. Foulonneau, M. Bjeković, M.Bilasco, S.Turki, S. Amir,	Feedback from CNRS on version 1.2

		J.Martinet	
22/10/2008	Version 1.4	M.Bjeković, S.Turki, M.Foulonneau	Organization of behavior and integrity rules section Section references and figures
23/10/2008	Version 1.5	M. Foulonneau, M. Bjeković, M.Bilasco, S.Turki, S. Amir, J.Martinet	Integrity rules with minor changes
24/10/2008	Version 1.6	M.Rautiainen, M. Foulonneau, M. Bjeković, M.Bilasco, S.Turki, S. Amir, J.Martinet	First inclusion of operations with comments from Tudor
31/10/2008	Version 1.7	M.Rautiainen, J.Laitakari, S.Amir, M.Bilasco, S.Turki, M.Foulonneau, M.Bjeković D.Alliez, E. Martínez Graciá	Modifications following the TELCO of 28/10/2008 Executive summary Final inclusion of modified operations in the document
31/10/208	Version 2.0	M.Rautiainen, J.Laitakari, S.Amir, M.Bilasco, S.Turki, M.Foulonneau, M.Bjeković	Final version

Table of contents

Executive Summary

CAM4Home project implements the concept of Collaborative Aggregated Multimedia. CAM4Home Metadata Framework will enable a novel way of content provisioning by bundling different types of content and content services into content bundles.

CAM Metamodel provides the core concepts and required metadata level information for collaborative distribution of multimedia and software content as a structured model, which can be partially or fully instantiated as metadata and used as the basis for services development. CAM Metamodel is composed of: 1) CAM Core Metamodel, which represents the core CAM entities (Bundles and Objects); 2) CAM Supplementary Metamodel, which represents application-dependent entities such as devices and users, and 3) CAM External Metamodel which acts as an interface towards external metadata formats (i.e. SMIL, MPEG7).

This document defines CAM Core Metamodel. It a complete description of the structure and behavior of core CAM entities that are necessary to represent and manipulate Collaborative Aggregated Multimedia.

CAM Core Metamodel supports the representation of a wide variety of Multimedia content in CAM Objects: downloadable applications, software services, images, video, etc. Specific metadata is attached to different types of Multimedia entities. This metadata describes both the content file or service deployment method and the actual content or service that is provided.

This metamodel also describes the mechanisms by which CAM Bundles aggregate CAM Objects (CAM Elements and their CAM Element Metadata). CAM Bundles are aggregations of two or more CAM Objects and a description of that aggregation. They include CAM Bundle Metadata and the relationships between CAM Objects inside a CAM Bundle.

CAM Element Metadata and CAM Bundle Metadata can be versioned. Versions allow the evolution of the descriptions during the lifecycle of CAM Bundle Metadata and CAM Element Metadata. CAM Core Metamodel provides the support for versioning, as well as conditions under which new versions of metadata descriptions are created.

Glossary

1.1.Definitions

Term	Description
CAM Element	Atomic unit of aggregation in CAM Bundles. May be content or service. (e.g. digital picture, video clip, MMS service, …)
CAM Element Metadata	Existing metadata of a CAM Element. (e.g. Metadata of a video clip, digital picture, MMS service, …)
CAM Object	CAM Element Metadata having reference to the essence file (CAM Element's EssenceFileIdentifier) defined.
CAM Bundle	A file or stream that represents aggregation of two or more CAM Elements, their CAM Element Metadata, and a description of the aggregation using CAM Metadata.
CAM Bundle Metadata	Descriptive information about aggregation in a CAM Bundle using the attributes and relations defined by the CAM4Home metadata framework.

1.2.Acronyms and abbreviations

Abbreviation	Description
N/A	Not Applicable
TBC	To Be Completed
CAM	Collaborative Aggregated Multimedia
Class	Class from CAM Core Metamodel
Attribute	Attribute from CAM Core Metamodel
association	Association from CAM Core Metamodel

2. Introduction

The CAM4Home project aims to specify and demonstrate a novel CAM4Home metadata framework that is able to encapsulate existing metadata technologies for multiple types of content and also able to incorporate references to content related services. This CAM4Home metadata framework will enable a novel way of content provisioning by bundling different types of content and content services into content bundles on the level of metadata.

The purpose of this document is to specify CAM Core Metamodel. The modelling is relying on concepts defined in CAM Concept description document (D.Pakkala from 12.6.2008), and on D1.3 Metadata Framework Requirements v1.6.

CAM Core Metamodel covers the structure and the behavior of core CAM entities (CAM Object, CAM Bundle), their metadata description and relationships. The principal objective of CAM Core Metamodel is to provide the common ground of CAM4HOME Metadata Framework.

This document is organized as follows: Section 3 gives an overview of CAM Metamodel and resumes the responsibilites of CAM Metamodel sections. Section 4 introduces CAM Core Metamodel: Firstly, the modelling domain background is introduced, and afterwards CAM Core Metamodel is discussed through its structure, integrity rules and operations sub-sections.

3. CAM Metamodel overview

CAM Metamodel is an explicit model of the constructs and rules needed to build the metadata for describing the content and its aggregation in CAM Bundles. CAM Metamodel provides the core concepts and required metadata level information for collaborative distribution of multimedia and software content as a structured model which can be partially or fully instantiated as metadata and used in the system. In addition, CAM Metamodel is designed to allow easy encapsulation of existing metadata formats into the structures of the instantiated metadata. External metadata formats can be for example MPEG-7 formatted description of the content or WSDL description of a service's interface that are nested in the metadata. Furthermore, the most important design goal of the CAM Metamodel is to create easily extendible model which allows definition of new structures and associations that a system might need in its operation, without requiring any alteration or redefinition of existing structures and associations. As the CAM Metamodel can be extended to meet the requirements of a content distribution system without breaking the core structures, it maintains its compatibility with other systems. Extended CAM Metamodel provides added-value for the systems that understand the extensive structures, but still can be used for collaborative content distribution by systems conforming to original CAM Metamodel. CAM Metamodel design is illustrated in Figure 1.

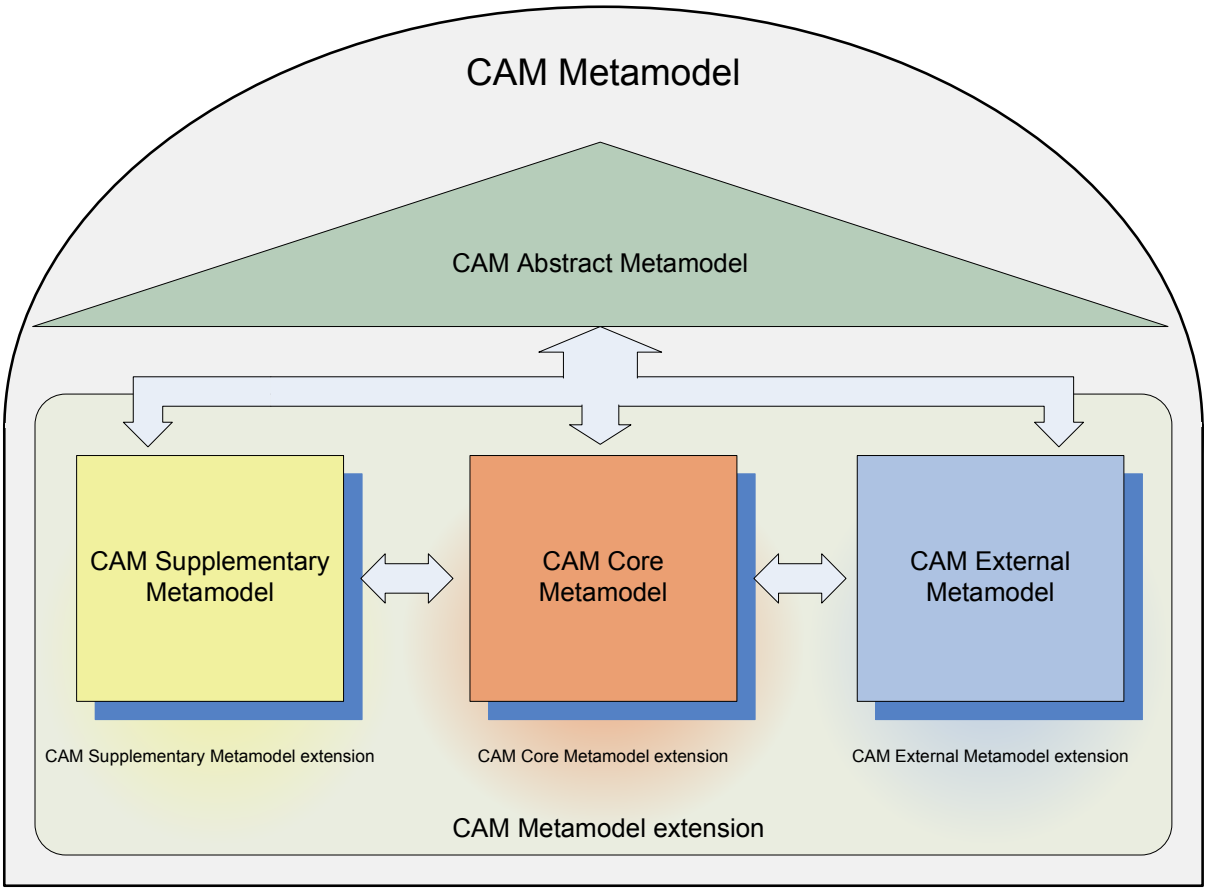

Figure 1. CAM Metamodel design.

The design of the CAM Metamodel is based on a higher level description (CAM Abstract Metamodel) of the metamodel which defines the basic constructs and associations between them on an absract level. Abstract level representation of the CAM Metamodel acts as a connecting element between different categories of the CAM Metamodel and enables the extension of the model with new structures. As new structures extending the metamodel conforms the constraints and classifications presented in the abstract model the extended CAM Metamodel remains compatible with the rest of the CAM4Home system, and also provides added-value for the applications in form of additional information about the content or its aggregation in the CAM Bundle.

The lower level design of CAM Metamodel is divided into three different categories, each of them providing conceptual distinction for the metadata that the metamodel category describes. These metamodel categories conform to the abstract level of the metamodel and extend it providing more detailed metadata constructs and rules. Categories are CAM Core Metamodel, CAM Supplementary Metamodel and CAM External Metamodel and they are defined as follows:

- **CAM Core Metamodel:** Defines the core structures and associations that are related to the distributed content and its basic metadata. This metamodel defines the classification of different types of CAM Elements, related CAM Element Metadata and constrained associations between them. CAM Core Metamodel also specifies the structure of a CAM Bundle and defines the metadata structures that describe the aggregation between two or more CAM Objects inside the bundle.
- **CAM Supplementary Metamodel**: Defines the metamodel for metadata that is required to enable interoperability of the platform services and supplement the Core metadata. CAM Supplementary

Metamodel provides the structures of eg. user profiling, community, administrative and system specific metadata into the CAM Metamodel.

- **CAM External Metamodel**: Defines a structure into CAM Metamodel which acts as an interface towards the external metadata formats (e.g. MPEG-7, TVAnytime, etc.) and encapsulates them into the CAM Metamodel.

All these three categories can be extended when a need for new type of metadata arises. For example, if new types of content are wanted to be described on the metadata level, CAM Core Metamodel is extended respectively. Similar ability for extension applies also to CAM Supplementary and CAM External Metamodel. When user profile metadata is wanted to cover more detailed information the CAM Supplementary Metamodel is extended and if new existing metadata format for a distributed content is represented, appropriate extension to CAM External Metadata is defined.

The **CAM Abstract Metamodel** purpose is to comprehensively define the required higher level constructs that allow the extension of CAM Core Metamodel, CAM Supplementary Metamodel and CAM External Metamodel. On this account the CAM Abstract Metamodel shouldn't require any modifications when the metamodel is widened, and the extensive structures would be targeted on the lower-level metamodel categories. This is illustrated in Figure 1.

4. CAM Core Metamodel

4.1. Background

CAM Core Metamodel relies on the base concepts of CAM4HOME domain, defined in CAM Concept document [1]. On the other side, Metadata Framework Requirements document [2] served as an input for detailed specification of metadata related to those base concepts.

CAM Concept document defines the following concepts that are crucial for CAM Core Metamodel:

- CAM Element: Atomic unit of aggregation in CAM
- CAM Element Metadata: Descriptive or technical metadata related to CAM Element
- CAM Object: The representation of CAM Element and CAM Element metadata.
- CAM Bundle:The aggregation of two or more CAM Objects and the description of that aggregation (contained in CAM Metadata).
- CAM Bundle Metadata: Descriptive information about aggregation of CAM Objects in a CAM Bundle.

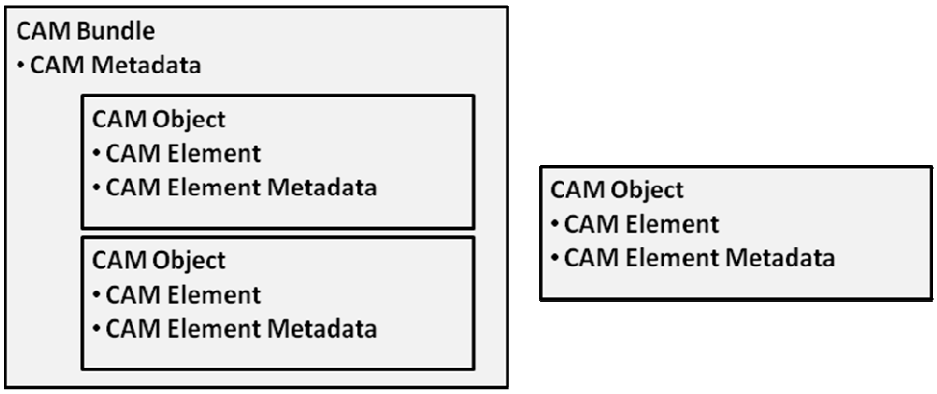

Figure 2: Conceptual view of CAM Object and CAM Bundle

4.1.Modeling approach

The proposed approach to build the CAM Core Metamodel is inspired by information systems engineering approach [3, 4]. The approach is basically object oriented. Its main characteristic is that it elaborates system specification by gathering the structural (static aspects), behavioral (dynamic aspects) and integrity rules (constraints) into the same specification.

Accordingly, CAM Core Metamodel work is organized in three main interelated perspectives: (i) structure, (ii) behavior and (iii) rules, every perspective tackling basic CAM concepts.

UML (Unified Modelling Langage) has been adopted as a notation to specify some aspects of the CAM Core Metamodel:

1. **CAM Core Structure**

   The CAM Core structure organizes the concepts related to CAM elements and CAM bundles, as well as their attributes and relationships.

   The CAM Core Metamodel structure is presented through a set of documented UML class diagrams.

2. **CAM Core Behavior**

   The CAM Core behavior describes the allowed treatments on metamodel core concepts defined in the CAM Core structure.

   The CAM Core behavior is specified through (i) a life cycle view and (ii) a set of CAM Core operations. The life cycle view is presented through UML state-chart diagrams.

3. **CAM Core Rules**

   The CAM Core rules are integrity rules, specified to preserve the coherence and consistency of core concepts and their relationships through their entire life cycle.

   An integrity rule is a condition that must be validated by the information stored and manipulated in the system. When all integrity rules are respected, the system is said to be coherent.

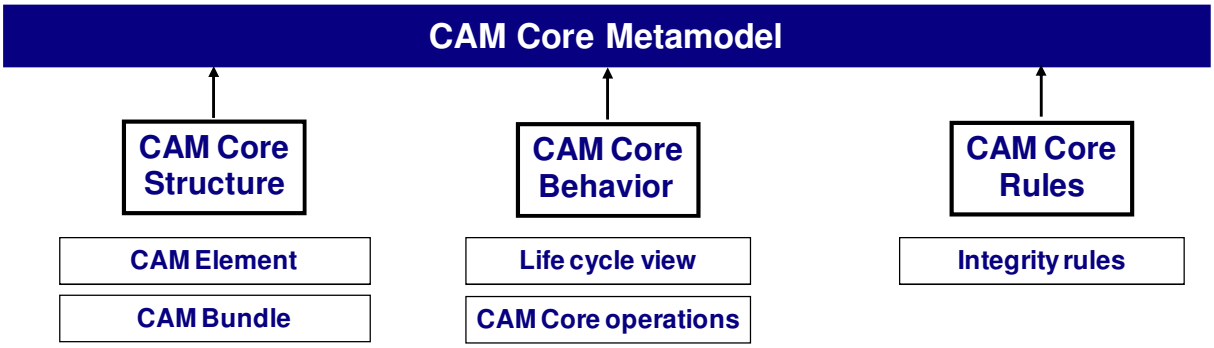

Figure 3: CAM Core Metamodel

4.2.CAM Core Structure

The CAM Core Metamodel structure organizes the entities related to CAM Elements and CAM Bundles, and their relationships.

The CAM Core Metamodel sturcture provides one possible representation (at this level of abstraction) of CAM Element Metadata and CAM Bundle Metadata. The metadata description of CAM Element (CAM Element Metadata) provides description on either the multimedia content or the techincal information.

The metadata description of Bundle (CAM Bundle Metadata) provides the information on aggregation of CAM Objects in a CAM Bundle.

CAM Object concept is not explicitly represented in CAM Core Metamodel. It corresponds in the model to CAM Element Metadata with completely specified CAM Element (addressed as valid CAM Element Metadata). Moreover, CAM Element Metadata and CAM Element follow the same life-cycle, being treated as “one”, as inseparable from each other. CAM Element is nothing more than a container of the refrence to essence file, whose content is described through corresponding metadata. This approach is compliant with CAM Object definition in [1].

Boh CAM Element Metadata description and CAM Bundle Metadata description can be versioned, in order to keep corresponding description’s evolution history.

The section 4.2 is organized as follows: 4.2.1 CAM Objectintroduces the specification of CAM Object. Basic definitions are presented in 4.2.1.1, and taxonomy of content in 4.2.1.2. CAM Element metadata specification is presented in 4.2.1.3. The specification regarding CAM Bundle is presented in 4.2.2, intorducing definitions in 4.2.2.1, and CAM Bundle Metadata specification in 4.2.2.2.

4.2.1. CAM Object

4.2.1.1. Definitions

CAM Element

CAM Element is considered as an atomic aggregation unit within CAM Metadata framework, representing either multimedia content or service providing content.

CAMElement class only provides the reference to the physical file bearing multimedia content or service. The physical file that bears the content is addressed as *essence file*, while its identifier (reference) is addressed as *EssenceFileIdentifier*.

EssenceFileIdentifier identifies the actual content that is stored outside or inside the system. Ensuring that the content exists is outside the scope of CAM Metamodel.

In CAM Core Meta-model, **CAMElement** class is also root class for specifing taxonomy of the content supported by CAM4Home Metadata Framework (see section 4.2.1.2).

CAM Element Metadata

Essence file's content is described through CAM Element Metadata. The *EssenceFileIdentifier*, encapsulated in **CAMElement**, is practically considered as some sort of metadata, as well. **CAMElement** instance follows the same life-cycle as **CAMElementMetadata** instance (version) it is attached to (reflected as defined omposition association *isMetadataOf* as in Figure 4).CAM Element Metadata is semantically dependent on the content of the essence file abstracted in **CAMElement**. However, registering of CAM Element Metadata (and **CAMElement**) in the system is possible prior to the existence of (or knowledge of the reference to) the essence.

The same essence (uniquely identified by *EssenceFileIdentifier*) can be described differently by different users, thus the same content can have several CAM Element Metadata descriptions.

In addition, one CAM Element Metadata description can be versioned. Versions keep trace of the description's evolution. Rules are defined to determine which modifications in the description trigger the creation of its new version.

One instance of **CAMElementMetadata** (with its associations) corresponds to the concrete version of CAM Element Metadata description. The description being versioned is identified through *CAMElementMetadataID*, and versions inside the description are identified by *VersionNumber*.

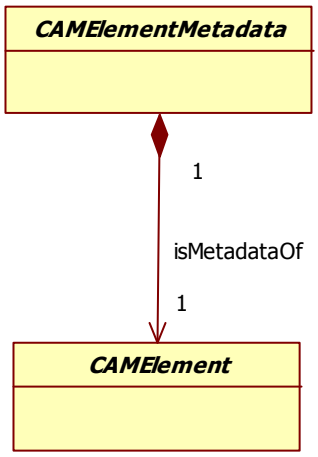

Figure 4: CAMElement and CAMElementMetadata

CAM Object

CAM Object concept corresponds to CAM Element Metadata with completely specified **CAMElement** (having defined *EssenceFileIdentifier*), that is, to valid **CAMElementMetadata.** CAM Object is not explicitly represented in CAM Core Metamodel.

CAM Object is considered as a sort of "wrapper" for accessing CAM Element Metadata and CAM Element at once. In order to shorten addressing CAM Element and CAM Element Metadata, when dealing with operations in our CAM Core Metamodel, the term CAM Object (= valid CAM Element Metadata) is used.

4.2.1.2. Taxonomy

In CAM Core Metamodel, both **CAMElementMetadata** and **CAMElement** are following the same taxonomy. This taxonomy organizes the types of content supported in CAM4HOME domain.

Supported content falls into the following two categories (Figure 5):

1. **Multimedia** category – comprising
    - **Image** - references a downloadable image file.
    - **Document** - references a document file containing formatted text and pictures (e.g. Microsoft Word document, Adobe PDF, etc.).
    - **Audio** - references either a downloadable audio file of an arbitratry format, or a live/recorded audio stream available through streaming service. Additional attributes of **Audio** class (*AudioType*, *StreamingType*) enable the distinction between types of audio content:
        - *AudioType* enables the distinction between f*ile or streaming type.*

- *StreamingType* in case of streaming, enables the distinction whether it is live or recorded audio streaming.

- **Video** - references either a downloadable video file of an arbitratry format, or a live/recorded video stream available through streaming service. Additional attributes of **Video** class (*VideoType*, *StreamingType*) enable the distinction between types of video content:

  - *VideoType* enables the distinction between file or streaming type.

  - *StreamingType* in case of streaming, enables the distinction whether it is live or recorded audio streaming.

2. **Service** category – comprising

   - **UserService** - references a software application producing an added value to user(s) and potentially utilizing software services in its operation. It can an be a reference to a remotely executable file containing an application that is operable in client devices (e.g. Flash application that cannot be stored for local use).

   - **SoftwareService** - references an instance of functionality of software executed in one or more host(s) that can be registered, discovered and invoked by other software executed in the same or another host. Software service cannot be utilized directly by a user but needs a User Service or an application to do that.

   - **DownloadableApplication** – references a downloadable executable file containing an application that is operable in client devices (e.g. a Java jar-file for mobile devices).

The essential difference between these two categories is in the nature of information that the respective metadata provide:

- Metadata description for audio, video, image document (**Multimedia**) provides the information related to the very content (i.e author of a film, film genre, ….) of a CAM Element.

- Metadata description for user service, software service and application (**Service***)* basically provides the technical information, (i.e. execution requirements) and very few (or none) information on the very content of a CAM Element.

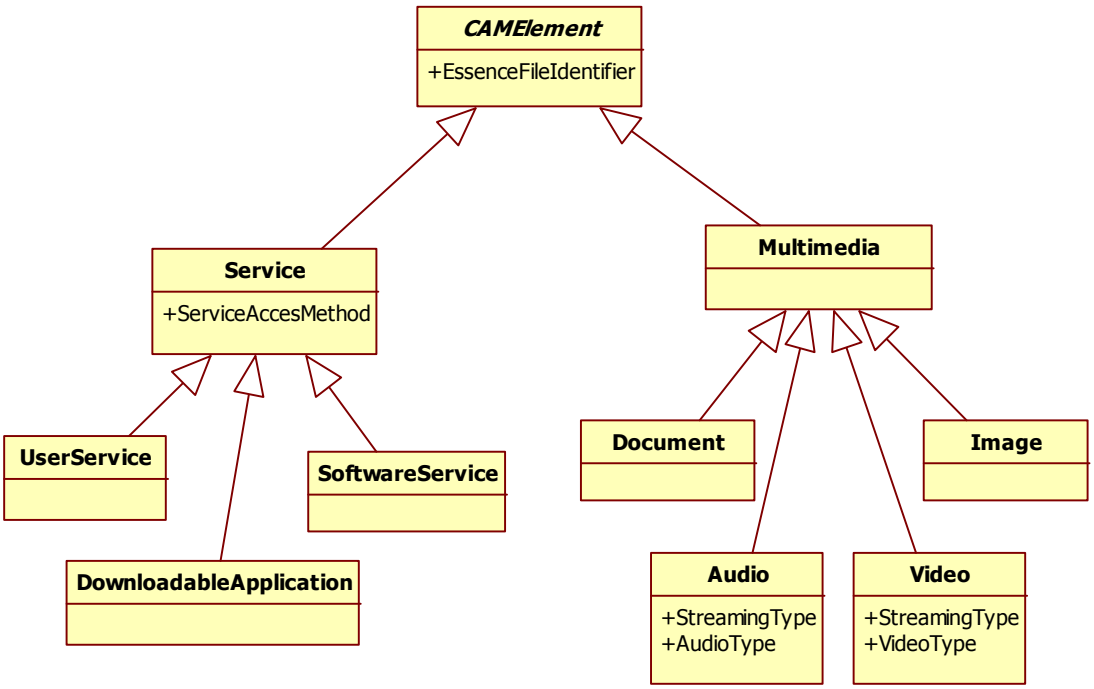

Figure 5: Content taxonomy (CAMElement)

Parallel taxonomies (Figure 6) allow flexible extension of CAM Core Metamodel to support additional type of content.

The constraints (see rules in section 4.3.2) on *isMetadataOf* association between CAMElement and CAMElementMetadata further define which CAMElementMetadata specialization associates with which type of content (corresponding CAMElement specialization). For example, ImageElementMetadata could only be associated to Image CAMElement.

Since **CAMElement** and **CAM ElementMetadata** specializations are only to be instantiated, CAMElement and CAMElementMetadata as hierarchy roots are specified as abstract classes.

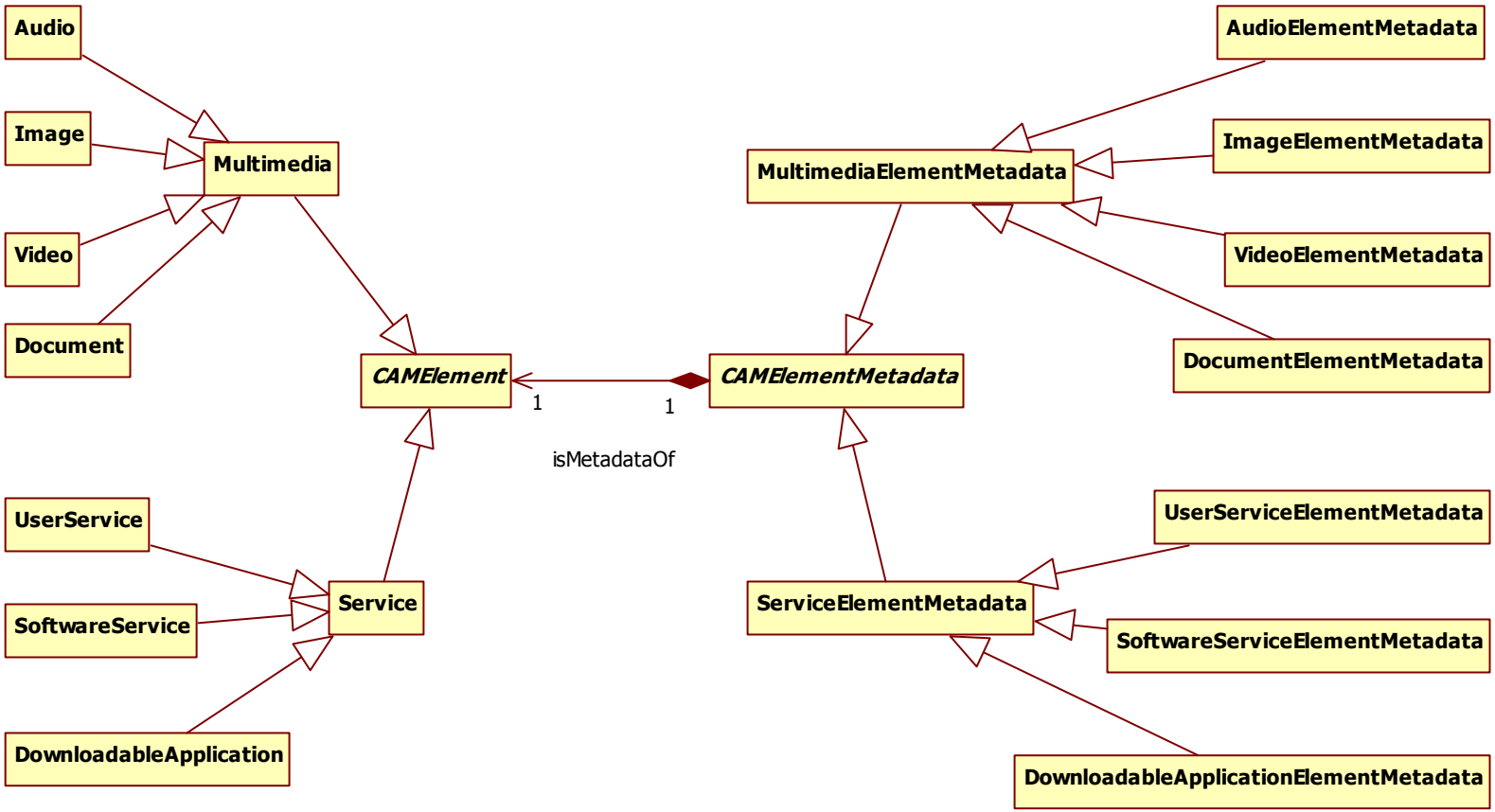

Figure 6: Parallel taxonomies of CAMElement and CAMElementMetadata

4.2.1.3. CAM Element Metadata specification

4.2.1.3.A. Introduction

Metadata gives detailed description of the content or service whose reference is encapsulated by **CAMElement**. Metadata is distributed in two "layers" in CAM Core Metamodel, according to the following logic:

1. Metadata that describe the content file or service deployment method (ie. web service on some server) should be placed under the classes that inherit **CAMElement** class. (Figure 7)
2. Metadata that describe the actual content or the service are specified under classes that inherit **CAMElementMetadata** class. Metadata description is presented in the meta-model either with a set of attributes (of **CAMElementMetadata** class) or as associations between **CAMElementMetadata** and various classes (i.e **ContentGenre**, **Author**, etc.).

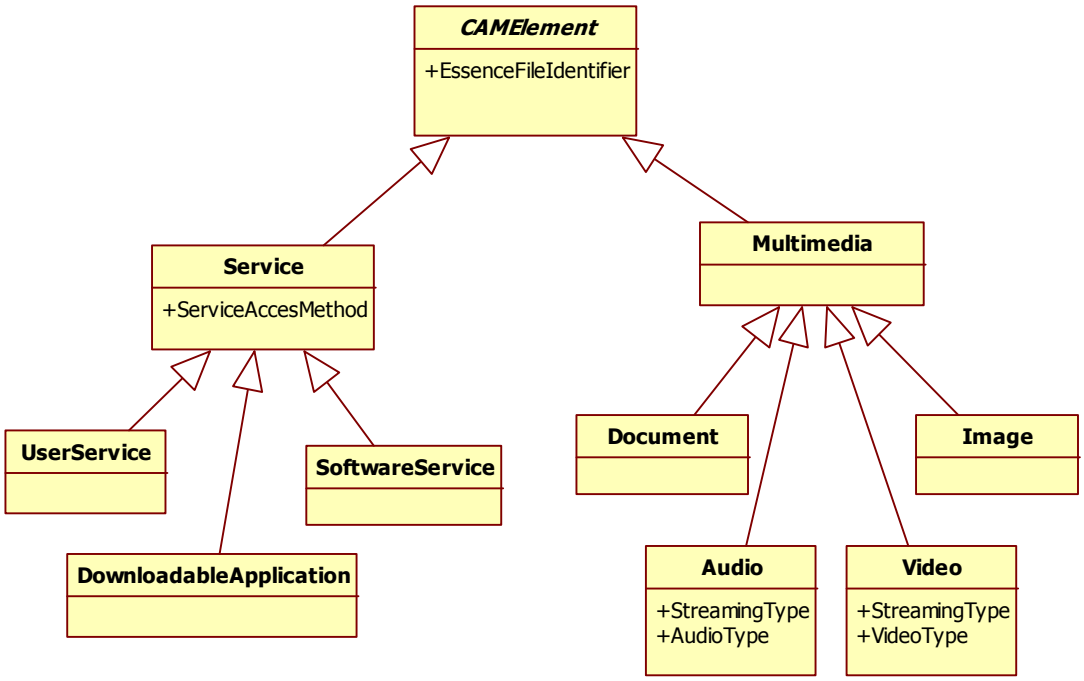

Figure 7: Metadata distribution - on CAM Element

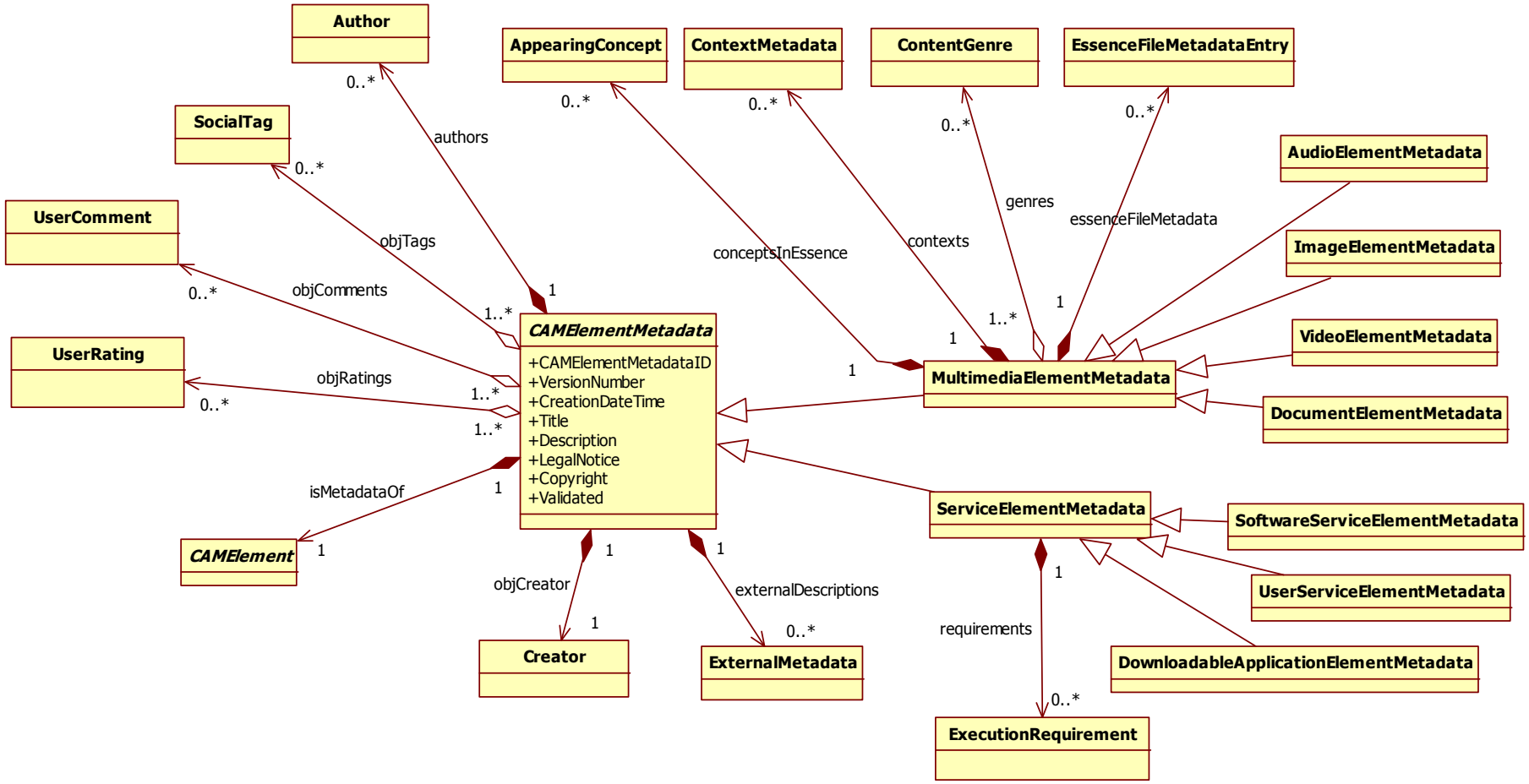

Figure 8: Metadata distribution on CAMElementMetadata - overview

4.2.1.3.B. Versioning

CAM Element Metadata description can have multiple versions. Versions allow the evolution of the descriptions during the lifecycle of CAM Element Metadata and CAM Bundle Metadata. Rules define which modifications in the description trigger the creation of a new version. Regarding their effect on versioning the metadata, following cases can be identified:

1) *CAMElementMetadataID* and metadata on the creator of CAM Element Metadata description (*objCreator*) cannot be modified inside the same *CAMElementMetadataID*.
2) *VersionNumber* and *CreationDateTime* cannot be modified inside the version.
3) As for metadata that can be modified, some metadata modifications does not require creating new versions. This metadata is defined as being *dynamic* (i.e. their values can change inside the same version). Following metadata is currently defined as dynamic:
   - *UserComments*
   - *UserRatings*
   - *SocialTags*
   - *ContentGenre*
4) In addition to the metadata described above, all other metadata is declared to be *static*, and their modification should trigger the creation of a new version of Object. Some examples of static metadata are:
   - *Author*, *Title*, *Description*, *Copyright*, *LegalNotice*, etc.

New version of CAM Element Metadata would be equal to duplication of the current version contents and modification of intended metadata.

Remark on the realization of dynamic metadata in the versioning context: Since *dynamic* metadata contain values that are updated constantly by the users of the system, a mechanism is required to allow accessibility to the latest version. The recommended mechanism is defined hereafter:

Dynamic metadata consists of three entities: *ReflectorURI* and *CachedMetadata* and *RetrievedOnDate*.

ReflectorURI contains a URI to a location where the latest contents of the metadata are maintained and can be retrieved to a local structure. One *ReflectorURI* is required for every dynamic metadata entity, e.g all *UserComments* should have exactly one *ReflectorURI*.

CachedMetadata contains metadata content that has been retrieved from the *ReflectorURI* on a date described in *RetrievedOnDate* entity. *CachedMetadata* is not guaranteed to be up-to-date, but it allows consumption of the metadata when the location in *ReflectorURI* is unreachable.

Each new version of CAM Element Metadata is defined as invalid until it is explicitly validated (*Validated* attribute). Validation is an activity that happens at the end of distribution and before the beginning of delivery phase (of Bundle life cycle). Invalid metadata is not allowed to be registered for delivery in the system.

In the context of CAM Element Metadata, validation activity should check whether a CAM Element Metadata version has *EssenceFileIdentifier* defined.

4.2.1.3.C. Metadata

Metadata gives detailed description of the content or service whose reference is encapsulated by **CAMElement**. Depending on the type of described content (multimedia or service), the metadata description can be different. Metadata description that is common for all CAM Element categories is presented in section ***Common metadata***, metadata specific to Multimedia in section ***Multimedia-related metadata***, while metadata specific to Service is treated in section ***Service-related metadata***.

Before proceeding to detailed description of metadata, we first give some general considerations regarding metadata presented.

Modeling choices regarding metadata within versions

Versions allow the evolution of the descriptions during the lifecycle of CAM Element Metadata. New versions are created by duplicating current version contents and modification of intended metadata.

Metadata being shared by different versions of metadata description under the same *CAMElementMetadataID* is said to have family scope. Entities presenting those metadata (**UserComment, UserRating, SocialTag, ContentGenre**) are associated with **CAMElementMetadata** class by means of aggregation (see Figure **8**, Figure 10, Figure 11). The cardinalities for mentioned associations (*objRatings, objComments, objTags,* genres) reflect this: association cardinalities on the side of **CAMElementMetadata** are 1..*.

Metadata having the lifespan of a version (*CAMElementMetadataID,VersionNumber*) is said to have version scope. Entities presenting those metadata (**CAMElement, ExternalMetadata, Author, AppearingConcept, ContextMetadata, EssenceFileMetadataEntry, ExecutionRequirement)** are associated with **CAMElementMetadata** class by means of composition (see Figure **8**, Figure 10, Figure 11). The cardinalities for mentionned associations (*isMetadataOf*, *externalDescriptions, authors, conceptsInEssence, contexts, essenceFileMetadata, requirements*) reflect this: association cardinalities on the side of **CAMElementMetadata** are 1.

Remark on the usage of references in metamodel

In CAM Core Metamodel, the following rules are applied for representing (and naming attributes containting) references:

- For attributes addressing metadata structures within CAM Metamodel, a suffix *Reference* is used:
    - Several metadata entities have attributes addressing users of the system (**UserRating, UserComment, Author, Creator).** However, all aspects regarding users (as well as devices) are going to be specified in Supplementary Metamodel (see section 3), while CAM Core Metamodel only references to those specifications through attributes: attribute *AuthorProfileReference* defined in **Author** and *UserProfileReference* in other classes.

- For attributes that are referencing to the external structures, URL references are used, and thus a suffix *URI* is used in naming. Some examples of applying this rule are the following:
    - In **ExternalMetadata** class, attributes *SchemaURI, MetadataURI* (see Figure 10 : Common metadataFigure 10).

Common metadata

One instance of **CAMElementMetadata** (with its associations) corresponds to the concrete version of CAM Element Metadata description. The description being versioned is identified through *CAMElementMetadataID*, and versions inside the description are identified by *VersionNumber*.

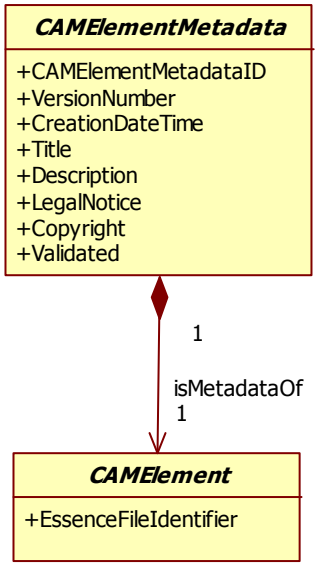

Figure 9: CAMElementMetadata definition

CAMElement			
Identifiers: *InternalID*[1]			
Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
EssenceFileIdentifier	The reference to the physical file bearing the content (essence file)	Optional	Yes

[1] *InternalID* stands for the identifier of the class that will be generated by the system. This is case for those classes in CAM Core Metamodel that do not have their own semantic identifiers.

CAMElementMetadata			
Identifiers: *CAMElementMetadataID & VersionNumber*			
Attribute name	***Description***	***Optional/Mandatory***	***Modifiable (inside version)***
CAMElementMetadataID	Identifier of CAM Element Metadata description.	Mandatory	No
VersionNumber	Version number of CAM Element Metadata description. Inside the same description (for the same CAMElementMetadataID), version number must be unique.	Mandatory	No
CreationDateTime	Date and time of registering CAM Element Metadata version in the system	Mandatory	No
Title	The title given to described content	Optional	No
Description	The textual description of the content	Optional	No
Copyright	Authorship information over the essence file	Optional	No
LegalNotice	Legal issues (i.e "Not under 10 years") related to	Optional	No
Validated	Boolean information on whether CAM Element Metadata description has been validated (true for valid, false for invalid).	Mandatory	Yes

CAMElementMetadataID, VersionNumber and *CreationDateTime* is the mandatory part of metadata description. The rest of CAMElementMetadata description is optional, and it includes:

- *Title*, *Description*, *Copyright*, *LegalNotice* (In **CAMElementMetadata** class, see Figure 9)
- **Author, Creator, SocialTag, ExtenralMetadata, UserComment, UserRating** (associated to **CAMElementMetadata** class, see Figure 10)

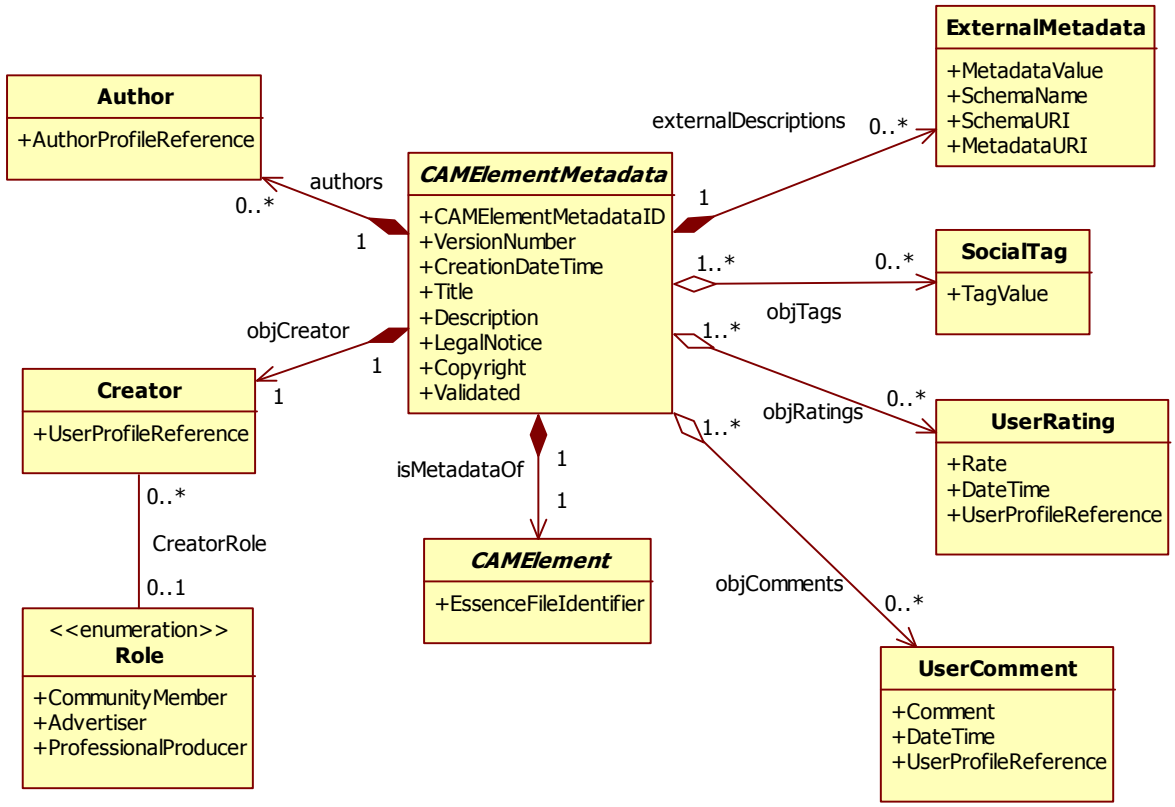

Figure 10 : Common metadata

Association name	***Description***
authors	Content described by CAM Element Metadata can have multiple authors.
objCreator	CAM Element Metadata is registered in the system by one actor (**Creator**) who can assume none or one role in the producer/consumer chain (*creagtorRole)*. During CAM Element Metadata, creator must not be modified.
creatorRole	Role in producer/consumer chain that **Creator** of CAM Element Metadata can assume.
objTags	Content described by CAM Element Metadata can be tagged with multiple user assigned keywords (**SocialTag**). These tags are free-text and are used to create a folksonomy (e.g a keyword describing one entity in the content: people_watching_TV).
externalDescriptions	Multiple external metadata[2] related to CAM Element essence can be stored, in order that the service or user can get more external information on the content described in CAM4HOME domain. A service or client that is not able to interpret the schema could ignore the contained metadata or try to download a plug-in able to process the schema.
objComments	Potentially multiple comments on the essence content can be left by users. These comments provide better insight of the content CAM Element encapsulates.
objRatings	Potentially multiple user ratings on the essence content can be left by users.

[2] External to CAM4HOME domain.

Author

Identifiers: *AuthorProfileReference*

Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
AuthorProfileReference	The reference to profile of the user or content author in the CAM4HOME domain.	Mandatory	No

Creator

Identifiers: *UserProfileReference*

Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
UserProfileReference	The reference to profile of the user in the CAM4HOME domain.	Mandatory	No
CreatorRole	The role that creator assumed when registering CAMElementMetadata in the system.	Optional	Yes

UserComment

Identifiers: *UserProfileReference & DateTime*

Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
DateTime	Date and time of the comment	Mandatory	No
Comment	Textual comment in free-text	Mandatory	No
UserProfileReference	The reference to profile of the user in the CAM4HOME domain. *XOR* The name the anonymous visitor defined for him/herself.	Mandatory	No

ExternalMetadata

Identifiers: *InternalID*

Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
MetadataValue	External metadata (as string).	Optional	Yes
MetadataURI	If *MetadataValue* is not provided, MetadataURI, specyfing the location from which external metadata can be retreived has to be provided.	Optional (Mandatory if *MetadataValue* is not provided)	Yes
SchemaName	The name of external metadata schema	Optional	Yes
SchemaURI	URI to the schema specification	Optional	Yes

SocialTag

Identifiers: *TagValue*

Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
TagValue	Free-text keyword aimed for tagging the content.	Mandatory	No

UserRatings			
Identifier: *UserProfileReference & DateTime*			
Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
Rate	Mark or rating attributed to the content.	Mandatory	No
DateTime	Date and time when the comment is rated.	Mandatory	No
UserProfileReference	The reference to profile of the user in the CAM4HOME domain. *XOR* The name the anonymous visitor defined for him/herself.	Mandatory	No

Multimedia-related metadata

Metadata specific to multimedia content is further descirbed through the classes organized around **MultimediaElementMetadata**:

Association name	***Description***
contexts	The contexts in which the multimedia content is placed.
conceptsInEssence	The concepts appearing in the multimedia content described through **MultimediaElementMetadata.**
genres	One or multple[3] genres that the multimedia content described through **MultimediaElementMetadata** belongs to.
essenceFileMetadata	Represents the collection of metadata originating from the essence file (i.e format or enconding of the content file, size (kb) of the content file, resolution of the visual content file, sampling rates for auditory content file etc). This metadata is useful for making decision whether essence can be downloaded to resource limited device.

[3] For example ETSI EN 300 468 standard specifies DVB SI descriptor, which has a categorization of program content (Genre) using Content Descriptor.
See details in http://www.dvb.org/technology/standards/a038r3.tm1217r14.dEN300468.V1.8.1.pdf Table 28 in section 6.2.9 for more information.

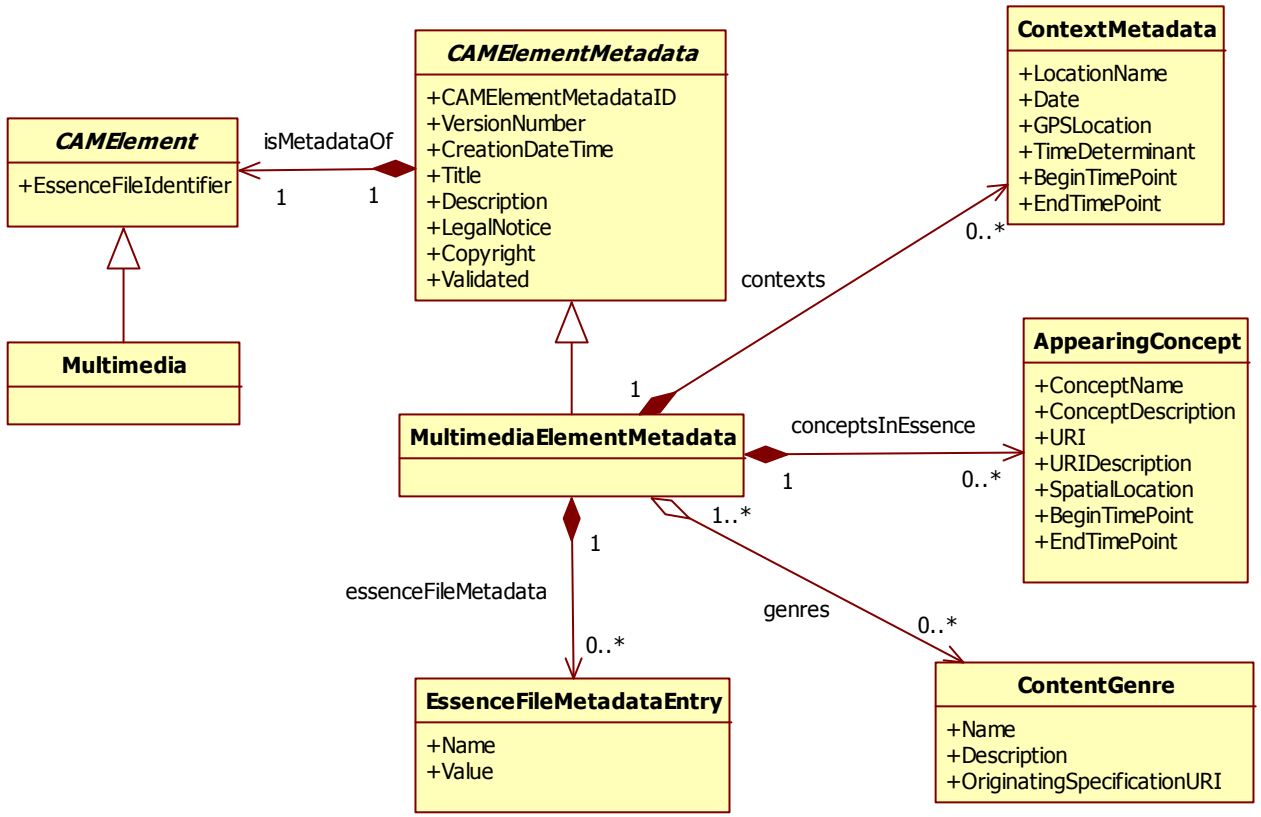

Figure 11: Multimedia-related metadata

ContextMetadata

Identifier: *InternalID*

Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
LocationName	Name of the location where the content is placed.	Optional	Yes
Date	Date of the given context.	Optional	Yes
TimeDeterminant	Time determinant in which the given context is placed.	Optional	Yes
GPSLocation	GPS coordinates (in which form) of the described location.	Optional	Yes
BeginTimePoint	Begin time when the described context appears in the essence (for the description of multiple contexts in time-continuous multimedia).	Optional	Yes
EndTimePoint	End time when the described context appears in the essence (for the description of multiple contexts in time-continuous multimedia).	Optional	Yes

ContentGenre

Identifier: *Name*

Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
Name	Name of the genre.	Mandatory	No
Description	Description of the genre.	Optional	Yes
OriginatingSpecificationURI	The specification (external) from which the genre is originating.	Optional	Yes

AppearingConcept			
Identifer: *InternalID*			
Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
ConceptName	Name of the appearing concept	Mandatory	No
ConceptDescription	Free-text annotation of the concept.	Optional	Yes
URI	Reference to any external location for further information on concept.	Optional	Yes
URIDescription	Summary of the information on appearing concept that can be found in the location defined through *URI* reference.	Optional	Yes
SpatialLocation	Bounding box of the region of appearance using x1, y1, x2, y2 coordinates.	Optional	Yes
BeginTimePoint	Relative time point from the start of the media file where the concept first appears.	Mandatory	No
EndTimePoint	Relative time point from the start of the media file where the concept's appearance stops.	Mandatory	No

EssenceFileMetadataEntry is individual entry of metadata originating from the essence file. For example, this can be a format or enconding of the content file, size (kb) of the content file, resolution of the visual content file, sampling rates for auditory content file etc. This metadata is useful for making decision whether essence can be downloaded to resource limited device.

EssenceFileMetadataEntry			
Identifier: *Name*			
Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
Name	Name of the single metadata (i.e resolution).	Mandatory	No
Value	Value, in free-text format, of that single metadata (i.e 1024x768).	Mandatory	Yes

Service-related metadata

Besides common information, the content of service CAM Element is further descirbed through *ServiceAccessMethod* and several execution *Requirements.*

Association name	***Description***
Requirements	These requirements address hardware and software requirements for executing the service. This is, for example, a platform requirement (Windows) or certain amount of RAM needed to run the application or execute the service.

Service			
Identifier: *InternalID*			
Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
ServiceAccessMethod	Defines how the services is accessed. For example, it can define that the service in question is Web Service so the binding process to use the service can be identified.	Optional	Yes

ExecutionRequirement

Identifier: *Name*

Attribute name	***Description***	***Optional/Mandatory***	***Modifiable***
Name	Name of the concrete requirement	Mandatory	No
Description	Description of the concrete requirement	Optional	Yes
Value	Value of the concrete requirement.	Mandatory	No
OriginatingSpecificationURI	If fixed vocabulary for the requirement is specified somewhere external to CAM4Home metamodel, the URI reference to that specification.	Optional	Yes

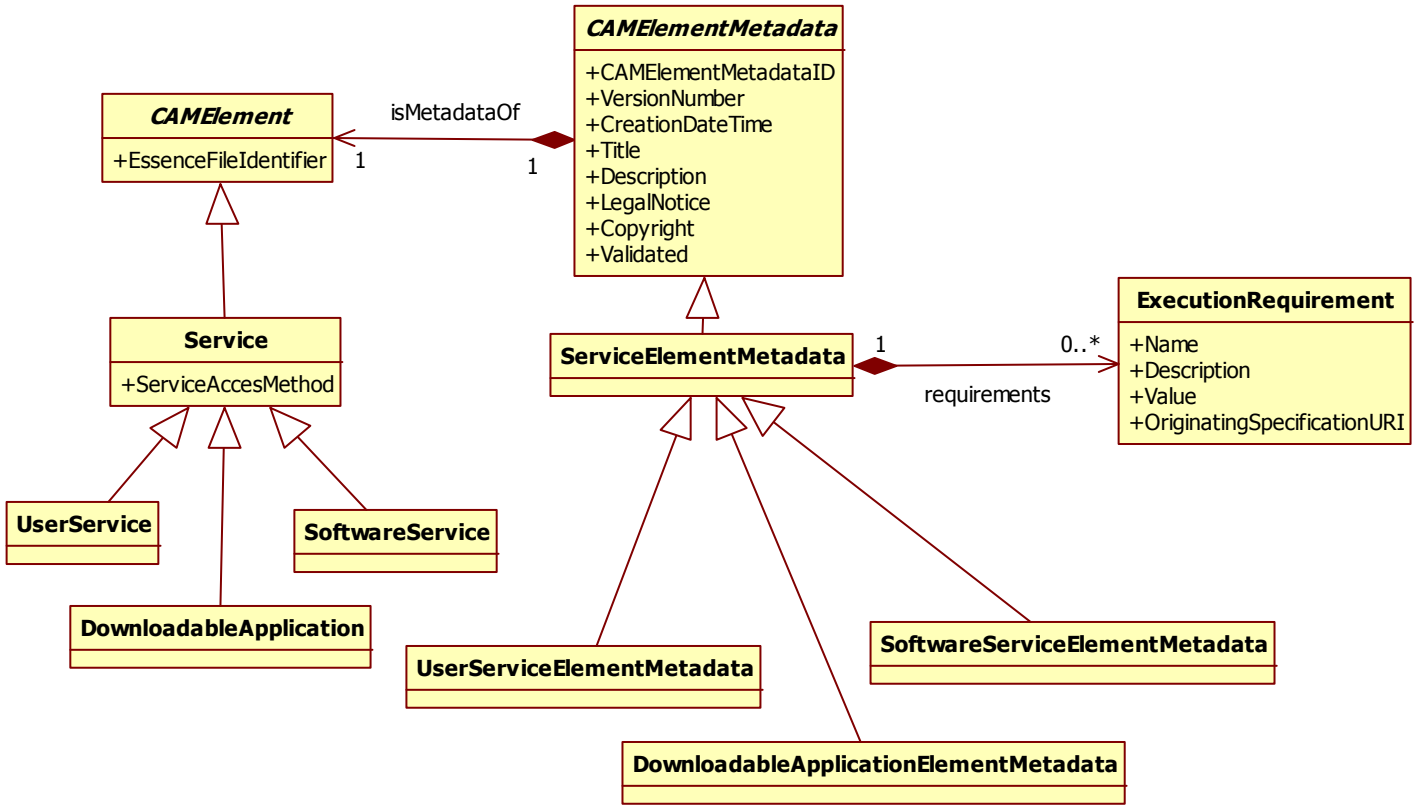

Figure 12: Service-related metadata

4.2.2. CAM Bundle

4.2.2.1. Definitions

CAMBundle

CAM Bundle is considered as the aggregation of two or more CAM Objects and a description of that aggregation, represented within CAM Bundle Metadata. A plain aggregation, not containing any description of the aggregation of CAM Objects is not to be considered as CAM Bundle.

In the model, CAMBundle is used as the term to address valid CAMBundleMetadata.

CAMBundleMetadata

Only valid **CAMElementMetadata**, that is, **CAMObject**s can be aggregated in the bundle. CAMBundleMetadata is a description of aggregation of **CAMObject**s in the bundle.

When bundle contains two or more **CAMObjects**, as well as the description of their aggregation, **CAMBundleMetadata** is considered as valid (this validation needs to be explicitly done through the operations, see section 4.3.3). Valid CAMBundleMetadata is addressed as CAMBundle.

One CAM Bundle Metadata description can be versioned. Versions keep trace of the description's evolution. Rules are defined to determine which modifications in the description trigger the creation of its new version.

Concrete instance of **CAMBundleMetadata** (with its associations) corresponds to the concrete version of CAM Bundle Metadata description. The description being versioned is identified through *CAMBundleMetadataID*, and versions inside the description are identified by *VersionNumber*.

4.2.2.2. CAM Bundle Metadata specification

4.2.2.2.A. Versioning

CAM Bundle Metadata description (if valid addressed as CAMBundle) can have multiple versions. Versions allow the evolution of the descriptions during the lifecycle of CAM Element Metadata and CAM Bundle Metadata. Rules define which modifications in the description trigger the creation of a new version. Regarding their effect on versioning the metadata, following cases can be identified:

1) *CAMBundleMetadataID* and metadata on the creator of CAM Bundle Metadata description *(bdlCreator*) cannot be modified inside the same *CAMBundleMetadataID*.
2) *VersionNumber* and *CreationDateTime* cannot be modified inside the version.

3) As for metadata that can be modified, some metadata modifications does not require creating new versions.These metadata are addressed as *dynamic* (i.e. their values can change inside the same version). Following metadata is currently defined as dynamic:

- *UserComments*
- *UserRatings*
- *SocialTags*

4) In addition to the metadata described above, all other metadata is addressed as *static*, and their modification should trigger the creation of a new version of Bundle.Some examples of static metadata are:

- *Title*, *Description*, *Copyright*, *LegalNotice*, etc.

New version of CAM Bundle Metadata would be equal to duplication of the current version contents and modificaiton of intended metadata.

Remark on the realization of dynamic metadata in the versioning context: Since *dynamic* metadata contain values that are updated constantly by the users of the system, a mechanism is required to allow accessibility to the latest version. The recommended mechanism is defined hereafter:

Dynamic metadata consists of three entities: *ReflectorURI* and *CachedMetadata* and *RetrievedOnDate*.

ReflectorURI contains a URI to a location where the latest contents of the metadata are maintained and can be retrieved to a local structure. One *ReflectorURI* is required for every dynamic metadata entity, e.g all *UserComments* should have exactly one *ReflectorURI*.

CachedMetadata contains metadata content that has been retrieved from the *ReflectorURI* on a date described in *RetrievedOnDate* entity. *CachedMetadata* is not guaranteed to be up-to-date, but it allows consumption of the metadata when the location in *ReflectorURI* is unreachable.

Each new version of CAM Bundle Metadata is defined as invalid until it is explicitly validated (*Validated* attribute). Validation is an activity that happens at the end of distribution and before the beginning of delivery phase (of Bundle life cycle). Invalid metadata is not allowed to be registered for delivery in the system.

In the context of CAM Bundle Metadata, validation activity should check whether a CAM Bundle Metadata version respects Bundle composition rule (see section 4.3.1, 4.3.2).

4.2.2.2.B. Metadata

Metadata description of a bundle provides the description of the aggregation of CAMObjects in that bundle.

Modeling choices regarding metadata within versions

Versions allow the evolution of the descriptions during the lifecycle of CAM Bundle Metadata. New versions are created by duplicating current version contents and modification of intended metadata.

Metadata being shared by different versions of metadata description under the same *CAMBundleMetadataID* is said to have family scope. Entities presenting those metadata (**UserComment, UserRating, SocialTag**) are associated with **CAMBundleMetadata** class by means of aggregation (see Figure 13, Figure 14). The cardinalities for mentioned associations (*bdlRatings, bdlComments, bdlTags*) reflect this: association cardinalities on the side of **CAMBundleMetadata** are 1..*.

Metadata having the lifespan of a version (*CAMBundleMetadataID,VersionNumber*) is said to have version scope. Entities presenting those metadata (**ObjectsRelationship)** are associated with **CAMBundleMetadata** class by means of composition (see Figure 13, Figure 14). The cardinalities for mentionned associations (*relationshipsInBundle*) reflect this: association cardinalities on the side of **CAMBundleMetadata** are 1.

Remark on the usage of references in metamodel

In CAM Core Metamodel, the following rules are applied for representing (and naming attributes containting) references:

- For attributes addressing metadata structures within CAM Metamodel, a suffix *Reference* is used:
    - Several metadata entities have attributes addressing users of the system (**UserRating, UserComment, Author, Creator).** However, all aspects regarding users (as well as devices) are going to be specified in Supplementary Metamodel (see section 3), while CAM Core Metamodel only references to those specifications through attributes: attribute *AuthorProfileReference* defined in **Author** and *UserProfileReference* in other classes.
- For attributes that are referencing to the external structures, URL references are used, and thus a suffix *URI* is used in naming.

Metadata

Concrete instance of **CAMBundleMetadata** (with its associations) corresponds to the concrete version of CAM Bundle Metadata description. The description being versioned is identified through *CAMBundleMetadataID*, and versions inside the description are identified by *VersionNumber*.

CAMBundleMetadata			
Identifier: *CAMBundleMetadataID & VersionNumber*			
Attribute name	***Description***	***Optional/Mandatory***	***Modifiable (inside version)***

CAMBundleMetadataID	Identifier of CAMBundleMetadata description.	Mandatory	No
VersionNumber	Version number of CAM Element Metadata description. Inside the same description (for the same CAMElementMetadataID), version number must be unique.	Mandatory	No
CreationDateTime	Date and time of registering CAM Bundle Metadata version in the system	Mandatory	No
Title	The title given to bundle's content	Optional	No
Description	The textual description of the bundle's content	Optional	No
Copyright	Authorship information over the essence file	Optional	No
LegalNotice	Legal issues (i.e "Not under 10 years") related to	Optional	No
Validated	Boolean information on whether CAM Bundle Metadata description has been validated (true for valid, false for invalid).	Mandatory	Yes

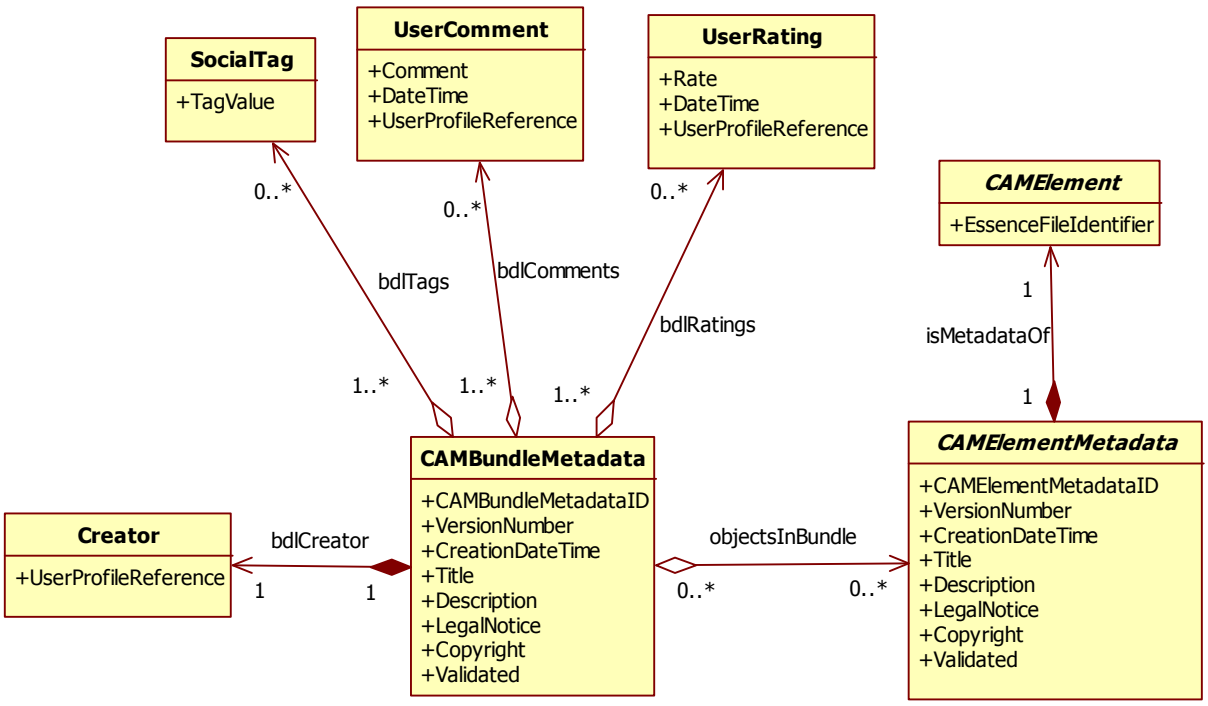

Figure 13: CAM Bundle

Association name	***Description***
bdlCreator	**CAMBundleMetadata** is registered in the system by one actor (**Creator**) who can assume one role in the producer/consumer chain. During CAM Bundle Metadata lifecycle, the information on creator must not be modified.
bdlTags	Content described by CAM Bundle Metadata can be tagged with multiple user assigned keywords (**SocialTag**). These tags are free-text and are used to create a folksonomy (e.g a keyword describing one entity in the content: people_watching_TV).
bdlComments	Potentially multiple comments on the Bundle content can be left by users. These comments provide better insight of the content.
bdlRatings	Potentially multiple user ratings on the bundle can be left by users.
objectsinBundle	CAM Element Metadata (versions) being aggregated into a bundle (version). One CAM Element Metadata version can be aggregated to multiple versions of the bundle inside the family (same *CAMBundleMetadataID*), as well as to multiple bundle families (having different *CAMBundleMetadataID*).

4.2.2.3. Relationships between CAM Objects inside CAM Bundle

The following figure presents relationships between CAM objects inside a bundle (*ObjectsRelationship)*. The semantics of those relationships is not currently specified in more detail.

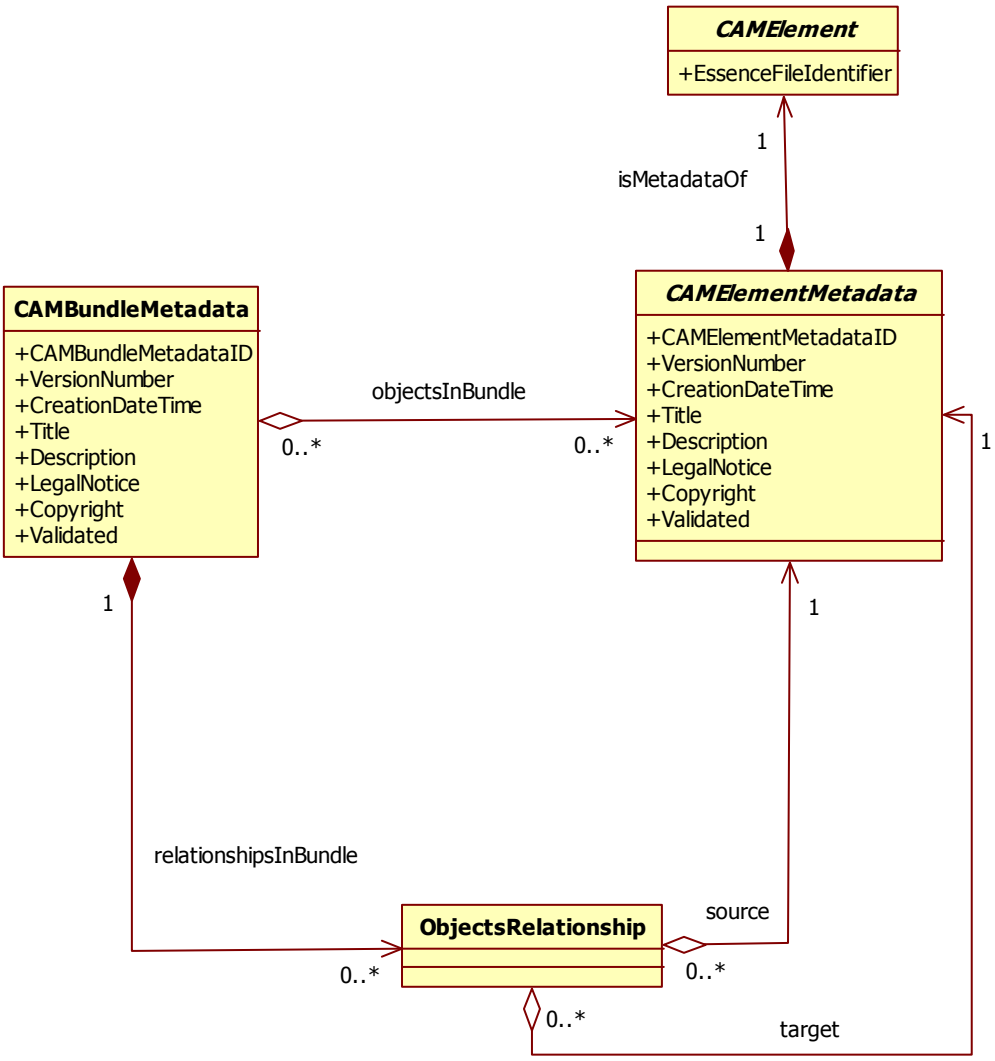

Figure 14: Relationships between objects in a bundle

Association name	*Description*
relationshipsInBundle	The relationships between objects aggregated in a bundle (version).
source	CAM Object being source of the relationship
target	CAM Object being target of the relationship

4.2.2.3.A. Types of relationships between CAM Objects inside CAM Bundle

The CAM Core Metamodel currently does not treat in detail the relationship types between objects aggregated into bundle. It only represents the existence of that relationship.

From the requirements specified in D1.3, different types of relationships are needed in order to support implementation scenarios:

- Synchronization relationships aimed to specify for instance that 2 instrument scores have to be played with a specific delay. This type of relationships is not part of the Core Metamodel but of the Supplementary Metamodel. They are also dealt with in the scope in particular of the W3C SMIL specification [http://www.w3.org/AudioVideo/].
- Structural relationships, in particular PartOf: when for instance a Picture is part of a Video, or it has been extracted from it (Derivative relationship). Restrictions may exist on the possibility to create a relationship between 2 Elements that are of different types. The rules to apply to Relationships have to be defined in the scope of the different applications.
- Specific Delivery relationships, including the definition of Alternative versions for delivery.
- Descriptive relationships, including the existence of additional (user-generated or not) complementary descriptions (potentially structured as descriptive metadata) about an Element. Those include Chronological relationships for Elements that represent a sequence.

However, CAM Core meta-model currently limits to only representing the existence of the relationship between objects in the bundle. The semantics of many of those relationships can be discovered as an interpretation by the user. The addition of its semantics is dependent on domain applications. It belongs either to the Supplementary Metamodel or to the External Metadata.

4.3.CAM Core Behavior

The CAM Core behavior tackles the allowed treatments on entities defined in the CAM Core structure.

4.3.1. CAM Bundle – life cycle view

The lifecycle for CAM Bundle is composed of 6 phases [1]:

- Creation
- Modification
- Distribution
- Delivery
- Interpretation
- Consumption

During its lifecycle, a bundle can be in two different statuses:

- *Invalid Bundle* - a Bundle not respecting its composition integrity rules (see section 4.3.2). That means that any Bundle aggreagting less than two CAM Objects, or not containing the description of that aggregation (CAM Bundle Metadata), is considered as *Invalid*.
- *Valid Bundle* – A Bundle respecting its composition integrity rule (see section 4.3.2), which is explicitly validated, becomes valid CAM Bundle.

Any modification occuring after a Bundle is validated, can invalidate a Bundle and thus requires the explicit re-validation of respecting integrity rules.

Bundle can remain invalid during its 3 initial lifecycle phases (*Creation, Modification and Distribution*). In order to be published (starting from Delivery phase), Bundle needs to be valid. When this is the case, bundle is addressed as CAM Bundle.

In the model, a bundle is presented through **CAMBundleMetadata** and its associations with other metadata entities. This description being versioned, the validity of bundles is resolved in its latest version's scope. This means that the Bundle will be considered valid if its last version is valid.

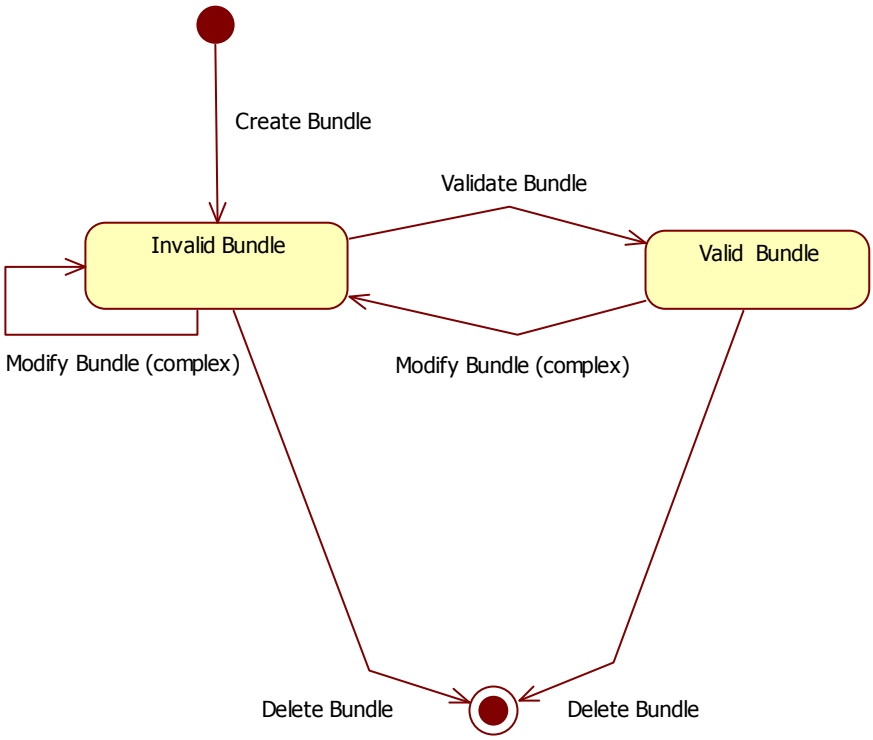

Figure15: CAM Bundle

4.3.2. CAM Core Integrity Rules

The role of integrity rules is to preserve coherence and consistency of enitities defined in the CAM Core structure.

An integrity rule (IR) is a condition that must be validated by the information stored and manipulated in the system. It is defined on one or more entities or relationships of CAM Core structure.

When all integrity rules are valid, the system is said to be coherent.

An IR is specified as a set: {Condition, Context, Range, Response}

- The *context* of an IR, designates the set of classes affected by its definition and validation.
- The *condition* of an IR must be verified for any state of the system or any change in his state by the objects of context's classes.
- The *range* of an IR is the set of operations which must contain a validation algorithm of the IR, so the coherent state of the system is transformed by this method in another coherent state. The range gathers all operations that might transgress the IR while executed. These operations are called the risk of IR.
- The *response* of an IR indicates the actions to be undertaken when the IR is transgressed. The system may simply reject inconsistent changes or make some compensation changes in order to maintain its coherence.

4.3.2.1. CAM Object rules

IR name	**CAMElementMetadata composition constraint**
Condition	**ImageElementMetadata** can only be associated to **Image** CAMElement. **AudioElementMetadata** can only be associated to **Audio** CAMElement. **DocumentElementMetadata** can only be associated to **Document** CAMElement. **VideoElementMetadata** can only be associated to **Video** CAMElement. **UserServiceElementMetadata** can only be associated to **UserService** CAMElement. **SoftwareServiceElementMetadata** can only be associated to **SoftwareService** CAMElement. **DownloadableApplicationElementMetadata** can only be associated to **DownloadableApplication** CAMElement.
Context	CAMElementMetadata, CAMElement, isMetadataOf association
Range	
Response	
IR description	The metadata attribute set is different depending to the type of content that metadata describes. The constraints on isMetadataOf association between CAMElement and CAMElementMetadata further define which CAMElementMetadata specialization associates with which type of content (corresponding CAMElement specialization).
Related requirements	

IR name	**CAMObject definition (CAMObject = valid CAMElementMetadata)**
Condition	In order for CAMElementMetadata to be considered as valid, the EssenceFileIdentifier referring to content must be defined.
Context	CAMElementMetadata and inherited classes, CAMElement, association isMetadataOf
Range	
Response	
IR description	• CAMElementMetadata that has EssenceFileIdentifier specified is considered as valid.Valid CAMElementMetadata is addressed as CAMObject. • This rule applies for every version of CAMElementMetadata.
Related requirements	• Validation of CAMElementMetadata needs to be done explicitly.

4.3.2.2. CAM Bundle rules

IR name	**CAM Bundle definition (CAMBundle = valid CAMBundleMetadata)**
Condition	In order for CAMBundleMetadata to be considered as valid, it must represent the aggregation of minimum 2 CAMObjects, as well as the description of that aggregation.
Context	CAMBundleMetadata, CAMElementMetadata, association objectsInBundle
Range	
Response	
IR description	• CAMBundleMetadata aggregating two or more CAMObjects as well as the description of that aggregation is considered as valid. Valid CAMBundleMetadata is addressed as CAMBundle. • This rule applies for every version of CAMBundleMetadata.
Related requirements	• Validation of CAMBundleMetadata needs to be done explicitly. • In order to be in delivery phase of CAMBundle lifecycle, CAMBundleMetadata must be valid. Only CAMBundles can be delivered.

4.3.2.3. Versioning rules

IR name	**Dynamic metadata of CAM Element Metadata description**
Condition	Any modification in dynamic metadata of CAM Element Metadata description can be done inside the version, without the need to create a new version of metadata description.
Context	CAMElementMetadata (and its subclasses) and its associations with dynamic metadata entities.
Range	
Response	
IR description	Dynamic metadata in CAM Element Metadata description are: UserComment UserRating SocialTag ContentGenre
Related requirements	Management of versioning of CAM Element Metadata description

IR name	**Dynamic metadata of CAM Bundle Metadata description**
Condition	Any modification in dynamic metadata of CAM Bundle Metadata description can be done inside the version, without the need to create a new version of metadata description.
Context	CAMBundleMetadata and its associations with dynamic metadata entities.
Range	
Response	
IR description	Dynamic metadata in CAM Bundle Metadata description are: UserComment UserRating SocialTag ContentGenre
Related requirements	Management of versioning of CAM Bundle Metadata description

IR name	**Static metadata of CAM Element Metadata description**
Condition	Any modification in static metadata of CAM Element Metadata description yields the creation of new version of that metadata description. The addition or the removal of ExternalMetadata, Author, ContextMetadata, AppearingConcept, EssenceFileMetadataEntry, ExecutionRequirement to CAMElementMetadata or any of its subclasses, as well the update of the Title, Description, Copyright, LegalNotice requires the creation of a new version.
Context	CAMElementMetadata (and its subclasses) and its associations with static metadata entities.
Range	
Response	

IR description	• Static metadata within CAM Element Metadata description are:
	  ○ Title, Description, Copyright, LegalNotice
	  ○ Author, ExternalMetadata, ContextMetadata, AppearingConcept, EssenceFileMetadataEntry, ExecutionRequirement, EssenceFileIdentifier, Creator
	• This rule applies for every version of CAMElementMetadata.

Related requirements	Management of versioning of CAM Bundle Metadata description

IR name	**Static metadata of CAM Bundle Metadata description**
Condition	Any modification in static metadata of CAM Bundle Metadata description yields the creation of new version of that metadata description.
Context	CAMBundleMetadata and its associations with static metadata entities.
Range	
Response	

IR description	• Static metadata within CAM Bundle Metadata description are:
	  ○ Title, Description, Copyright, LegalNotice, Creator
	  ○ relationshipsInBundle
	  ○ ObjectsInBundle
	• This rule applies for every version of CAMBundleMetadata.

Related requirements	Management of versioning of CAM Bundle Metadata description.

4.3.3. Operations

CAM Core Metamodel operations should be the basic bricks enabling the definition of services in CAM platform (as described in D3.1).

4.3.3.1. CAM Object operations

Operation name	**Create CAMElementMetadata (first version)**
Parameters	CAMElementMetadataID, UserProfileReference [, Title, Description, LegalNotice, Copyright, EssenceFileIdentifier.
Pre-conditions	CAMElementMetadataID is new in the system EssenceFileIdentifier is valid
Post-actions	CAMElementMetadata is created. CAMElementMetadata is set to non validated. CreationDateTime is attributed to the CAMElementMetadata. VersionNumber is set to 1. Creator is associated to CAMElementMetadata if UserProfileReference exists, if it does not exist: Creator is created and associated to CAMElementMetadata. CAMElement is created and associated to CAMElementMatedata
Description	Create CAMElementMetadata. Title, Desciption,LegalNotice,Copyright are optional.
Related requirements	

Operation name	**Duplicate CAMElementMetadata version**
Parameters	CAMElementMetadataID, oldVersionNumber, [preserveUserComments, preserveUserRating, preserveSocialTags,preserveContentGenre].
Pre-conditions	CAMElementMetadataID and oldVersionNumber are valid
Post-actions	New version (new instance of CAMElementMetadata) is created and set to non validated. CreationDateTime is attributed to the version. VersionNumber is set to LastVersionNumber+1 Static metadata from previous version is duplicated. If reserveUserComments then add each UserComment (existing in OldVersionNumber) to new version. If preserveUserRating then add each UserRating (existing in OldVersionNumber) to new version If preserveSocialTag then add each SocialTag (existing in OldVersionNumber) to new version If preserveContentGenre then add each ContentGenre (existing in OldVersionNumber) to new version
Description	Create a new version of CAMElementMetadata starting from an existing version by duplicating static and/or dynamic metadata. The new version number corresponds to the last version number + 1.
Related requirements	

Operation name	**Delete CAMElementMetadata version**
Parameters	CAMElementMetadataID, VersionNumber.
Pre-conditions	CAMElementMetadataID and VersionNumber exist. CAMElementMetadata does not belong to any CAMBundleMetadata.
Post-actions	Delete CAMElementMetadata version (having VersionNumber)
Description	Deleting CAMElementMetadata version. Dynamic metadata associated to other CAMElementMetadata (families) are kept in the system.
Related requirements	

Operation name	**Update CAMElementMetadata attributes**
Parameters	CAMElementMetadataID, VersionNumber, NewTitle, NewDescription, NewLegalNotice, NewCopyRight, NewEssenceFileIdentifier.
Pre-conditions	Valid CAMElementMetadataID and VersionNumber Valid EssenceFileIdentifier
Post-actions	New version is created (through operation **DuplicateCAMElementMetadata version**). The values from old version are replicated, and new values are set.
Description	CAMElementMetadata and/or CAMElement are updated.
Related requirements	

Operation name	**Add Author to CAMElementMetadata**
Parameters	CAMElementMetadataID, AuthorProfileReference, VersionNumber
Pre-conditions	CAMElementMetadataID, and VersionNumber exist.
Post-actions	New version is created (through operation **DuplicateCAMElementMetadata version**). Author is added to the new CAMElementMetadata.
Description	Register the author to CAMElementMetadata author list.
Related requirements	

Operation name	**Remove Author from CAMElementMetadata**
Parameters	CAMElementMetadataID, AuthorProfileReference, VersionNumber
Pre-conditions	CAMElementMetadataID, AuthorProfileReference and VersionNumber exist.
Post-actions	New version is created (through operation **DuplicateCAMElementMetadata version**). Author is removed from the new CAMElementMetadata version
Description	Remove Author from CAMElementMetadata
Related requirements	

Operation name	**Add ExternalMetadata to CAMElementMetadata**
Parameters	CAMElementMetadataID, VersionNumber, MetadataValue, SchemaName, SchemaURI, MetadataURI
Pre-conditions	Valid ElementMetadataID, VersionNumber.
Post-actions	New version is created (through operation **DuplicateCAMElementMetadata version**). ExternalMetadata is created and associated to new CAMElementMetadata version.
Description	Add ExternalMetadata to CAMElementMetadata
Related requirements	

Operation name	**Remove ExternalMetadata from CAMElementMetadata**
Parameters	CAMElementMetadataID, VersionNumber, ExternalMetadataID
Pre-conditions	Valid ElementMetadataID, VersionNumber and ExternalMetadataID
Post-actions	New version is created (through operation **DuplicateCAMElementMetadata version**). ExternalMetadata (containing same values as ExternalMetadataID of the old version) is removed from new CAMElementMetadata version.
Description	Remove ExternalMetadata from CAMElementMetadata
Related requirements	

Operation name	**Create UserComment**
Parameters	UserProfileReference, Comment
Pre-conditions	Valid UserProfileReference
Post-actions	UserComment is created. DateTime is attributed by the system. Comment and UserProfileReference for the comment are set.
Description	Create UserComment
Related requirements	This operation cannot be used independently from the operation of adding user comment either to CAMElementMetadata or to CAMBundleMetadata.

Operation name	**Delete UserComment**
Parameters	UserProfileReference, DateTime
Pre-conditions	Valid UserProfileReference, DateTime
Post-actions	UserComment instance is deleted.
Description	Delete UserComment
Related requirements	This operation cannot be used independently from the operation of removing user comment from either CAMElementMetadata or CAMBundleMetadata.

Operation name	**Update UserComment**
Parameters	UserProfileReference, DateTime, New comment
Pre-conditions	Valid UserProfileReference, DateTime
Post-actions	Textual comment of UserComment is updated.
Description	Update UserComment with the new textual comment left by user.
Related requirements	

Operation name	**Add UserComment to CAMElementMetadata**
Parameters	UserProfileReference, DateTime, CAMElementMetadataID, VersionNumber
Pre-conditions	Valid UserProfileReference, DateTime, CAMElementMetadataID and VersionNumber
Post-actions	UserComment is created (through operation **CreateUserComment).** UserComment is associated to CAMElementMetadata
Description	UserComment is added to CAMElmentMetadata.
Related requirements	

Operation name	**Remove UserComment from CAMElementMetadata**
Parameters	UserProfileReference, DateTime, CAMElementMetadataID, VersionNumber
Pre-conditions	Valid UserProfileReference, DateTime, CAMElementMetadataID and VersionNumber
Post-actions	UserComment is removed from CAMElementMetadata UserComment is deleted (through operation **DeleteUserComment**).
Description	Remove UserComment from CAMElementMetadata.
Related requirements	

Operation name	**Create UserRating**
Parameters	UserProfileReference, Rate.
Pre-conditions	Valid UserProfileReference
Post-actions	UserRating is created DateTime, UserProfileReference, and Rate are attributed to UserRating.
Description	Create UserRating
Related requirements	This operation can not be used independently from adding UserRating either to CAMElementMetadata or to CAMBundleMetadata.

Operation name	**Delete UserRating**
Parameters	UserProfileReference, DateTime
Pre-conditions	Valid UserProfileReference
Post-actions	UserRation is deleted
Description	Remove this UserRating from all objects and bundle families Delete UserRating
Related requirements	This operation cannot be used independently from the operation of removing user rating from either CAMElementMetadata or CAMBundleMetadata.

Operation name	**Update UserRating**
Parameters	UserProfileReference, DateTime, New Rating
Pre-conditions	Valid UserProfileReference
Post-actions	UserRating is updated with new rate.
Description	Update UserRating.
Related requirements	

Operation name	**Add UserRating to CAMElementMetadata**
Parameters	UserProfileReference, DateTime, CAMElementMetadataID
Pre-conditions	Valid UserProfileReference and CAMElementMetadataID
Post-actions	UserRating is created (through operation **Create UserRating).** UserRating is associated to CAMElementMetadata
Description	Add UserRating to CAMElementMetadata
Related requirements	

Operation name	**Remove UserRating from CAMElementMetadata**
Parameters	UserProfileReference, DateTime, CAMElementMetadataID
Pre-conditions	Valid UserProfileReference and CAMElementMetadataID
Post-actions	UserRating is removed from CAMElementMetadata UserRating is deleted (through operation **DeleteUserRating**).
Description	Remove UserRating from CAMElementMetadata
Related requirements	

Operation name	**Create SocialTag**
Parameters	TagValue
Pre-conditions	TagValue does not already exist in the system.
Post-actions	SocialTag is created and TagValue is set
Description	Create SocialTag
Related requirements	This operation can not be used independently from adding SocialTag either to CAMElementMetadata or to CAMBundleMetadata.

Operation name	**Delete SocialTag**
Parameters	TagValue
Pre-conditions	TagValue exists and is not associated to any bundle or object.
Post-actions	SocialTag is Deleted
Description	Delete SocialTag
Related requirements	This operation can not be used independently from removing SocialTag either from CAMElementMetadata or CAMBundleMetadata.

Operation name	**Add SocialTag to CAMElementMetadata**
Parameters	SocialTag, CAMElementMetadataID, VersionNumber
Pre-conditions	Valid CAMElementMetadataID, VersionNumber
Post-actions	If SocialTag doesn't exist in social tags (in the system), create SocialTag (through operation **CreateSocialTag).** SocialTag is associated to CAMElementMetadata.
Description	Add SocialTag to CAMElementMetadata
Related requirements	

Operation name	**Remove SocialTag from CAMElementMetadata**
Parameters	SocialTag, CAMElementMetadataID, VersionNumber
Pre-conditions	Valid SocialTag , CAMElementMetadataID, VersionNumber
Post-actions	SocialTag is removed from CAMElementMetadata
Description	SocialTag is removed from CAMElementMetadata
Related requirements	

Operation name	**Validate CAMElementMetadata version**
Parameters	CAMElementMetadataID, VersionNumber
Pre-conditions	CAMElementMetadataID and VersionNumber exist
Post-actions	If CAMElement is associated with the CAMElementMetadata version and the EssenceFileIdentifier is valid, then CAMElementMetadata version is validated (validated = true).
Description	Validate a CAMElementMetadata version. If validated a CAMElementMetadata version is designated as CAMObject.
Related requirements	

Operation name	**Invalidate CAMElementMetadata version**
Parameters	CAMElementMetadataID, VersionNumber
Pre-conditions	CAMElementMetadataID and VersionNumber exist
Post-actions	CAMElementMetadata is invalidated (validated = false). Invalidate each CAMBundleMetadata version to whom this CAMElementMetadata instance is associated with (if they have not reached the delivery phase).
Description	Invalidate a CAMElementMetadata version and related CAMBundleMetadata versions.
Related requirements	

Service and ServiceElementMetadata

Operation name	**Create ServiceElementMetadata (first version)**
Parameters	CAMElementMetadataID, UserProfileReference [, Title, Description, LegalNotice, Copyright, EssenceFileIdentifier, ServiceAccessMethod.]
Pre-conditions	CAMElementMetadataID is new in the system EssenceFileIdentifier is valid
Post-actions	ServiceElementMetadata is created ServiceElementMetadata is set to non validated. CreationDateTime is attributed to the ServiceElementMetadata VersionNumber is set to 1. Creator is associated to ServiceElementMetadata if UserProfileReference exists, if it does not exist: Creator is created and associated to ServiceElementMetadata. Service is created. EssenceFileIdentifier and ServiceAccessMethod for Service are set. Service is associated to ServiceElementMatedata.
Description	Create ServiceElementMetadata. Title, Desciption,LegalNotice,Copyright, EssenceFileIdentifier and ServiceAccessMethod are optional.
Related requirements	

Operation name	**Duplicate ServiceElementMetadata version**
Parameters	CAMElementMetadataID, oldVersionNumber [, preserveUserComments, preserveUserRating, preserveSocialTags,preserveContentGenre].
Pre-conditions	CAMElementMetadataID and OldVersionNumber are valid
Post-actions	Create new CAMElementMetadata version (CAMElementMetadataID,oldVersionNumber, [, preserveUserComments, preserveUserRating, preserveSocialTags,preserveContentGenre]) Static metadata from previous ServiceElementMetadata version is duplicated.
Description	Create a new version of CAMElementMetadata starting from an existing version by duplicating static and/or dynamic metadata. The new version number corresponds to the last version number + 1.
Related requirements	

Operation name	**Add ExecutionRequirement to ServiceElementMetadata**
Parameters	CAMElementMetadataID, VersionNumber, Name, Description, Value, Origin atingSpecificationURI
Pre-conditions	CAMElementMetadataID and VersionNumber exist There is no another ExecutionRequirement associated with the current CAMElementMetadata with the same Name.
Post-actions	New version of ServiceElementMetadata is created (through operation **DuplicateServiceElementMetadata version**). ExecutionRequirement is created and Name, Description, Value,OriginatingSpecificationURI are set. ExecutionRequirement is added to ServiceElementMetadata.
Description	Add ExecutionRequirement to ServiceElementMetadata
Related requirements	

Operation name	**Remove ExecutionRequirement from ServiceElementMetadata**
Parameters	Name, CAMElementMetadataID, VersionNumber
Pre-conditions	Name, CAMElementMetadataID, VersionNumber exist
Post-actions	New version of ServiceElementMetadata is created (through operation **DuplicateServiceElementMetadata version**). ExecutionRequirement is removed from new version of ServiceElementMetadata.
Description	Remove ExecutionRequirement from ServiceElementMetadata
Related requirements	

MultimediaElementMetadata

Operation name	**Duplicate MultimediaElementMetadata version**
Parameters	CAMElementMetadataID, oldVersionNumber [preserveUserComments, preserveUserRating, preserveSocialTags,preserveContentGenre].
Pre-conditions	CAMElementMetadataID and OldVersionNumber are valid.
Post-actions	Duplicate CAMElementMetadata version (CAMElementMetadataID,oldVersionNumber[, preserveUserComments, preserveUserRating, preserveSocialTags,preserveContentGenre]) Static metadata from previous MultimediaElementMetadata version is duplicated.
Description	Create a new version of MultimediaElementMetadata starting from an existing version
Related requirements	

Operation name	**Create VideoElementMetadata (first version)**
Parameters	CAMElementMetadataID, UserProfileReference [, Title, Description, LegalNotice, Copyright, EssenceFileIdentifier, StreamingType, VideoType].
Pre-conditions	CAMElementMetadataID is new in the system EssenceFileIdentifier is valid
Post-actions	VideoElementMetadata is created. VideoElementMetadata is set to non validated. CreationDateTime is attributed. VersionNumber is set to 1. If UserProfileReference exists, Creator is associated to VideoElementMetadata. If it does not exist: Creator is created and associated to VideoElementMetadata. Video is created. EssenceFileIdentifier, StreamingType and VideoType are set to Video. Video is associated to VideoElementMetadata.
Description	Create VideoElementMetadata. Title, Desciption,LegalNotice,Copyright are optional.
Related requirements	

Operation name	**Duplicate VideoElementMetadata version**
Parameters	CAMElementMetadataID, oldVersionNumber [, preserveUserComments, preserveUserRating, preserveSocialTags, preserveContentGenre].
Pre-conditions	CAMElementMetadataID and OldVersionNumber are valid.
Post-actions	Create new MultimediaElementMetadata version (CAMElementMetadataID, oldVersionNumber [, preserveUserComments, preserveUserRating, preserveSocialTags,preserveContentGenre]. New Video instance is created. The existing StreamingType, VideoType are attributed to the new Video instance.
Description	Create a new version of VideoElementMetadata starting from an existing version
Related requirements	

Operation name	**Update VideoElementMetadata attributes**
Parameters	CAMElementMetadataID, VersionNumber, NewTitle, NewDescription, NewLegalNotice, NewCopyRight, NewEssenceFileIdentifier, New VideoType.
Pre-conditions	Valid CAMElementMetadataID and VersionNumber Valid EssenceFileIdentifier
Post-actions	New version is created (through operation **Duplicate VideoElementMetadata version**). The values from old version are replicated, and new values are set for VideoElementMetadata and/or for Video.
Description	VideoElementMetadata and/or Video are updated.
Related requirements	

Operation name	**Create AudioElementMetadata (first version)**
Parameters	CAMElementMetadataID, UserProfileReference [, Title, Description, LegalNotice, Copyright, EssenceFileIdentifier, StreamingType, AudioType].
Pre-conditions	CAMElementMetadataID is new in the system EssenceFileIdentifier is valid
Post-actions	AudioElementMetadata is created. AudioElementMetadata is set to non validated. CreationDateTime is attributed to the AudioElementMetadata VersionNumber is set to 1. If UserProfileReference exists, Creator is associated to AudioElementMetadata. If it does not exist: Creator is created and associated to AudioElementMetadata. Audio is created. EssenceFileIdentifier, StreamingType and AudioType are set to Audio. Audio is associated to AudioElementMetadata.
Description	Create AudioElementMetadata. Title, Desciption,LegalNotice,Copyright are optional.
Related requirements	

Operation name	**Duplicate AudioElementMetadata version**
Parameters	CAMElementMetadataID, oldVersionNumber, [, preserveUserComments, preserveUserRating, preserveSocialTags,preserveContentGenre].
Pre-conditions	CAMElementMetadataID and OldVersionNumber are valid
Post-actions	Create new MultimediaElementMetadata version(CAMElementMetadataID, oldVersionNumber, [, preserveUserComments, preserveUserRating, preserveSocialTags,preserveContentGenre]. New Audio instance is created. The existing StreamingType, AudioType are attributed to the new Audio instance.
Description	Create a new version of AudioElementMetadata starting from an existing version
Related requirements	

Operation name	**Update AudioElementMetadata attributes**
Parameters	CAMElementMetadataID, VersionNumber, NewTitle, NewDescription, NewLegalNotice, NewCopyRight, NewEssenceFileIdentifier, New AudioType.
Pre-conditions	Valid CAMElementMetadataID and VersionNumber Valid EssenceFileIdentifier
Post-actions	New version is created (through operation **DuplicateAudioElementMetadata version**). The values from old version are replicated, and new values are set for AudioElementMetadata and/or Audio.
Description	AudioElementMetadata and/or Audio are updated.
Related requirements	

Operation name	**Create Content Genre**
Parameters	Name, Description, OriginatingSpecificationURI
Pre-conditions	There is no other Content Genre with the same Name.
Post-actions	Content Genre is created. Name, Description, OriginatingSpecificationURI are set to ContentGenre instance.
Description	Create Content Genre
Related requirements	

Operation name	**Delete Content Genre**
Parameters	Name
Pre-conditions	ContentGenre with Name exists. Content Genre Name is not associated to any CAMElementMetadata.
Post-actions	Content Genre is deleted.
Description	Delete Content Genre
Related requirements	

Operation name	**Update Content Genre**
Parameters	Name, New Description, New OriginatingSpecificationURI
Pre-conditions	ContentGenre with Name exists.
Post-actions	Content Genre attributes are updated with new values.
Description	Update Content Genre with new attribute values.
Related requirements	

Operation name	**Add Content Genre to MultimediaElementMetadata**
Parameters	CAMElementMetadataID, Content Genre Name, VersionNumber
Pre-conditions	CAMElementMetadataID, VersionNumber are valid.
Post-actions	If Content Genre Name does not exist, create content genre (through operation **Create Content Genre**) Content Genre is associated to MultimediaElementMetadata.
Description	Add Content Genre to MultimediaElementMetadata
Related requirements	

Operation name	**Remove Content Genre from MultimediaElementMetadata**
Parameters	CAMElementMetadataID, Content Genre Name, VersionNumber
Pre-conditions	CAMElementMetadataID and Content Genre Name exist
Post-actions	Content Genre is dissociated from MultimediaElementMetadata.
Description	Remove Content Genre from MultimediaElementMetadata
Related requirements	

Operation name	**Add ContextMetadata to MultimediaElementMetadata**
Parameters	CAMElementMetadataID, VersionNumber, LocationName, Date, GPSLocation, TimeDeterminant, BeginTimePoint, EndTimePoint.
Pre-conditions	CAMElementMetadataID, VersionNumber are valid.
Post-actions	A new version of MultimediaElementMetadata is created (through operation **DuplicateMultimediaElementMetadata version**) ContextMetadata is created and LocationName, Date, GPSLocation, TimeDeterminant, BeginTimePoint, EndTimePoint are set. ContextMetadata is added to MultimediaElementMetadata
Description	Add ContextMetadata to MultimediaElementMetadata
Related requirements	

Operation name	**Remove ContextMetadata from MultimediaElementMetadata**
Parameters	CAMElementMetadataID, VersionNumber, InternalID from ContextMetadata
Pre-conditions	CAMElementMetadataID, InternalID and VersionNumber exist
Post-actions	A new version of MultimediaElementMetadata is created (through operation **DuplicateMultimediaElementMetadata version**) ContextMetadata is removed from the new version of MultimediaElementMetadata
Description	Remove ContextMetadata from MultimediaElementMetadata
Related requirements	

Operation name	**Add EssenceFileMetadataEntry to MultimediaElementMetadata**
Parameters	CAMElementMetadataID, VersionNumber, Name, Value.
Pre-conditions	CAMElementMetadataID, VersionNumber exist
Post-actions	A new version of MultimediaElementMetadata is created (through operation **DuplicateMultimediaElementMetadata version**) EssenceFileMetadataEntry is created and Name, Value are set. EssenceFileMetadataEntry is added to MultimediaElementMetadata.
Description	Add EssenceFileMetadataEntry to MultimediaElementMetadata
Related requirements	

Operation name	**Remove EssenceFileMetadataEntry from MultimediaElementMetadata**
Parameters	CAMElementMetadataID, VersionNumber, InternalID for EssenceFileMetadataEntry
Pre-conditions	CAMElementMetadataID, VersionNumber and InternalID exist
Post-actions	A new version of MultimediaElementMetadata is created (through operation **DuplicateMultimediaElementMetadata version**) EssenceFileMetadataEntry is removed from the new version of MultimediaElementMetadata
Description	Remove EssenceFileMetadataEntry from MultimediaElementMetadata
Related requirements	

Operation name	**Add AppearingConcept to MultimediaElementMetadata**
Parameters	CAMElementMetadataID, VersionNumber, ConceptName, ConceptDescription, URI, URIDescription, SpatialLocation, BeginTimePoint, EndTimePoint.
Pre-conditions	CAMElementMetadataID, VersionNumber exist There is no another AppearingConcept related to CAMElementMetadata with the same ConceptName
Post-actions	A new version of MultimediaElementMetadata is created (through operation **DuplicateMultimediaElementMetadata version**) AppearingConcept is created and ConceptName, ConceptDescription, URI, URIDescription, SpatialLocation, BeginTimePoint, EndTimePoint are set. AppearingConcept is added to MultimediaElementMetadata
Description	Add AppearingConcept to MultimediaElementMetadata
Related requirements	

Operation name	**Remove AppearingConcept to MultimediaElementMetadata**
Parameters	CAMElementMetadataID, Internal ID from AppearingConcept
Pre-conditions	CAMElementMetadataID and AppearingConcept Identifier exist
Post-actions	A new version of MultimediaElementMetadata is created (through operation **DuplicateMultimediaElementMetadata version**) AppearingConcept is removed from the new version of MultimediaElementMetadata
Description	Remove AppearingConcept to MultimediaElementMetadata
Related requirements	

4.3.3.2. CAM Bundle operations

Operation name	**Create CAMBundleMetadata (first version)**
Parameters	CAMBundleMetadataID, UserProfileReference, [, Title, Description, LegalNotice, Copyright]
Pre-conditions	VersionNumber = 1 CAMBundleMetadataID is not attributed to any CAMBundleMetadata family
Post-actions	CAMBundleMetadata is created CAMBundleMetadata is set to non validated VersionNumber is set to 1. Creator is associated to CAMBundleMetadata if UserProfileReference exists, if it does not exist: Creator is created and associated to CAMBundleMetadata.CreationDateTime is attributed to the CAMBundleMetadata.
Description	Create CAMBundleMetadata. Title, Desciption,LegalNotice,Copyright are optional.
Related requirements	

Operation name	**Duplicate CAMBundleMetadata version**
Parameters	CAMBundleMetadataID, OldVersionNumber, UserProfileReference [, preserveUserComments, preserveUserRating, preserveSocialTags, preserveContentGenre].
Pre-conditions	NewVersionNumber=LastVersionNumber+1 CAMElementMetadataID and OldVersionNumber are valid UserProfileReference is valid
Post-actions	New version (new instance of CAMBundleMetadata) is created and set to non validated. CreationDateTime is attributed to the version. VersionNumber is set to LastVersionNumber+1 Static metadata from previous version is duplicated. If reserveUserComments then add each UserComment (existing in OldVersionNumber) to new version. If preserveUserRating then add each UserRating (existing in OldVersionNumber) to new version. If preserveSocialTag then add each SocialTag (existing in OldVersionNumber) to new version. If preserveContentGenre then add each ContentGenre (existing in OldVersionNumber) to new version.
Description	Create a new version of CAMBundleMetadata starting from an existing version by duplicating static and/or dynamic metadata. The new version number corresponds to the last version number + 1.
Related requirements	

Operation name	**Delete CAMBundleMetadata version**
Parameters	CAMBundleMetadataID, VersionNumber
Pre-conditions	CAMBundleMetadataID and VersionNumber exist
Post-actions	Delete CAMBundleMetadata version (having VersionNumber).
Description	Deleting CAMBundleMetadata version. Dynamic metadata associated to other CAMBundleMetadata (families) are kept in the system.
Related requirements	

Operation name	**Update CAMBundleMetadata attributes**
Parameters	CAMBundleMetadataID, VersionNumber, NewTitle, NewDescription, NewLegalNotice, NewCopyright.
Pre-conditions	Valid CAMBundleMetadataID, VersionNumber.
Post-actions	New version is created (through operation **DuplicateCAMBundleMetadata version**). The values from old version are replicated, and new values are set.
Description	Update CAMBundleMetadata
Related requirements	

Operation name	**Add ObjectsRelationship to CAMBundleMetadata**
Parameters	CAMBundleMetadataID, VersionNumber, Source(CAMElementMetadataID, VersionNumber) Target(CAMElementMetadataID, VersionNumber) [Relationship type Type parameters (depending on the type of retationship)]
Pre-conditions	There is no other relationship with the same Source, Target and Type within the current bundle version. *Depending on the Relationship Type and a specific application additional pre-conditions may exist (eg:for an Acyclic relation the target and the source must differ)*
Post-actions	New version is created (through operation **DuplicateCAMBundleMetadata version**). ObjectRelationship is created and Source, Target and Type are set. ObjectsRelationship is added to the new version of CAMBundleMetadata
Description	Add ObjectsRelationship to CAMBundleMetadata
Related requirements	

Operation name	**Remove ObjectsRelationship from CAMBundleMetadata**
Parameters	CAMBundleMetadataID, CAMBundleMetadata VersionNumber. InternalID of ObjectsRelationship
Pre-conditions	Valid CAMBundleMetadataID, VersionNumber exists. InternalID of ObjectsRelationship exists and is associated with CAMBundleMetadata
Post-actions	New version is created (through operation **DuplicateCAMBundleMetadata version**). ObjectsRelationship is removed from the new version of CAMBundleMetadata
Description	
Related requirements	

Operation name	**Add UserComment to CAMBundleMetadata**
Parameters	UserProfileReference, DateTime, CAMBundleMetadataID, VersionNumber
Pre-conditions	Valid UserProfileReference, DateTime, CAMBundleMetadataID and VersionNumber
Post-actions	UserComment is created (through operation **CreateUserComment).** UserComment is associated to CAMBundleMetadata
Description	Add UserComment to CAMBundleMetadata
Related requirements	

Operation name	**Remove UserComment from CAMBundleMetadata**
Parameters	UserProfileReference, DateTime, CAMBundleMetadataID, VersionNumber
Pre-conditions	Valid UserProfileReference, DateTime, CAMElementMetadataID and VersionNumber
Post-actions	UserComment is removed from CAMBundleMetadata UserComment is deleted (through operation **DeleteUserComment**).
Description	Remove UserComment from CAMBundleMetadata.
Related requirements	

Operation name	**Add UserRating to CAMBundleMetadata**
Parameters	UserProfileReference, DateTime, CAMBundleMetadataID, VersioNumber
Pre-conditions	Valid UserProfileReference and CAMBundleMetadataID, VersioNumber
Post-actions	UserRating is created (through operation **Create UserRating).** UserRating is associated to CAMBundleMetadata
Description	Add UserRating to CAMBundleMetadata
Related requirements	

Operation name	**Remove UserRating from CAMBundleMetadata**
Parameters	UserProfileReference, DateTime, CAMBundleMetadataID, VersionNumber
Pre-conditions	Valid UserProfileReference and CAMBundleMetadataID, VersionNumber
Post-actions	UserRating is removed from CAMBundleMetadata UserRating is deleted (through operation **DeleteUserRating**).
Description	Remove UserRating from CAMBundleMetadata
Related requirements	

Operation name	**Add SocialTag to CAMBundleMetadata**
Parameters	SocialTag, CAMBundleMetadataID, VersionNumber
Pre-conditions	Valid CAMBundleMetadataID, VersionNumber
Post-actions	If SocialTag doesn't exist in SocialTags, create SocialTag (through operation **CreateSocialTag).** SocialTag is associated to CAMBundleMetadata
Description	Add SocialTag to CAMBundleMetadata
Related requirements	

Operation name	**Remove SocialTag from CAMBundleMetadata**
Parameters	SocialTag, CAMBundleMetadataID, VersionNumber
Pre-conditions	Valid SocialTag, CAMBundleMetadataID, VersionNumber
Post-actions	SocialTag is dissociated from CAMBundleMetadata
Description	Remove SocialTag from CAMBundleMetadata
Related requirements	

Operation name	**Validate CAMBundleMetadata version**
Parameters	CAMBundleMetadataID, VersionNumber
Pre-conditions	CAMBundleMetadataID and VersionNumber exist
Post-actions	If CAMBundleMetadata version aggregates at least two CAMObjects and contains metadate description of that aggreation, CAMBundleMetadata version is validated (validated = true).
Description	Validate a CAMBundleMetadata version. If validated a CAMBundleMetadata version is designated as CAMBundle.
Related requirements	

Operation name	**Invalidate CAMBundleMetadata version**
Parameters	CAMBundleMetadataID, VersionNumber
Pre-conditions	CAMBundleMetadataID and VersionNumber exist
Post-actions	If CAMBundleMetadata version does not respect all conditions to be validated, CAMBundleMetadata version is invalidated (validated = false).
Description	Invalidate a CAMBundleMetadata version.
Related requirements	

5. References

[1] D. Pakkala, ITEA-CAM4Home project document, “CAM Concept Definiton, 2008”

[2] ITEA-CAM4Home project deliverable D1.3, Metadata Framework Requirements, 2007

[3] Turki S., Des hyperclasses aux composants pour l'ingénierie des systèmes d'information. PhD thesis, July 2005.

[4] Turki S., Léonard M., Arni-Bloch N., From Hyperclasses to IS Components. Proc. of the 10th International Conference on Concurrent Engineering (CE'2003), July 2003, Madeira, Portugal, P. 235-242, R. Jardim-Goncalves, H. Cha, A. Steiger-Garcao (eds.), Balkema Publishers.

Annex

CAM Element high-level category	CAM Element Type	Description
Service	User Service	A software application that produces added value to user(s) and may utilize software services in its operation. Can be URL to a remotely executable file containing an application that is operable in client devices (e.g. Flash application that cannot be stored for local use)
	Software Service	Software service is an instance of functionality of software executed in one or more host(s) that can be registered, discovered and invoked by other software executed in the same or another host. Software service cannot be utilized directly by an user but needs a User Service or an application to do that.
	Downloadable Application	URL to a downloadable executable file containing an application that is operable in client devices (e.g. a Java jar-file for mobile devices)
MultimediaContent	Video	URL to either : • a downloadable video file of an arbitratry format • a live or recorded video stream available through a streaming service
	Audio	URL to either : • a downloadable audio file of an arbitratry format • a live or recorded audio stream available through a streaming service
	Image	URL to a downloadable image file (jpg, png, bmp, tiff etc.)
	Document	URL to a document file containing formatted text and pictures (e.g. Microsoft Word document, Adobe PDF, etc.)

Table 1. CAM Element categorization