Survey of Metadata and Knowledge for Automated Scripting

Deliverable D3.1.1

FascinatE identifier:	FascinatE-D311-HHI-MetadataSurvey-v10.doc
Deliverable number:	D3.1.1
Author(s) and company:	W. Bailer, R. Kaiser (JRS); A. Engström (TII); J. Ruiz Hidalgo (UPC), A. Kochale (DTO); J.-F. Macq, P. Rondao Alface, N. Verzijp (ALU); M. Masetti, A. Poggi (SES); O. Niamut (TNO); B. Shirley, R. Oldfield (UOS); O. Schreer (HHI); G. Thomas (BBC)
Internal reviewer:	A. Kochale (DTO), O. Schreer (HHI)
Work package / task:	WP3
Document status:	Final Version
Confidentiality:	Public

Version	Date	Reason of change
1	2010-07-15	Document created (e.g. structure proposed, initial input…)
2	2010-07-30	Added definitions and proposed responsibilities
3	2010-09-20	Input from DTO, HHI, SES, UPC, ALU, and JRS
6	2010-10-16	Corrections in content descr., input from UOS and JRS
7	2010-10-17	Input from UOS, text for Scripting Templates
8	2010-10-18	Input from TII, BBC, JRS
9	2010-10-22	Corrections, input from JRS, ALU & others
9.4	2010-10-25	TII input, user annotations, user profiles, rights; gaps
9.8	2010-11-01	Version for internal review
10	2010-11-30	Final version

Acknowledgement: The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 248138.

Table of Contents

1 Executive Summary

This deliverable surveys the types of metadata and knowledge that are relevant in the FascinatE system for automated scripting and beyond. It collects the requirements for the different types of metadata and information about existing formats to represent them. It analyses how they can be aligned and combined, and identifies gaps that need to be filled with application specific formats, some of them possibly leading to future input to standardisation.

This document defines the various types of metadata in the FascinatE system and discusses representation requirements and candidate formats. The document considers various types of metadata describing capture, production, context, content, scripts, network, terminals and users. Representation of essence and essence-like metadata (e.g. depth maps) are out of scope of this document.

The document is related to the overall system requirements (D1.1.1), which are the basis to define the interfaces in the system (D1.4.1) and the requirements for the different types of metadata. This document defines requirements and discusses possible formats and thus serves as a basis for the definition of the FascinatE metadata and knowledge representation model in D3.1.2 (due in July 2011), which will also contain a proposal for filling the gaps identified in this document.

For some types of metadata there are obvious candidate formats, which seem to (mostly) cover the requirements from the FascinatE system. These include content description, rights and licensing, device and network properties and capabilities. For sensor parameters, calibration metadata and user profiles, one or more formats exist that partly cover the requirements. The gaps can be closed by defining extensions for the candidate formats. Finally, for the other types of metadata discussed in this document no obvious candidate format could be identified. For those an application specific format (or a comprehensive extension of an existing format) will need to be defined.

2 Introduction

2.1 Purpose of this Document

This deliverable surveys the types of metadata and knowledge that are relevant for representing and exchanging metadata in the FascinatE system as well as the formats used, and analyses how they can be aligned and combined.

2.2 Scope of this Document

This document defines the various types of metadata in the FascinatE system and discusses representation requirements and candidate formats. The document considers the following types of metadata:

- Sensor Parameters, Calibration Metadata
- Content Description
- Domain/Scene Knowledge
- Production Rules and Visual Grammar
- Script Templates
- Scripts
- User Annotations
- Rights and Licensing
- Device Properties and Capabilities
- Network Properties and Capabilities
- User Profiles
- User Interactions

The document does not discuss representation of

- Audio and video essence and their integral header metadata
- Essence-like metadata (e.g. depth maps)

2.3 Status of this Document

This is the final version of D3.1.1.

2.4 Related Documents

Before reading this document it is recommended to be familiar with the following documents:

- *D1.1.1 End user, production and hardware and networking requirements* defines the FascinatE scenarios, use cases and requirements, from which requirements on the metadata representation can be derived.
- *D2.1.1 Draft specification of generic data representation and coding scheme* describes some metadata elements that are an integral part of the layered scene representation.
- *D5.1.1 AV Renderer Specification and Basic Characterisation of Audience Interaction* provides an initial overview of the possible interactions in the system and serves as a basis for defining requirements on user interaction metadata and terminal properties.

This document serves as a basis for the definition of the FascinatE metadata and knowledge representation model in *D3.1.2* (due in July 2011).

3 Overview

Within the FascinatE system different types of metadata are exchanged between the components. An overview is given in Figure 1 on the following page[1].

The following aspects need to be considered:

- Real-time/non-real-time: metadata created/extracted/processed/transmitted online, or stored for offline processing in the knowledge base/at a network component/at the terminal
- Metadata source
    - Layered scene metadata: everything related to the capture of the scene and required to correctly interpret the scene representation, this metadata is also embedded in the same container as the layered scene essence
    - Analysis metadata: metadata resulting from online or offline content analysis
    - Scene/domain knowledge: static (per event, organisation, etc.) information about setup of the scene, areas of interest, excluded/restricted areas (e.g. audience), knowledge about relevant types of events in the domain
    - Production rules, visual grammar
    - Rights, permissions
    - Network properties/capabilities
    - Terminal properties/capabilities
    - User preferences/profiles
    - Interaction
- Metadata flow: only downstream, only upstream, or bidirectional
- Amount and granularity: per frame/regular interval, one-time/infrequently
- Scope
    - Temporal: to be stored or discarded after some interval, possibly linked to the content by a time stamp
    - Component: to be consumed by receiving component or to be forwarded

[1] The diagram shows the metadata interfaces at the time of writing this document. It may be revised are completing the system architecture deliverable.

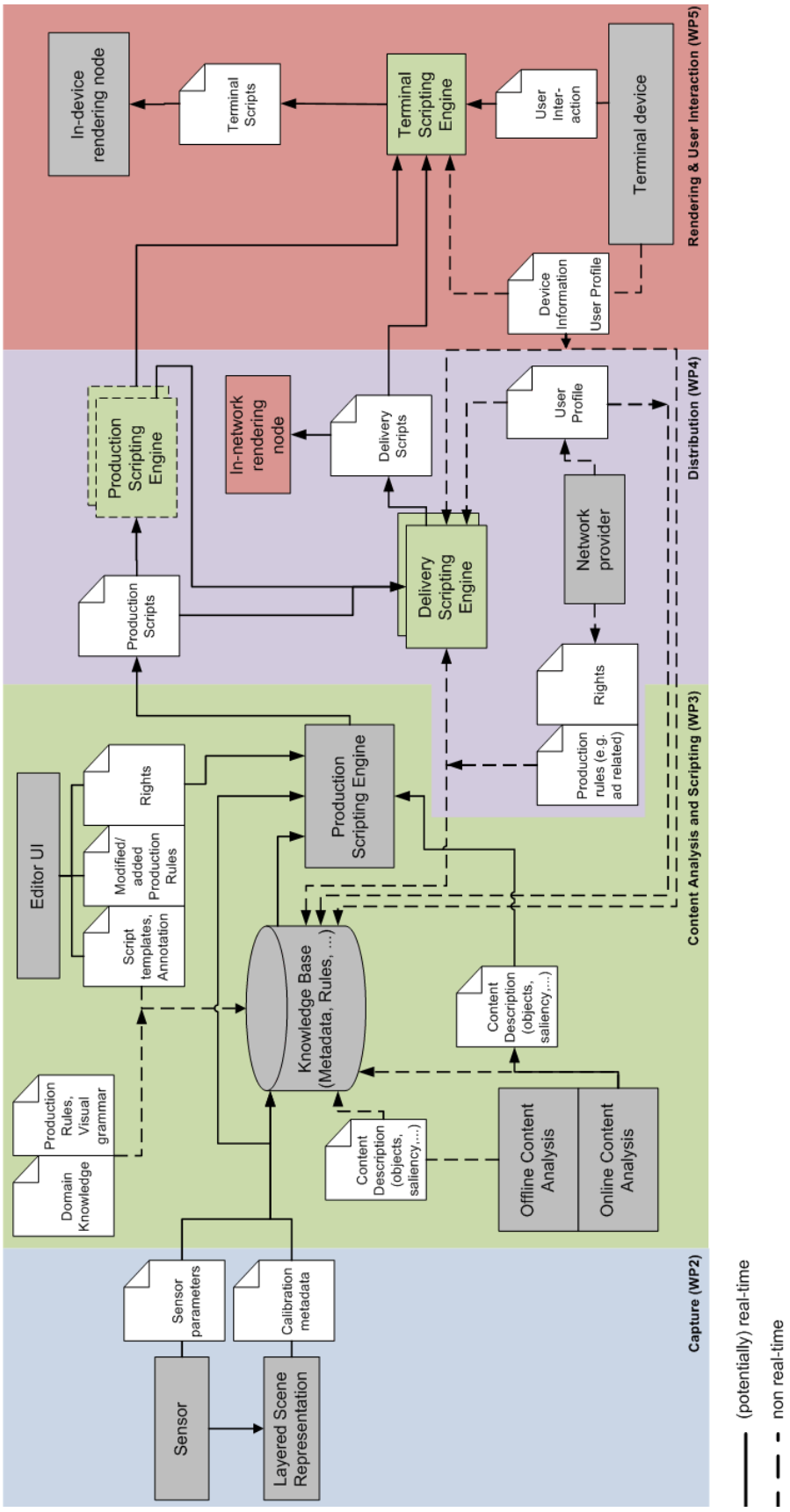


Figure 1: Metadata flow in the FascinatE system.

4 Metadata Types, Requirements and Candidate Formats

4.1 Sensor Parameters, Calibration Metadata

4.1.1 Definition

Position and parameters of audiovisual sensors.

In order to allow a format agnostic representation of the scene, a large variety of metadata from production needs to be made available in different succeeding modules of the FascinatE processing chain. In Deliverable D2.1.1 *Draft Specification of Generic Data Representation and Coding Scheme*[2], the complete set of metadata has been described, which results from a layered scene representation. In Figure 2, a tree structure is presented which builds up a scene with many different layers. Each scene contains one audio scene and one video scene. The video scene usually consists of several spatially distributed camera clusters. The audio scene consist of individual audio objects and one or more sound field representations. Audio objects are recorded with close up microphones or estimated using an array of microphones incl. gunshot microphones. Sound field representation, transmitted as Ambisonics signals, are recorded using special Ambisonics microphones and/or composed out of individual signals.

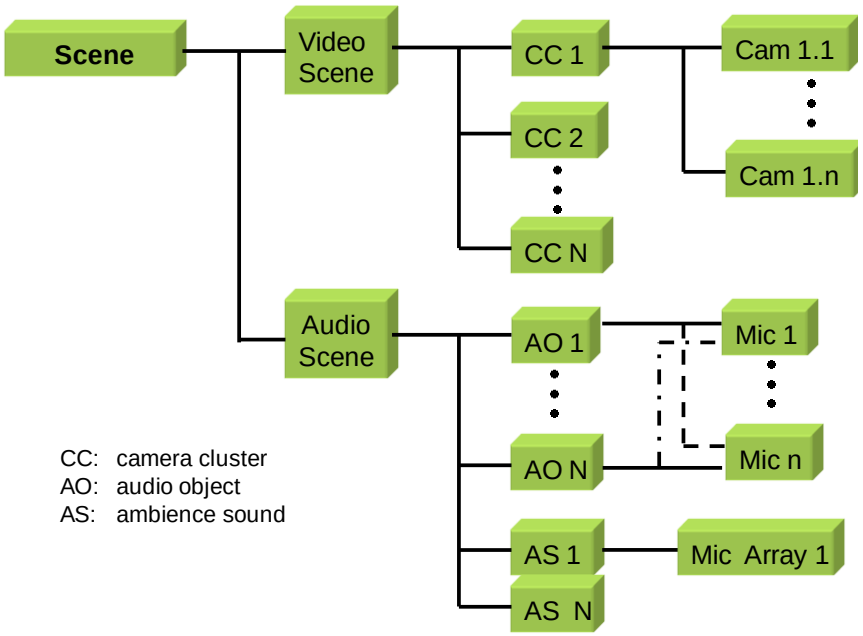


Figure 2: Hierarchical structure of the layered scene description.

4.1.2 Requirements

For each layer of the hierarchical tree structure, a set of metadata as well as their scope has been defined (see D2.1.1, section 3 [Schreer, 2010]). These metadata need to be made available in a common format, which fulfils the following requirements.

- Due to the dynamic nature of the FascinatE scenarios, the metadata structure can be changed. Hence the format must allow a real-time update of values, change, addition or deletion of metadata.

[2] D2.1.1 is a living document, a final specification is scheduled later in the project.

- The metadata can be classified in two main groups of data, either static or dynamic ones. Metadata describing the scene geometry will be defined once during the setup of the complete AV capturing framework. Hence, these metadata remain fixed during the whole lifetime of the live event capturing. On the other hand, some sensors such as accompanied broadcast cameras will pan and zoom during capture. Therefore, internal and external parameters will change accordingly. These metadata have to be updated on a frame basis and represent the dynamic set of metadata. Dynamic metadata also needs to represent the fact that sensor can be switched on and off during the event.
- For audio, a distinction is made between two types of audio objects, explicit and implicit. Explicit sources are close-miced sound sources that will move during capture and therefore must be defined by a dynamic set of metadata. Implicit sound objects are derived from static microphone signals hence microphones used for this can be defined by the static scene description metadata.

Microphone characteristics

- Frequency response*
- Polar response*
- Sensitivity*
- Type (condenser, dynamic etc)*
- Location (x,y,z)
- Position (static or moving)
- Input device (pre-amp) – model and input impedance

* These microphone parameters may be obtained from a FascinatE Transducer Database of possible transducer types at the renderer end, i.e. the metadata needs only contain the microphone make and model.

Metadata for Audio Objects and Sound Fields:

Audio Objects

- Current Position or estimated Area
- Current Direction
- Loudness Unit
- Loudness Range
- True Peak
- Name of Object E.g. referee, Cliff Richard
- Type of Object. E.g. person, drum, car

Sound Field

- Position
- Direction
- Ambisonics Order
- Ambisonics Format type
- Loudness Unit
- Loudness Range
- True Peak

For explicit audio objects the location data will be frame dependent thus changes will be tracked with respect to time code. For static microphones the location will be a fixed coordinate in space for each microphone.

4.1.3 Candidate Formats

The metadata of the layered scene description can be represented in the following tree structure (Figure 3), whereby each part of the tree contains a set of metadata. As it can be seen in the figure below, some parameters are frame based, which requires a dynamical update of the metadata during capture and processing.

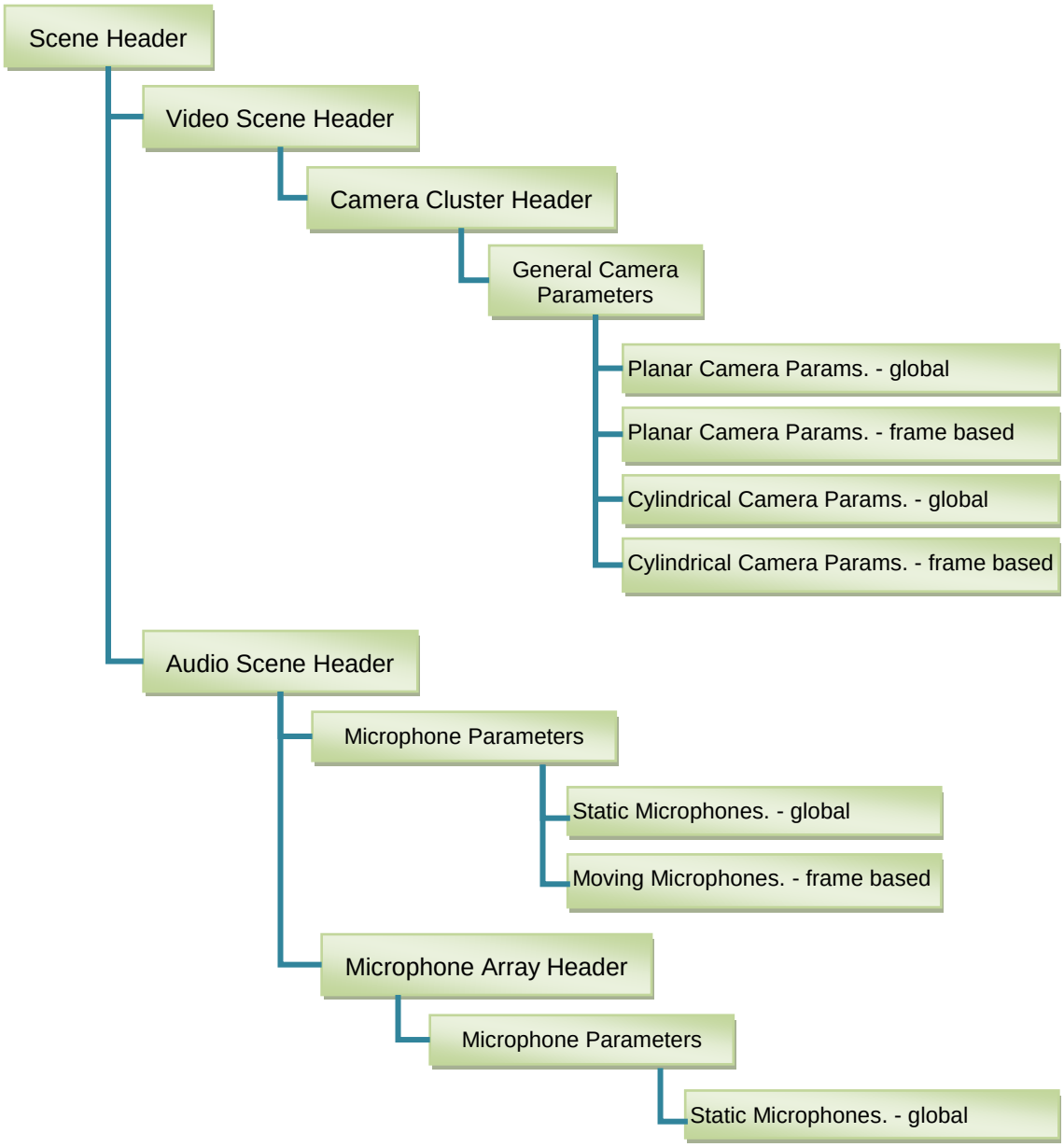


Figure 3: Hierarchical structure of related metadata.

SMPTE RP210

The SMPTE Metadata Dictionary [RP210, 2008] defines an extensive list of (mostly technical) metadata properties. Many of the required elements are covered including microphone identifiers, characteristics and placement techniques. However, there is a lack of calibration and lens parameters.

As the name suggests, this standard just describes a dictionary of metadata elements, but no structure to put them into context. The elements are typically KLV encoded and embedded in the headers of MXF (Material Exchange Format)) or DPX (Digital Picture Exchange Format) files. However, the XML (eXtensible markup language) structure defined in SMPTE 434M [MXFXML, 2006] can be used to represent MXF header structures as XML files.

EBU Tech 3301 and HIPS-META

The EBU has defined a set of metadata elements for tapeless cameras [EBU3301, 2005]. Currently this set is being extended and updated as part of the EBU HIPS (Harmonisation and Interoperability of HDTV Production) initiative [HIPS, 2010]. In the draft version from August 2010 there are more parameters than in SMPTE RP210 are covered. However, device position orientation and lens distortion information are missing.

SMPTE RDD 18:2010

This is a registered document provided by Sony and describing a set of camera and lens metadata [RDD18, 2010]. The metadata set has been considered in the development of HIPS-META and is thus a subset.

4.2 Content Description

4.2.1 Definition

Metadata describing low-, mid- or high-level features of audiovisual content, which are typically extracted using automatic tools.

We define the following content description metadata as follows:

- *Low- and mid- level*: Metadata produced by content analysis. They have no or only limited semantic content.
- *High-level*: metadata related to FascinatE attributes (ROI, OOI, available streams, content description, etc.). They are the product of relations between mid-level attributes and have a semantic content, e.g. represent objects, actions or events.

4.2.2 Requirements

The metadata of interest for FascinatE has the following properties:

- *Dynamic*: most of the relevant metadata is time-dependent and only information of a certain time window is relevant.
- *Different granularity*: metadata ranging from detailed description of a region in a frame to a classification of a time segment across all streams.
- *Modality:* some of the features are specific to one modality of stream, while others are created by integrating information from different streams (views) and/or modalities.
- *Strong relation to calibration metadata*: content description is often based on spatiotemporal information about objects in the scene. Thus the system needs calibration metadata to correctly interpret content analysis results obtained for certain areas in the image coordinates of one view and to integrate results from different streams. For streams acquired with a moving camera this includes the change of camera orientation and parameters over time.

In particular, the following metadata for content description is required:

- *Results of tracking feature points (position, confidence for each point in each frame)*: This is considered as an intermediate result to be used for subsequent analysis steps and will not be stored in the metadata store.
- Precise description of object regions and their trajectories over time.

- Detection results of specific classes of objects (e.g. persons), consisting of candidate region and confidence, if applicable also centre position and/or orientation.
- Identification of specific object instances (referencing an instance from a production specific set)
- Classification of actions (time span, objects involved, reference to one of a production specific set of actions), as well as similarity/dissimilarity of observed actions (time span, objects involved, similarity value).
- Amount of activity/saliency/relevance in a certain area of the depicted/recorded scene
- Movement of a close-miced audio source (explicit audio object) such as a referee in a rugby match, whose movements vary with time and must be tracked.
- Co-ordinates of the implicit sound object vertices.

4.2.3 Candidate Formats

MPEG-7

The ISO/IEC standard *Multimedia Content Description Interface* (MPEG-7) [MPEG-7, 2001] has been defined as a format for the description of multimedia content in a wide range of applications. MPEG-7 defines a set of description tools, called description schemes and descriptors. Descriptors represent single properties of the content description, while description schemes are containers for descriptors and other description schemes. The definition of description schemes and descriptors uses the *Description Definition Language* (DDL), which is an extension of XML Schema. MPEG-7 descriptions can be either represented as XML (textual format, *TeM*) or in a binary format (binary format, *BiM*).

A core part of MPEG-7 are the Multimedia Description Schemes (MDS), which provide support for the description of media information, creation and production information, content structure, usage of content, semantics, navigation and access, content organisation and user interaction. Especially the structuring tools are very flexible and allow the description of content on different levels of granularity. In addition, the *Audio* and *Visual* parts define low- and mid-level descriptors for these modalities. The metadata required in FascinatE are covered by the available tools.

This standard has been designed to describe multimedia content and it features all needed tools to describe both audio and video streams. Its key features are:

- Provides tools to describe any kind of multimedia stream, including still pictures, graphics, 3D models, audio, speech, text/font and video.
- Descriptions can be defined both at low level and high level. For example we can describe resolution, colour space and timing at a low level; objects in scene and their motion at a high level.
- Designed to work both in an online and offline environment
- Can be streamed along with multimedia content, or be stored on a server and be accessed remotely
- Comes both in text format (XML, for direct use inside FascinatE scripts) and binary format (more suitable for streaming)
- Supports extensions

MPEG-7 features suit perfectly to the scope of the project: by adopting it the consortium will save the work needed to define a basic set of attributes and its transport methods, focusing only on the development of tools required by its services.

The concept of profiles has been introduced to define subsets of the comprehensive standard which target certain application areas. Three profiles have been standardised: the Simple Metadata Profile (SMP), which describes single instances or collections of multimedia content, the User Description Profile (UDP), containing tools for describing personal preferences and usage patterns of users of multimedia content in order to enable automatic discovery, selection, personalisation and

recommendation of multimedia content, and the *Core Description Profile* (CDP), which consists of tools for describing general multimedia content such as images, videos, audio and collections thereof.

Recently the EBU ECM SCAIE working group[3], to which two FascinatE partners are contributing, has proposed the *AudioVisual Description Profile* (AVDP) which targets applications in audiovisual production using automated content analysis tools.

Low- and mid level metadata

These metadata describe features that are produced by the analysis of multimedia content and cannot be defined at production time. For example, objects detected in the scene and their positions will be included in this category. These data change over time.

- Visual segment: describes a section of a video stream
- Audio segment: describes a section of an audio stream
- Still region: a still portion of a video segment
- Moving region: a moving portion of a video segment
- Shape descriptors
    - Region shape
    - Contour shape
    - 3D shape
- Dominant colour: most suitable for representing local (object or image region) features where a small number of colours are enough to characterize the colour information in the region of interest.
- Colour layout: represents the spatial distribution of colour of visual signals. It provides image-to-image matching as well as ultra high-speed sequence-to-sequence matching, which requires many repetitions of similarity calculations.
- Texture descriptors
- Face recognition descriptor: can be used to retrieve face images which match a query face image. The descriptor represents the projection of a face vector onto a set of basis vectors which span the space of possible face vectors.
- Motion trajectory: is a simple, high level feature, defined as the localization, in time and space, of one representative point of an object. It describes a list of key points (x,y,z,t) along with a set of optional interpolating functions that describe the path of the object between key points, in terms of acceleration. Its key properties are:
    - independent of the spatio-temporal resolution of the content (e.g., 24 Hz, 30 Hz, 50 Hz, CIF, SIF, SD, HD, etc.), i.e. if the content is available in multiple formats simultaneously, only one set of descriptors is needed to describe an object trajectory in all instances of that content.
    - compact and scalable. Instead of storing object coordinates for each frame, the granularity of the descriptor is chosen through the number of key points used for each time interval. Besides, interpolating function-data may be discarded, as key point-data are already a trajectory description.
- Parametric motion: is associated with arbitrary (foreground or background) objects, defined as regions (group of pixels) in the image over a specified time interval. In this way, the object motion is captured in a compact manner as a set of a few parameters. Such an approach leads to a very efficient description of several types of motions, including simple translations, rotations and zoomings, or more complex motions such as combinations of the above-mentioned elementary motions.

[3] http://tech.ebu.ch/groups/pscaie

- Defining appropriate similarity measures between motion models is mandatory for effective motion-based object retrieval. It is also necessary for supporting both low level queries, useful in query by example scenarios, and high level queries such as "search for objects approaching the camera", or for "objects describing a rotational motion", or "search for objects translating left", etc.
- Spatio-temporal locator: describes spatio-temporal regions in a video sequence, such as moving object regions, and provides localisation functionality. This descriptor enables localization of regions within images or frames by specifying them with a brief and scalable representation of a box or a polygon.

High level metadata

These metadata include all descriptions that link one or more streams to a specific FascinatE service. For example, to group all streams that compose a layered scene, the following descriptors will be used:

- Creation and production: describe author-generated information about the generation/production process of the AV content.
- Variation descriptor: describes variations of the AV content, such as compressed or low-resolution versions, summaries, different languages, and different modalities, such as audio, video, image, text, and so forth. One of the targeted functionalities of the Variation DS is to allow a server or proxy to select the most suitable variation of the AV content for delivery according to the capabilities of terminal devices, network conditions, or user preferences.
- The variation descriptor indicates the type of variation, such as summary, abstract, extract, modality translation, language translation, colour reduction, spatial reduction, rate reduction, compression, and so forth.
- Collection descriptor: describes collections of audio-visual content or pieces of audio-visual material such as temporal segments of video. The Collection Structure DS groups the audio-visual content, segments, events, or objects into collection clusters and specifies properties that are common to the elements. It describes also statistics and models of the attribute values of the elements, such as a mean colour histogram for a collection of images.
- Summarization description: describes multiple summaries of the same AV content, such as to provide different levels of detail or highlight specific features, objects, events, or semantics. By including links to the AV content in the summaries, it is possible to generate and store multiple summaries without storing multiple versions of the summary AV content. It can also be organized into a hierarchical structure, in order to describe different levels of temporal detail.
- Semantic description: describe specific types of semantic entities, such as narrative worlds, objects, agent objects, events, places, and time. It includes:
    - Object description: describes an entity that exists, i.e. has temporal and spatial extent, in a narrative world
    - Agent description: describes a person, an organisation, a group of people, or personalised objects
    - Event description: describes an event, which is a dynamic relation involving one or more objects and agents occurring in a region in time and space of a narrative world
    - Concept description: describes a semantic entity that cannot be described as a generalization or abstraction of a specific object, event, time place, or state. It is expressed as a property or collection of properties (e.g. "harmony" or "ripeness").

ViPER

The Video Performance Evaluation Resource (ViPER) [Doermann, 2000] is a framework for the evaluation of video analysis tools. The ViPER XML format [Viper, 2003] has been defined based on XML Schema as an exchange format for manually produced *ground truth* and results from automatic analysis tools. A ViPER XML description consists of two main parts: configuration information (such as types of events, information about the setup and capture) and the actual data. The data is organized by source files. A list of events is described for each source file, with the time span, spatial attributes and additional properties such as detection confidence for each of the events.

The format is sufficiently flexible to support media types other than video; however, the time representation is frame oriented which makes it difficult to combine sensor data sampled at different rates. The format is source file oriented; properties of sensor and detectors have to be associated with a source file or a set of them. The concept of source file is flexible and in a recent version a virtual file can be defined as a sequence of separate files. Descriptions on different abstraction layers are difficult due to the file orientation and would require some workarounds with application defined semantics. The format is highly flexible for defining new types of objects, with a set of attributes (using the basic ViPER data types) and having static and dynamic values for these attributes. Such definitions can be made in any description instance. Thus the semantics and units of the attribute values are not formally specified. Object instances are not defined in advance; however, all occurrences of an instance in the current video can be collected in one XML element. Identifying object instances across several documents requires application logic. In the current set of basic data types only image but no world coordinates are supported, which is a problem for multi-sensor systems. Trajectories of arbitrary shapes can be modelled.

CVML

The CVML language (Computer Vision Markup Language) was presented at IPCR 2004 [List, 2004]. The language defines a common data interface specifically designed for computer vision. Its focus is on content description of video and image sequence data. The structure of CVML allows only the definition of one dataset or sequence per XML file. CVML does not differentiate between sensors and dataset and it does not facilitate the description of (physical) media locations. Further the concept of non-visual (e.g. audio) features is not supported at all.

Although CVML comes with open-source cross-platform libraries, such as the C++ based CoreLibrary which supports reading and writing the information in XML, the documentation of the large amount of available language tags is weak and incomplete.

DMS-1

The SMPTE Descriptive Metadata Scheme 1 (DMS-1, formerly known as Geneva Scheme) [DMS1, 2004] uses metadata sets defined in the SMPTE Metadata Dictionary. Metadata sets are organised in descriptive metadata (DM) frameworks. DMS-1 defines three DM frameworks, which correspond to different granularities of description: production (entire media item), clip (continuous AV essence part) and scene (narratively or dramatically coherent unit). When DMS-1 descriptions are embedded into MXF files they are represented in KLV (key, length, value) format, but there exists also a serialised format based on XML Schema. However, the format lacks capabilities for low-level information and object trajectories.

RDF/OWL

Representation formats from the Semantic Web domain based on RDF and OWL [Dean, 2004] have become increasingly popular in recent years. The main advantages are well defined semantics and that inference tools can be directly applied to the annotations. This representation is applicable to abstracted higher level information, such as classifications. However, the representation is not suitable for low-level information such as trajectories and low-level features, as descriptions become unreasonably large to handle and computationally expensive to process.

4.3 Domain/Scene Knowledge

4.3.1 Definition

Knowledge about the content domain or type of event, static and dynamic external metadata about the event and static description of the scene setup.

For different use cases and scenarios, basic and additional knowledge about the domain and the scene needs to be made available for different modules in the FascinatE processing chain. An example scenario might be the provision of additional information on players in a football game, game statistics and so on which can be requested from the user. This additional knowledge can be classified in three groups:

- General knowledge about the domain of the production (e.g. structure of a football game, relevant types of events)
- Scene setup (e.g. area of field, stage, audience, reverberation time and limits of acoustic space).
- Dynamic event information (e.g. sports scores, statistics - of the current or related simultaneous events; music concert playlist information etc.). Beyond text, icons/graphics/images could be necessary to convey the information. Could be displayed automatically or at user request.

4.3.2 Requirements

As the knowledge of the domain and scene will be of quite diverse nature (text, tables, figures, diagrams, curves, ...), a highly flexible structure of metadata is required. This structure must allow different types in order to provide the desired information in the most effective way. Besides the information itself, layout descriptions might be included.

In case that a laser scan of the scene is performed, the following metadata need to be represented:

- Reference to the file containing the merged point cloud
- Positions of the scans and of the calibration targets

4.3.3 Candidate Formats

As to the best of our knowledge no format comprehensive enough to cover all our diverse requirements is available, it is likely that an OWL/OWL2 ontology model will be developed to model the knowledge about the domain and scene. Initially, a simple agreed XML structure can be good enough to express setup properties and domain dependent behaviour. Even simpler, the configuration could be persisted and accessed as a key/value dictionary file with or without hierarchy, or a NoSQL database.

The following two formats might be useful for representing scene layout information, one very specific aspect among the broad range of domain and scene related metadata.

MPEG-4 LASeR

[MPEG-4 LASeR] is a standard for rich media scene description on mobile and embedded devices. In our work it is used to link additional content to the main video stream which can be simple text, HTML pages, pictures, audio and video clips. It is based on SVG Tiny 1.2. SVG (Scalable Vector Graphics) is a W3C recommendation for the XML based representation of 2D vector graphics.

X3D

X3D is an ISO standard [X3D, 2008] that provides an XML-based file format for representing 3D computer graphics. The X3D specification includes various internal and external APIs and a full runtime, event and behaviour model. It is therefore much more than a simple exchange format. A subset of X3D is XMT-A, a variant of XMT, defined in MPEG-4 Part 11. It was designed to provide a link between X3D and 3D content in MPEG-4 -BIFS. BIFS vii is a binary format for two- or three-dimensional audiovisual content. It is based on X3D and Part 11 of the MPEG-4 standard. BIFS is a MPEG-4 scene description protocol for composing MPEG-4 objects, describing interactions with MPEG-4 objects and for animating 2D and 3D MPEG-4 objects.

4.4 Production Rules and Visual Grammar

4.4.1 Definition

General rules for view selection and virtual camera work, organisation/event specific rules for content selection.

There are a number of general principles that cameramen are trained to use when framing a shot, and that a programme director will use when positioning the cameras, instructing the cameramen and selecting shots. Some of these are common across many programme genres, whilst others are specific

to particular kinds of programmes, or may vary depending on the screen size and shape of the target viewing terminal. Furthermore, some of these principles will be applied more flexibly by some directors and cameramen than others, as each tends to have his/her own style. The FascinatE scripting system needs to capture these general principles so that the kinds of shots offered by production-side scripts adhere to the expected *look and feel* of a professional TV production.

4.4.2 Requirements

Examples of the kinds of requirements on the script generation that the production rules and visual grammar should take account of include the following:

- Shot framing: There are guiding principles that control the way in which a shot tends to be framed. For example:
    - A close-up shot of a single person tends to have their eyes roughly one third of the way down the image
    - A shot following a person moving forwards tends to have a point a little way in front of the person at the centre of the image, giving so-called *looking room* so that there is more empty image in front of the person than behind.
    - The camera will not rigidly follow a moving person or group of people, but will move smoothly. If a person is moving erratically, the shot will tend to be framed wider so that they stay in shot without the camera having to make rapid movements.
- Shot selection: The rules of thumb for cutting between cameras can vary according to the type of production, but examples include
    - Avoiding cutting between a tight and wide shot from a camera at the same position
    - Avoiding *crossing the line*, i.e. cutting from a camera on one side of the action to one on the other side. Indeed, cameras tend to be placed on one side on an imaginary line between the action and the audience; where a *reverse angle* shot is needed (for example to show a close-up of sporting action that is only clearly visible from this angle), the fact that it is a reverse angle shot is often indicated using an on-screen caption.
    - Avoiding rapid cutting between cameras: this is particularly important for productions that aim to give a more immersive feel (large screen or stereoscopic, for example). Generally, the larger or more immersive the display, the less cutting and camera movement should be used.
- Shot framing and selection rules may partly depend on the domain (e.g. type of sports).

4.4.3 Candidate Formats

There is no standard format for expressing these rules at present, and it is likely that the project will develop its own rule-based shot selection algorithms, with the parameters for each shot type being expressed in terms of things like parameters to control a filter that links the location of the target subject to the centre of the shot. The development of the format will strongly be influenced by the choice of an ECA (event - condition - action) rule processing engine. Efficient evaluation of complex structures of rules and principles is a key requirement for the quasi real-time decision making task of the Scripting Engine.

4.5 Script Templates

4.5.1 Definition

Templates of rendering and delivery scripts to be adapted to generate the actual scripts.

The Rendering Scripting Engines (SEs) will emit decisions stating which content/views will actually be displayed at a certain time, while the Delivery Scripting Engines will, roughly spoken, prepare individual video streams for subsequent screen composition. See Figure 4 for an overview. The information that represents those decisions/options are called *scripts*. Scripts are transmitted along the workflow chain

and in general state how the AV content should be composited/rendered on screen. Because there are multiple SEs in the chain, not all decisions will be fixed in the primary Production SE. The SEs composes scripts based on templates, which are prepared and fine-tuned (decision between options) for certain situations. There will be templates for many different situations (ideally all the interesting situations that the SEs should react to) which define the desired camera/view selection behaviour. Depending on the context, the SEs will choose between and/or assemble the most appropriate scrip templates and generate the actual script(s) based on them. The context, i.e. all knowledge available about the scene, will eventually allow an informed decision.

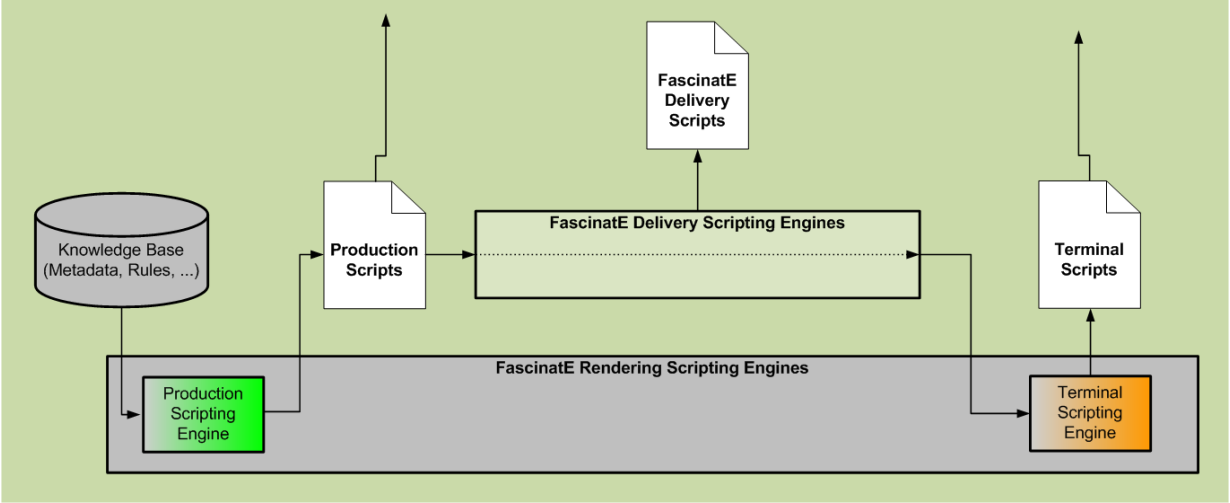


Figure 4: Scripting Engines & Scripts

4.5.2 Requirements

Depending on the specific context in which a script will be used, predefined templates of scripts will be instantiated. Templates are created for certain situations that are differentiated within FascinatE events and will include e.g. story idea, style or structure elements.

To be able to define script templates, we need to know what kinds of script are considered within FascinatE production. Basic script distinction that can be made is on visual, audio and audio visual (AV) scripts.

- *Visual scripts*: Include different types of scripts based on their visual distinction.
  - Framing covering overview view or ROI (*Region-Of-Interest*) view,
  - Entities covering different objects such as player/singer, team, referee, conductor etc.,
  - Location covering various physical areas of the event such as a goal, centre of stage etc., and
  - Activities covering incidents e.g. tackles or fouls.
- *Audio*: Include different types of scripts based on their audio distinction.
  - Entities like commentator which can be audible or off,
  - Ambience sound, and
  - Audio tracking of various incidents e.g. tackles, or the crowd cheering for a basket.
- *AV*: Include different types of scripts based on the combination of audio and video. Example is replays

Additional types of scripts that need to be investigated and which are not clearly separable as audio or visual are producer generated scripts, as well as (advanced) scripts generated from end user settings and profiles which other (non-advanced and more laid-back) users would follow (also named *follow friend* scripts).

Further, adaptation of content and accessibility for impaired people needs to be added. It can be considered as a special script type and will be addressed through interface design and specific *information presentation***.** Different insertions might be additionally investigated and observed as a specific script types (advertisement insertion scripts). Examples are overlays, banners or temporal inserts.

4.5.3 Candidate Formats

In this section, different formats for describing script templates will be presented.

NSL

The Narrative Structure Language (NSL) [Ursu, 2008; Ursu, 2010] has been developed by a team at the Goldsmith's College, University of London, for several years. Major contributions were developed in the FP6 project *NM2 - New Millennium, New Media*[4] and the ongoing FP7 project *TA2 - Together Anywhere, Together Anytime*[5]. The Narrative and Interactive Media (NIM) group[6] researches computational and technological underpinnings of interactive mediated narrativity, particularly by moving-image, considered in the whole spectrum: from the artistic end, as a means of creative expression, to the pragmatic end of informal communication between people separated by space and time. NIM created NSL, the first declarative language for the representation of interactive time-based media narratives, and was a key developer of the ShapeShifting Media Technology[7], founded in NSL.

A detailed specification of the NSL format is not yet available publicly, but in the following non-public deliverables:

- NM2 D5.9 ("the old NSL")
- TA2 D7.4 ("the new NSL")

The "new NSL" will be OWL-based and is currently developed in the TA2 project. Further, the NIM team is currently implementing reasoners for run-time instantiation of the structures with actual media. The group is further planning to release a public NSL specification by the end of the project.

In detail, the format allows to abstractly define the characteristics of content that can be played ("fit") in a certain situation. Those characteristics are based on a semantic vocabulary that, in turn, the content has to be associated with. NSL was designed based on the assumption that there is extensive knowledge (semantic tagging) about what is happening in each of the video clips in the repository, which the structure elements can refer to. While that assumption works well for offline scenarios, it has never been evaluated how far this approach can be applied in real-time situations.

A major difference that is specific to FascinatE is the exceptionally high resolution of visual content. Therefore, not (only) the selection of the camera stream is the central task of the Scripting Engine, but much more the selection of a proper cropping area.

There are several types of graphical structures which may be organized in multiple layers. Figure 5 shows a very simple graph example. The storyline starts with a decision point (round element, user has to take a decision) and then plays one of multiple options according to the directed graph, or a sequence of 2 clips in the lowest case.

[4] http://www.ist-nm2.org

[5] http://ta2-project.eu

[6] http://nim.goldsmiths.ac.uk

7 http://www.shapeshift.tv

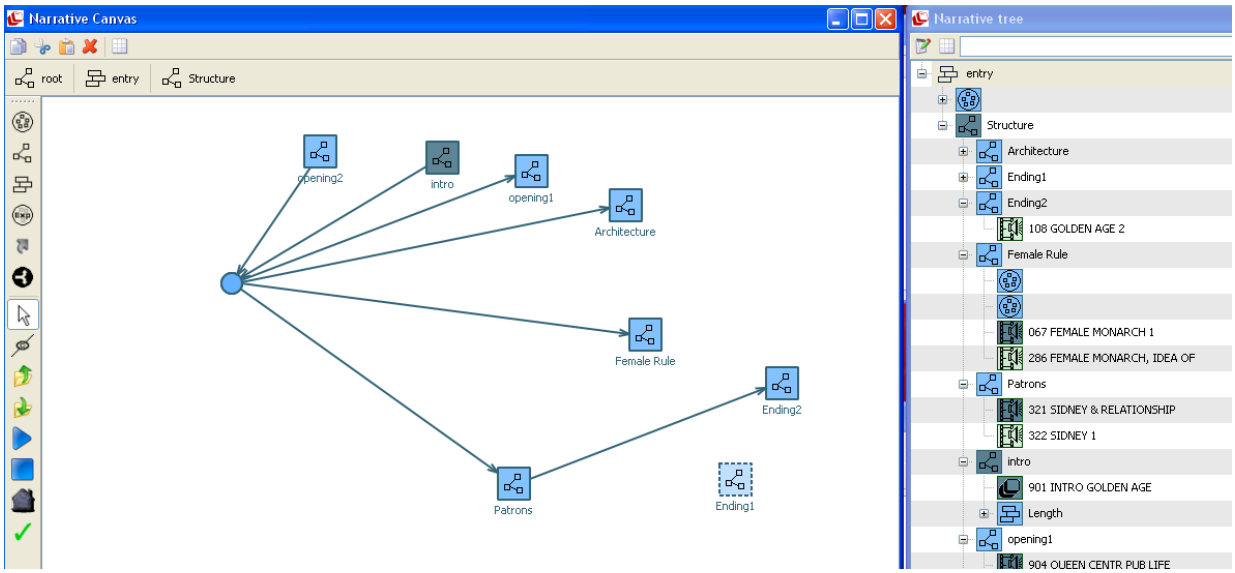


Figure 5: User interface for creating Narrative Structures

Real graphs are of course much more complex and consist of hundreds of nodes for a typical production. The most relevant node type is the aforementioned decision point (see Figure 6), which states that the storyline will continue on only one of multiple routes. The content building blocks can be divided into two groups: references to a specific clip (fixed), and abstract structures which describe the characteristics of a clip that may fit into the story at that time. In the latter case, the abstract description is referring to the semantic annotations that are linked to every video in the repository. At runtime, a clip matching the (possibly complex/semantically expressed) criteria is identified, which eventually enables the system to produce different videos e.g. for every user, taking some sort of context as a configuration into account. Default behaviour is also specified for cases where either no suitable clip is available or multiple clips match the criteria equally. Further, there are loop structures which may state that the system will continue playing a certain video clip or random videos matching certain criteria until a condition is met. In general, audio is considered as an attachment to a video and is not reasoned with, other than simple effects like e.g. audio overlays for background music. For a comprehensive list of structure elements, the interested reader may look for the documents mentioned in this section or contact the authors of the non-public deliverables.

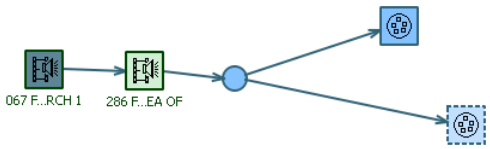


Figure 6: NSL Decision Point

The nodes of an NSL structure may define the characteristics of a storyline along multiple dimensions. Examples are:

- cinematographic principles: cutting certain views after each other or too fast may disrupt the viewer
- aesthetic rules: e.g. define that in a scene with a certain colour scheme (indoor) content from a totally different setting (beach) cannot be selected
- semantic story coherence: a simple example would be the occurrence of an object that is damaged within the storyline; as soon as the object has been shown in damaged state, clips showing that object undamaged can no longer be selected

- narrative arc: certain content types have basic narrative structures that help convey the story, e.g. there should be an introduction, a main plot with a climax and an ending (cp. Freytag's pyramid[8])

Isis / Cabbage / Viper

Isis[9] is a programming language for responsive media that was created in 1995 at the MIT Media Lab by Stefan Agamanolis[10]. He continued development at Media Lab Europe in the context of the Human Connectedness research group. His thesis was titled "Isis, Cabbage, and Viper: New tools and strategies for designing responsive media".

Responsive media are media that have the ability to sense and react intelligently to factors like presentation equipment or conditions, audience identity or profile, direct interaction, history or anticipation of involvement, and so on. Unlike the classic media model, responsive media systems consist of a two-way channel between the audience or participants and the person or device generating or controlling the delivery of the media. This channel, whether physical or conceptual in nature, allows important data gathered at the delivery points to be returned and used to alter the presentation in a suitable way based on the goals of the experience [Agamanolis, 2001].

According to the ISIS website[11], "The core of the Isis language can be built on any platform that has a standard C compiler. However, because interfaces for audio, video, and graphics vary greatly between operating systems, and because we have limited staff, we have only been able to create full implementations of Isis on a few platforms. At the moment, the Linux operating system is the preferred platform for running Isis."

Cabbage, an experimental visual tool, motivated in part by some of the shortcomings of Isis, that employs a purist form of case based reasoning in a system for creating responsive graphical layouts. (...) Viper[12], a tool for making video programs that can re-edit themselves during playback to adapt to different viewing situations and real-time audience activity, whose design is informed by the lessons learned from building the previous two tools. These tools all strive to provide simple, elegant interfaces that enable a creator, at one level, to build complex and meaningful input-output relationships for controlling media behaviours, but at another, to programmatically express a “philosophy of design” that travels along with the media objects and dictates how they are presented in response to whatever factors are of interest. A host of prototype applications, including hyperlinked video dramas, telepresence environments, responsive television advertisements, ambient information displays, and interactive artworks serve to test these tools and to support the validity of the three basic observations about responsive media that form the basis of their design [Agamanolis, 2001].

4.6 Scripts

4.6.1 Definition

Scripts contain statements to control renderers or delivery mechanisms.

In the FascinatE system, we envision at least two levels of scripts (note Figure 4 in the previous section for an overview):

1. At a semantic level, *FascinatE Rendering Scripts (FRS)* accompany the *Layered Scene Representation* (LSR) and consist of a complete (including metadata but not final w.r.t. decisions) script definition to be used by a FascinatE Rendering Node - possibly located in the terminal, or at other points in the FascinatE system. Scripts vary w.r.t. the degree of freedom for the consumer, i.e. they may be still generic (leaving options open) or they could already be a precise information for the

[8] e.g. http://oak.cats.ohiou.edu/~hartleyg/250/freytag.html

[9] http://web.media.mit.edu/~stefan/isis

[10] http://www.agamanolis.com

[11] http://web.media.mit.edu/~stefan/isis/getting-started.html

[12] The attentive reader may notice that another tool named ViPER was mentioned in section 4.2.3. While the tools share the same name, they have nothing in common beyond that.

renderer about what and when to select from the LSR. Besides selection decisions and/or options, the metadata necessary for taking remaining decisions has to be passed on down the FascinatE workflow chain. The options available in the scripts will generally be reduced the further they travel from the production-end towards the terminal by a cascade of Scripting Engines. In particular, we can focus on the Scripting Engines at these two end points of the chain:

- *Production Scripts* specify particular views of the scene (e.g. wide view of a group of people, or close-ups of particular people), and rules for how these views can be adapted to different kinds of terminal and/or by user input. These are derived by a *Production Scripting Engine* from a combination of professional user input (e.g. selecting regions of interest, with one being specified as the default view), (semi-)automated detection and tracking for regions of interest, metadata (e.g. shot framing from manned broadcast cameras), and shot framing and production rules (e.g. how loosely or tightly a view should follow a region-of-interest).
- At the other end of the chain, the *Terminal Scripting Engine* takes the views specified by the production scripts (which may specify general regions of interest rather than precise framing, may have a specified amount of flexibility in the exact framing), and generates *Terminal Scripts* which based on user input and device characteristics precisely specifying the content to render. Note that, in some scenarios, there may only be one Scripting Engine which is creating terminal scripts already in the first pass.

In the simplest case, there may be only one Production SE in the chain. In a more likely architecture, there might be a single Production SE that computes for various devices in parallel, and a series of further SEs down the chain taking final decisions. Note that Production Scripts and Terminal Scripts, as well as scripts at any other intermediate stage, have similar syntax requirements and should therefore be based on the same format. The nominal difference between Production and Terminal SEs is rather motivated by the system design than in requirements.

2. At the AV delivery level, *FascinatE Delivery Scripts (FDS)* are produced to steer the delivery mechanisms in the subsequent network elements. The reason to introduce this extra level of scripting functionality is the following. When the layered scene and the accompanying rendering scripts (e.g. production scripts) are ingested in the network and prepared for delivery, the layered scene representation may go through several processing stages. These can involve rendering one or more AV streams (e.g. according to the Production Scripts), segmenting and coding the rendered scene or the individual layers (e.g. spatially into tiles or even temporally). In such a case, the Delivery Scripts can be used to specify how the different coded streams or segments relate to the layers of the layer scene description and which ones should be delivered to the terminal. The delivery scripts reflect in particular the mapping between the layered scene description and the actual streams/segments created by the segmentation process.

4.6.2 Requirements

Since both rendering and delivery scripts contain information that must be time-aligned with the AV content, a general requirement for scripts (both FRS and FDS) is that they need to be “streamable”, i.e. the chosen syntax to represent them must have functionality to time-stamp the scripted information and allow the scripts themselves to be generated and pushed for delivery on-the-fly.

Requirements for FascinatE Rendering Scripts

The two kinds of rendering scripts introduced above, Production Scripts and Terminal Scripts, are almost identical w.r.t. the following requirements discussion even though they are used in different parts of the FascinatE system architecture. All scripts generated before the terminal may leave options open for subsequent decisions down the chain. Therefore, all scripts have to contain the metadata necessary for subsequent SEs to do so. This metadata might decrease (filtering) to the actual amount necessary downwards the chain.

However, there is a limit to the scope of the Rendering SEs, as e.g. optimizing dynamic video transmission bitrates (video quality) is out of concern and is taken care of by either the Delivery Scripting Engine or the network components. That boundary should also simplify the number of communication streams between the components in the architecture. In fact, the SE is one of the most central components w.r.t. the knowledge it has to receive from other components in order to do its job and therefore, limiting the complexity in the design phase wherever appropriate is a key principle.

To explain the issue, a simple example would be a production rule that states that for two people within a certain distance, close-up shots should not be shown directly after each other. Instead, going through a wide/landscape shot in between can allow viewers to grasp the geometry of the whole scene and the spatial arrangement of relevant actors in them. Thereby, the viewing experience can be improved. In order to account for that rule, the Production SE must be informed of final decisions on which view is actually shown to a particular terminal/user.

Requirements for FascinatE Delivery Scripts

As described above, in the FascinatE Rendering Scripts, the definitions will rather refer directly to the raw AV data present in the LSR. For delivery purposes though, the set of AV data to be transported may need to be adapted to be processable by the delivery mechanism. Therefore, in the course of the delivery of the layered scene, several processing stages can potentially influence the way the layered scene is represented. Available layers can be represented in different ways, e.g. depending on choices about tiling and coding. In addition, based on the production scripts, additional layers or streams may be added next to the original layers themselves. For example, the information pertaining to the location of Region of Interests (ROI) can be used upfront at the delivery ingest to create new streams making these ROI videos independently accessible. The essential roles of delivery scripts are therefore:

a. first, to signal the mapping between the raw layers of the LSR (as used by the rendering scripts) and the actual way the AV data is represented, segmented (in ROI, tiles…) and coded in various streams/segments. This way, at any point in the delivery chain, a FascinatE component, such as a FascinatE Rendering Node, is able to access the right portions of the layered scene and process them according to the rendering scripts.

b. to expose to the network elements the required information so that they can process the data accordingly (filtering, re-assembly, forwarding, caching…)

Depending on the actual use cases and scenarios (as defined in D1.1.1), delivery scripts should instruct whether:

- Layers are encoded as is
- Layers are encoded at various quality levels, resolutions, in tiles…
- New streams are created, e.g. ROIs
- Streams can be started and ended (e.g. an ROI may be short-lived). This can result in a number of streams that vary over time
- Switching points at pre-defined time boundaries in each stream allows network elements to re-assemble streams in a personalized way.

4.6.3 Candidate Formats

Scripts and metadata will influence the way multimedia content will be composed, personalized and distributed. The description can happen at various levels. A possible classification of these levels is the following:

1. *Resource level*: This refers to isolated and homogeneous multimedia content that can be consumed alone or in conjunction with other multimedia content. Resource level applies to single media content where a standardization body defines the whole format of a single media modality (e.g. .jpg or .mp3), or at a multimedia level (several modalities e.g. .mpg or .html)
2. *System level*: This level describes groups of different resources (such as media, multimedia or even entire scenes) gathered together forming high-level structures and concepts (e.g. MPEG-21 DIs [MPEG-21 P2, 2005] or NewsML [IPTC, 2007]). This level can also comprise metadata about the resources (e.g. MPEG-7 Media Information [MPEG-7 P5, 2003] or semantic information).
3. *Scene level*: Finally, this level describes a whole multimedia presentation. This is an extension of the system level with:

   a) layout information: cues or complete rendering description, (e.g. HTML), and

b) synchronisation information: synchronisation of the different elements of the scene and of the content along the timeline (e.g. SMIL, MPEG-4 BIFS or Adobe Flash).

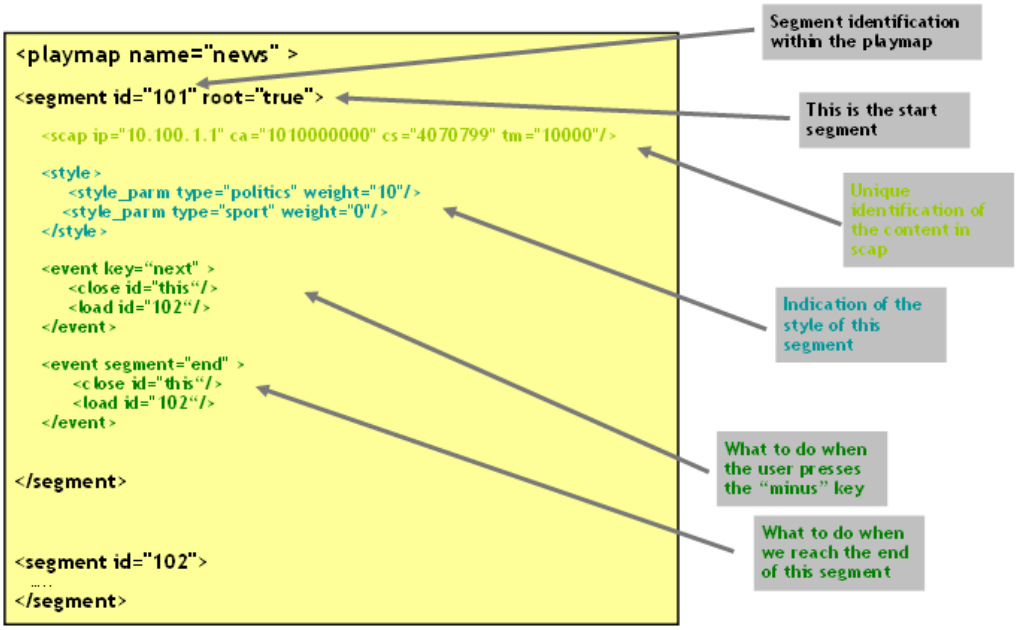
More specifically, scene level should be driven by the FascinatE Rendering Scripts while resource and system levels should be described and adapted by FascinatE Delivery Scripts.

We now describe some candidate formats that may apply to the rendering scripting requirements.

RUBENS play-map

The RUBENS play-map is an XML description that allows for a personalized assembly of media segments [RUBENS, 2010]. A segment is defined as a self-supporting atom of media that can be efficiently handled or even cached by the network. This definition implies that segments are encoded so that they start with an IDR frame and are closed GOP. The RUBENS play-map is basically a sequence of segment descriptors. Each segment descriptor may contain:

1. Transport information, i.e. some form of URL where segment content can be retrieved
2. Descriptive information (duration, names of actors, etc.) that can be used to personalize the playout
3. Allowed user-interactions and an implicit action on what to do when the current segments ends
4. References to other play-maps


<playmap name="news" >

<scap ip="10.100.1.1" ca="1010000000" cs="4070799" tm="10000"/>
<style>
<style_parm type="politics" weight="10"/>
<style_parm type="sport" weight="0"/>
</style>
<event key="next" >
<close id="this"/>
<load id="102"/>
</event>
<event segment="end" >
<close id="this"/>
<load id="102"/>
</event>

Segment identification within the playmap
This is the start segment
Unique identification of the content in scap
Indication of the style of this segment
What to do when the user presses the "minus" key
What to do when we reach the end of this segment

Figure 7: Example of RUBENS play map.

In this format, the RUBENS play-map is a single piece of metadata, containing all information needed for a personalized experience. The format is well suited for on-demand content.

I2Vision play-map

The I2Vision [I2Vision, 2010] play-map is an ALU in-house extension of the RUBENS play-map. It is still targeted to on-demand content, but transport information has been removed from it. This allows for a more generic description that only relates to the content. Depending on the target delivery platform, a separate descriptor is created (platform-map) that contains the mapping of the generic segment identifier used in the play-map, to a format that is known by the delivery platform. As an example, the platform-map for a Microsoft MediaRoom IPTV delivery platform would map the generic segment identifier to the GUID that MediaRoom assigned to the ingested file that corresponds to this segment.

At the renderer side, both play-map and platform-map are needed in order to retrieve the actual (segmented) content. This is illustrated in the next Figure 8.

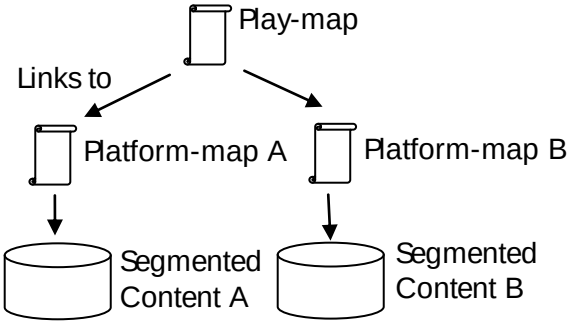


Figure 8: Play map & platform-map

The second point where the I2Vision play-map differs from the RUBENS play-map is that the allowed user-interactions do not always need to be specified per segment. Instead, user-interactions could be specified globally (valid for all segments) or on a package basis (i.e. a logical grouping of segments).

MPEG-7 and MPEG-21 for Content Adaptation

Among MPEG standards, MPEG-7 [Manjunath, 2002] focuses on the description of multimedia content, whilst MPEG-21 [Burnett, 2006] rather focuses on a complete set of tools enabling end-to-end solutions in multimedia systems. Within this framework, multimedia elements are represented using DIs (Digital Items) [MPEG-21 P2, 2005]. According to this framework, a DI can convey resources (media files) and descriptors (metadata). The DI is a generic container capable of representing a great variety of multimedia. MPEG-21 Part 7 DIA (Digital Item Adaptation) [MPEG-21 P7, 2004] standardizes a group of description tools capable of collecting the information that an adaptation engine can draw on to drive the adaptation of those DIs. One group of these description tools is the Usage Environment Description Tools (UED Tools), which are devoted to the description of the terminal, network, user characteristics and preferences, and natural environment.

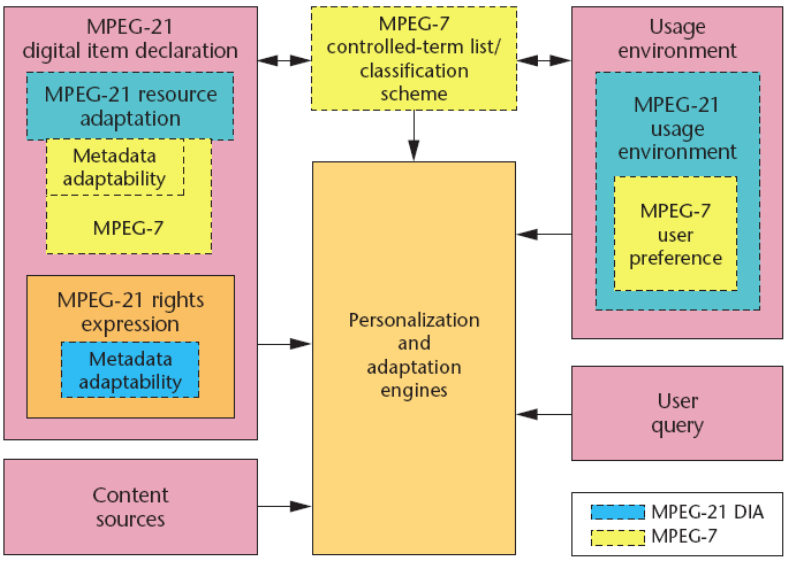


Figure 9: Block diagram of digital item adaptation using MPEG-7 and MPEG-21 [Tseng, 2004]

Some implementations of an efficient matching of the content description (according to MPEG-7 Part 5) with the context description (according to UED Tools in MPEG-21 Part 7) can be found in the CAIN framework [Martinez, 2005].

CAIN

CAIN (Content Adaptation INtegrator) is a metadata driven [van Beek, 2003] and extensible adaptation engine [Valdes, 2006] that integrates different Content Adaptation Tools (CATs), each of them with different adaptation capabilities. These CATs implement both, several adaptation levels (signal, system, and semantic [Magalhaes, 2004; Smith, 2003]) and several adaptation modalities (transcoding, transmoding, scalable content, temporal summarization, etc.). Also proposed by [Magalhaes, 2004], the CAIN DM takes into account both content and usage environment descriptions in order to make a decision about which CAT (and parameters) to select.

Nowadays a great deal of effort is being put into developing effective techniques to automatically adapt multimedia content to the usage environment. Roughly we can divide the automatic adaptation problem into two parts: first, the decision of which adaptation to perform and second, the execution of the selected adaptation.

However the proposed techniques in the literature widely vary in nature, features under consideration and modus operandi.

Certain techniques take into account a description of the environment [Martinez, 2005; Lopez, 2007], and model the adaptation decision problem making use of techniques like constraints satisfaction [Martinez, 2005] or automatic planning [Lopez, 2007]. Other techniques are concentrated on analyzing the resource to measure the utility or the quality of the adaptation [Jannach, 2006]. Usually there exists a distinction between mandatory constraints and user's preferences [Kohncke, 2007].

Several approaches have been proposed to perform content adaptation. In [Jannach, 2006] the authors propose a knowledge-based approach to decide the adaptations to perform. In [Wang, 2003] the authors propose to score feasible adaptations using an utility function. In [Tseng, 2004] the authors propose a method to summarize video with optimal coherent semantic segments within certain time constraints (cf. Figure 9). In [Lopez, 2006], authors propose modelling the decision phase, to select both the adaptation tool and the parameters, as a Constraints Satisfaction Problem (CSP) followed by an optimisation phase. Although this model is sound, it presents an important computational cost.

In [Lopez, 2006], some important characteristics of the nature of the adaptation problem are taken into account in order to remodel the CSP phase as a Constraints Matching Problem (CMP). As a consequence, the computational cost of the decision phase is significantly improved.

Most of the authors have made use of MPEG-21 Part 7 UED (Usage Environment Description) [Burnett, 2006] to represent the features of the usage environment.

SMIL[13]

SMIL (Synchronized Multimedia Integration Language) [Bulterman, 2008] is an established multimedia container format supported by W3C, the World Wide Web Consortium. SMIL is an XML format, developed since 1996. Unlike Flash/Flex and Silverlight, SMIL is a declarative format. This allows presentations (and presentation fragments) to be created dynamically and to be adapted locally. SMIL's timing and animation support are at the core of Scalable Vector Graphics (SVG's) temporal capabilities. SMIL also forms the core of Microsoft's HTML+Time engine. The MMS multimedia messaging service in 3GPP-compatible mobile telephones also use SMIL as the basis for message construction.

The current release of SMIL is version 3.0[14], which was confirmed by W3C in December 2008. SMIL supports a profile model, in which groups of language features may be tailored for use in special environments. Such profiles currently exist for simple playlists, mobile telephones, adaptive books for the blind, set-top boxes and full desktop media players.

W3C supports and maintains the SMIL language, but it does not provide a reference player. Several implementations of SMIL exist, in varying degree of openness. The CWI Ambulant player[15] is a full open source implementation of SMIL, based on L-GPL licensing.

---

[13] This description is based on a result of the TA2 project (http://www.ta2-project.eu/) as reported in D5.1 [Bulterman, 2009]. The FascinatE consortium thanks Centrum Wiskunde & Informatica (CWI, Amsterdam, http://www.cwi.nl/) for the permission and insight into current SMIL developments.

[14] http://www.w3.org/TR/SMIL/

[15] http://www.ambulantplayer.org/

SMIL is an integration format. As such, it does not directly define media objects (with the exception of timed text content). Instead, SMIL acts as a container format in which spatial, temporal, linking and interactive activation primitives can be used to place, schedule and control a wide assortment of media objects. In this sense, SMIL fits well with FascinatE requirements, in particular:

- *Rendering component support*: SMIL allows text, image, audio, video and graphics objects to be defined and integration content. The Ambulant SMIL player uses an open architecture to provide support for both platform specific and platform agnostic media objects. SMIL provides scheduling primitives that allow media to be temporally aligned. It also provides media control primitives to control the pre-fetching of media objects to improve performance. SMIL additionally provides declarative support for media object movement (pan/zoom), media object opacity (chroma key and full opacity support) and for layering hyper-link anchors on arbitrary media objects.
- *Spatial composition*: SMIL provides an explicit layout model, in which objects can be positioned within a 2-1/2D rendering surface (2D + z-order). The layout model allows for the definition of simple regions, nested hierarchical regions and for layout models that span multiple physical rendering windows. SMIL's layout is based on either absolute positioning or on a registration-point model, allowing abstract presentation layout that can be adapted to different screen topologies at the time of rendering.
- *Temporal composition*: SMIL provides a hierarchical temporal composition model from which individual presentation timelines can be generated. SMIL linking and event-based timing structure can be used to modify the strict hierarchical relationships and to support a directed graph timing model. The main temporal structuring elements are the *parallel* <par> and *sequential* <seq> containers, each of which provides a local time base for scheduling media objects or child time containers. SMIL also supports the *exclusive* <excl> time container, which allows for the dynamic composition of media elements based on user interaction or event-based activation. Specific presentation timing can be inherited from constituent media objects, from directed scheduled behaviour within the presentation hierarchy or from (interactive) event behaviour in the presentation. SMIL also allows a limited degree of constraint-based object activation and termination.
- *User interaction*: SMIL supports user interaction via two main constructs. The first is temporally constrained hyperlinking, the second is direct event-based user manipulation of structural begin/end conditions. Both implicit and explicit interaction is supported. The interaction defined within SMIL declarative structure defines the activation/termination moment for presentation fragments. Extensive media-based interaction is relegated to individual media codecs.

The main benefit of SMIL in this context is the declarative nature of the SMIL language. This allows presentations and presentation fragments to be dynamically generated, adapted and modified.

NSL

Like for script templates, the Narrative Structure Language is also a candidate format for scripts. The fact that a script is containing options that might not be replaced with decisions until the very last processing step at the terminal actually underlines the format requirement similarity between script templates and scripts.

An elaborate description of NSL was already presented in the previous section.

4.7 Production Team Annotations

4.7.1 Definition

Annotations by the production team to guide the scripting process.

Video and audio annotation is the operation of associating markers or other graphical elements to the AV content. Annotations can be made by the production team to control the scripting process, e.g. marking a shot for replay. The extent of such annotations can be spatial (coordinates) and/or temporal (time in/out). Besides the direct metadata acquisition, inherent metadata can be inferred by analyzing

user interactions. In real-time (live) setups the role and expected quantity/quality of manual annotations is somewhat limited yet may still contribute to an immersive video consumption experience.

4.7.2 Requirements

It will be necessary to provide a way for the production team to record metadata for use in shot selection and the offering of replays. Examples of the kind of information that should be recorded by the production staff include:

- Regions-of-interest within the scene
- Time codes of the start and end of key events (as would be used to offer replays of important moments during a football match, for example).
- Indication of certain events that just happened.
- References to objects, persons and events from a controlled set for the time spans and regions

The annotation process can be supported by the results of automatic analysis, such as detecting events, objects and persons, i.e. the professional operator validates and amends the automatic results.

Selection of live sequences for replay, as it is currently done in live broadcasts, is done verbally inside the OB studio and relies on the timely and repeated production of sequences by the replay operator. For the FascinatE operator to be able to annotate replay sequences, markers would have to refer to past, near-live events based on some ontology (model of concepts to annotate with, and relations between them). Alternatively, the system would need to be equipped with some of the functionalities of replay systems.

4.7.3 Candidate Formats

Time code is generally used to log the times of incidents, and is supported by various professional format specifications. There is at present no standard for recording regions-of-interest within a large broadcast image in a live situation, although there are elements of standards such as AAF (Advanced Authoring Format) that can specify cropping operations on the image.

MPEG-7 (cf. Section 4.2.3) could be used for describing time segments and regions-of-interest as well as reference objects, persons and events.

The W3C Media Fragments URI (Uniform Resource Identifier) [Troncy, 2010] provides means for specifying fragments of media items such as time ranges or simple regions. However, there is currently no construct to directly describe a region moving over time, this can only be represented as a sequence of regions.

4.8 Rights and Licensing

4.8.1 Definition

Information about rights attached to the content and related licensing information.

A platform for distributing media like FascinatE needs solutions for managing the way users can interact with digital contents: different types of contracts determine what content users are allowed to view (both live and recorded) and which operations they can perform. Therefore metadata are required for the access, delivery, management and protection processes of these different content types in an integrated and harmonized way, to be implemented in a manner that is entirely transparent to the many different users of multimedia services.

We can identify two main actors: the streamed content, from now on denoted Digital Item, and the user. The user will interact with the digital item according to owned rights.

4.8.2 Requirements

In the project we are dealing with complex digital items, made up of different layers of streams and described by metadata. In order to correctly handle such objects, we must be able to:

- uniquely identify digital items and their related resources
- uniquely identify intellectual properties related to the digital items (and parts thereof), for example abstractions
- use identifiers to link digital items with related information such as descriptive metadata
- identify different types if digital items

On the other hand, a user is any entity that interacts or makes use of a digital item and interacts with other users.

Possible interactions must include providing content, aggregating content, distributing content, consuming content and subscribing to content. Moreover we must be able to facilitate and regulate transactions that may occur from any of the above.

All interactions must be carried out according to rights owned by users. Therefore mechanisms are needed in order to protect digital content and honour the rights, conditions, and fees specified for digital contents.

4.8.3 Candidate Formats

MPEG-21

This standard aims at defining a normative open framework for multimedia delivery and consumption for use by all the players in the delivery and consumption chain. It is based on two essential concepts: the definition of a fundamental unit of distribution and transaction (the digital item) and the concept of users interacting with Digital Items (DI). MPEG-21 is therefore supposed to provide all the tools needed to support users to exchange, access, consume, trade and otherwise manipulate digital items in an efficient, transparent and interoperable way.

An MPEG-21 Digital Item can be a complex collection of information. Both still and dynamic media (e.g. images and movies) can be included, as well as digital item information, metadata, layout information, and so on. It can include both textual data (e.g. XML) and binary data (e.g. an MPEG-4 presentation or a still picture).

MPEG-21 provides the concept of Digital Item Declaration (DID), whose aim is to describe a set of abstract terms and concepts to form a useful model for defining digital items. Within this model, a digital item is the digital representation of a work, and as such, it is the thing that is managed, described, exchanged, collected, etc.. It features the following concepts:

- Container: is a structure that allows items and/or containers to be grouped. These groupings of items and/or containers can be used to form logical packages (for transport or exchange) or logical shelves (for organization).
- Item: An item is a grouping of sub-items and/or components that are bound to relevant descriptors. They can be stated as follows: items are declarative representations of digital items.
- Descriptors: contain information about the item, as a representation of a work. Items may contain choices, which allow them to be customised or configured. Items may be conditional (on predicates asserted by selections defined in the choices). An item that contains no sub-items can be considered an entity – a logically indivisible work. An item that does contain sub-items can be considered a compilation – a work composed of potentially independent sub-parts. Items may also contain annotations to their sub-parts.
- Component: is the binding of a resource to all of its relevant descriptors. These descriptors are information related to all or part of the specific resource instance. Such descriptors will typically contain control or structural information about the resource (such as bit rate, character set, start time points or encryption information) but not information describing the content within. A component itself is not an item; components are building blocks of items.
- Condition: describes the enclosing element as being optional, and links it to the selection(s) that affect its inclusion. Multiple predicates within a condition can be combined as an AND/OR relationship.

- Choice: describes a set of related selections that can affect the configuration of an item. The selections within a choice are either exclusive (choose exactly one) or inclusive (choose any number, including all or none).
- Selection: describes a specific decision that will affect one or more conditions somewhere within an item. If the selection is chosen, its predicate becomes true; if it is not chosen, its predicate becomes false; if it is left unresolved, its predicate is undecided.
- Resource: is an individually identifiable asset such as a video or audio clip, an image, or a textual asset. A resource may also potentially be a physical object. All resources must be locatable via an unambiguous address
- Predicate: is an unambiguously identifiable declaration that can be true, false or undecided

The standard User model fits FascinatE needs, featuring tools needed to handle required interactions. To define Rights, MPEG-21 features a Rights Expression Language (REL), a machine-readable language that can declare rights and permissions using the terms as defined in a Rights Data Dictionary. The data model of this language consists of four basic entities and the relationship among those entities. This basic relationship is defined by the assertion grant. Structurally, a grant consists of the following:

- The principal to whom the grant is issued: A principal encapsulates the identification of principals to whom rights are granted
- The right that the grant specifies: A right is the verb that a principal can be granted to exercise against some resource under some condition. Typically, a right specifies an action (or activity) or a class of actions that a principal may perform on or using the associated resource.
- The resource to which the right in the grant applies: A resource is the object to which a principal can be granted a right.
- The condition that must be met before the right can be exercised: A condition specifies the terms, conditions and obligations under which rights can be exercised.

Recently, the MPEG-21 Media Value Chain Ontology (MVCO, part 19 of MPEG-21 standard) has been proposed. It is a relatively small ontology, (less than 60 classes and 20 properties), simple to understand and accompanied by a forthcoming Java API. The MVCO represents the Intellectual Property (IP) along the Value Chain. There are different kinds of objects of the Intellectual Property (called IP Entities) and different actions that are performed on them, what defines the different roles that users can play regarding these IP Entities. These elements, along the permissions to execute the actions, constitute the essence of the MVCO. The MVCO Ontology can be extended as any other ontology by adding new derived terms, by adding new relations, etc. The MVCO can also be used in conjunction with other ontologies, provided that a matching of some key terms is adequate, for example, alignment with MPEG-21 REL Right or Creative Commons.

MPEG-7

MPEG-7 has been described in Section 4.2. In this context, content usage information tools are of interest. They describe information about the usage process of the AV content, including access rights, financial costs and incomes and the availability for use of the content.

4.9 Device Properties and Capabilities

4.9.1 Definition

This section discusses metadata describing the capabilities and features of terminal devices as well as static and dynamic properties and settings. The end terminal is considered to be the last node of operation within the workflow of a FascinatE media playback, performing the final tasks of presentation and perspective selection.

Hence, metadata kept here are either to be used by the rendering nodes to condition the media stream, to be used by the service management system to decide what will get passed, or to inform the user about what the device is capable of during the presentation of a FascinatE live stream.

4.9.2 Requirements

Initial scene rendering operations and potential interactions were defined in FascinatE Deliverable 5.1.1 [Borsum, 2010], indicating completion in a later version of that document. This section will define more detailed metadata descriptions from the terminal side and will be referred to as the final version of D5.1.1.

Considering that the terminal is contained within or connected to the video panel or sound reproducing box, a property list should contain information for:

- **General**
  - Firmware version (of FascinatE application)
  - Number of available (encoders)/decoders (dynamic: number of idle (de)coders)
  - Available (encoder)/decoder set (e.g. DTS, Dolby Digital)
  - Number of available/spare memory
  - Latency per decoder path i.e. audio lip synch parameters as defined in HDMI
  - Decoder switching times (has influence on the required buffering)
  - Regional settings (language, local time…)
  - Device status (idle, streaming, paused), potentially per decoder
  - Hot plugging capability, Device UUID
  - Offered level of user interaction, i.e. interactivity
  - A/V timing relationship, supported time code format, default synchronization method and capability (jitter performance)
  - Properties describing the presentation scenario (acoustic reflections, ambient light,…)
  - Parameters to provide statistics to ensure quality of service and quality of user experience (packet loss, response time until the system was able to fulfil a user request)
- **Acoustic**
  - Speaker/channel assignment
  - Number and arrangement of speakers, especially if a standard speaker setup such as stereo, 5.1, 3rd order Ambisonics[16] or wave field[17] is available
  - Speaker characteristics that have to be compensated, e.g. delay and frequency range for each single speaker
  - For high-end setups with Ambisonics or WFS also the real angle/position of all speakers, this might also be useful for 5.1 setups
  - World clock
  - Sample rate
  - Data format (PCM, 1 bit serial…)
  - Placement in packaging (relative to video) + packet timing
- **Visual**
  - (Panel) resolution
  - (Panel) refresh rate / frame rate
  - Number of display channels (forming one image)

[16] Superposition of soundfields with functions of describing the surface of a sphere; http://flo.mur.at/writings/HOA-intro.pdf/view

[17] http://www.holophony.net/Wavefieldsynthesis.htm

- Number, type and arrangement of locally connected displays, especially the angles and positioning of multiple displays to each other
- Pixel format/encoding (potentially of panel, e.g., 24/36/48bits RGB,YUV,YUVA)
- Brightness/Contrast/Saturation
- Colorimetry/Gamut

This could be simplified by defining terminal classes compared to MPEG profiles and levels as proposed in D5.1.1, Section 4. Suggested in D5.1.1 was additionally an interactivity performance indicator.

Metadata is passed to allow user selection or to inform on content related/unrelated topics. Hence, it needs an own presenting block, usually performing just an overlay of text or graphics. This results in required terminal properties to present fonts or graphic objects (see MHEG-5 spec).

4.9.3 Candidate Formats

The FascinatE terminal is likely to be a generic/hybrid platform able to run other applications beyond FascinatE plugins, apps or widgets. Hence, it is required to use common formats established in current or upcoming interactive TV terminals.

In this context the following specifications or industry standards forming interactive TV middleware are of interest:

- YouView: http://www.youview.com/developer-zone/resources/
- HbbTV: http://www.hbbtv.org/
- MHEG-5: http://www.mheg.org/users/mheg/archives/doc/mheg-reader/rd1206.html and http://www.impala.org/
- MHP/GEM: http://www.mhp.org/
- OpenIPTV: http://www.oipf.tv/specifications.html

Following that, terminal properties are normally defined as requirements (OIPF T1 R2 Service and Platform Requirements v2[18]). Here also commands for terminal/service control are referenced that can complement interactive commands defined in D5.1.1 chapter 6.1 or D1.1.1 - or are given in the form of conformance to profiles and levels (features and performance/resources). The optional return channel is provided in these standards by an IP based broadband connection.

Further, to support terminal identification in networks and differentiate terminal classes for common rendering operations the specifications given above reference to protocols such as *UPnP*[19], *NAT*[20] and interoperability standards like *HAVi*[21] or *Bluetooth/SDP*[22]. While interfaces supporting these are used for service discovery, an important aspect for FascinatE is the device discovery capability. This allows use cases as mentioned in D1.1.1 to engage mobile terminals selecting regions of user interests.

Finally, to provide cross platform functionality to present FascinatE content on a terminal, support for 2D and 3D graphics applications will be required. Using APIs like *WebGL*[23] and formats like *VRML/X3D*[24] or *MPEG 4 BIFS*[25] allows embedded Web browsers to create 3D graphical objects. *DOM* and *CSS* are used to structure the presented data and mechanisms using the “Digital storage media command and control” (*DSM-CC*) toolkit specified in MPEG 2 part 6 and the *XMLHttpRequest* (XHR) API are used to exchange messages with communication partners in the network.

---

[18] http://www.oipf.tv/docs/OIPF-T1-R2-Service-and-Platform-RequirementsV2_0-2008-12-12.pdf)

[19] Universal Plug and Play (http://www.upnp.org)

[20] Network Address Translation

[21] Home Audio/Video Interoperability Specficiation 1.1. issued by HAVi Inc. an industry consortium, introducing device classes

[22] Bluetooth Service Discovery Protocol (http://www.bluetooth.com/English/Technology/Works/pages/sdap.aspx)

[23] Cross platform 3D graphics API based on OpenGL ES 2.0 (http://www.khronos.org/webgl/)

[24] 3D graphics and multimedia framework (http://www.web3d.org/x3d/specifications/#vrml97)

[25] Binary format for scenes (http://mpeg.chiariglione.org/technologies/mpeg-4/mp04-bifs/index.htm)

Low performance terminals supporting FascinatE may get processing support by network rendering nodes but will not allow too complex structures to deal with metadata databases and presentation engines. While high performance terminals may host a Java Virtual Machine to supply automation features, common low power terminals will need to support simpler mechanisms such as HTML page presentation adapted by transmitted JavaScript modules. To overlay this information with media streams an API for the sound and graphic rendering engine is required supporting OpenGL or ASIO protocols.

While visual presentations are mainly influenced by ambient light conditions and viewing distance, acoustic presentation depends strongly on the behaviour of the listening conditions given by the sound reproduction environment. Properties of such environments can be retrieved by using calibration microphones (MMCA/Pioneer, YPAO/Yamaha, Audyssey MultiEQ) and compensated by equalizer settings. Recommendations how to assess such properties are given in ITU Recommendations (see ITU-R BS 1283 “A guide to ITU-R Recommendations for subjective assessments of sound quality).

4.10 Network Properties and Capabilities

4.10.1 Definition

Metadata describing network capabilities as well as static and dynamic properties/settings for groups of users in the distribution network.

4.10.2 Requirements

Main requirement is to provide to the FascinatE system some view on the network capabilities and actual performance, mainly in terms of transmission delay and bandwidth available for the FascinatE services.

Essentially two types of approaches may be considered, each with implications on the high-level design of the FascinatE system:

- Either the knowledge about the network capabilities is static and based on fixed assumptions (e.g. known network design), on long-term statistics or on predefined models of end-to-end delivery channels that the system is required to support.
- Or the network conditions are regularly monitored during operations and the model of the delivery channels updated in real-time or quasi real-time.

A second aspect is the usage of the network-related metadata in order to inform/steer the scripting processes.

- The most straightforward usage of the information gathered on the delivery channel conditions is to steer the rendering scripting process. They can for instance influence the way the AV data will represented and coded (various fidelity points, tiling patterns, maximum number of streams to be shown in a tiled image, …)
- In principle, the metadata on the network capabilities can also be used by the production scripting engine. But this actually means feeding metadata from the network/service provider domain back into the production domain. Deciding whether to allow this feedback when designing the FascinatE system can be driven by different types of considerations:
    - At the business level, this flow of information from the service provider into the production domain can be constrained by some business agreement and by the openness of the interfaces of the respective domains.
    - Production scripts pertain to semantic aspects of the layered scene and to how the scene can be consumed by end-users. This suggests that the impact that the network metadata have on the production scripts should be kept limited, depending on the actual use cases.
        - For instance, a real-time update of the production scripts depending on a dynamic model of the delivery channels is not a desirable feature. It would mean that the end-user would see the semantics of his FascinatE experience

being modified or even altered by the network conditions, including e.g. bandwidth fluctuations not expected by the end-user.

- A more practical use case is to steer the production scripting with a coarse and pre-defined classifications of the delivery channels conditions, which can be translated by the service provider in explicit SLAs towards its end-users. In this situation, the end-user is supposed knowledgeable of the service he/she subscribes to and of the (permanent) implications it has on its experience of the FascinatE content.

Finally, a third aspect is the scope of the network metadata and how they are gathered:

- Monitoring Network Segments: As will be seen in the next section on candidate formats, the tools for network monitoring are very diverse and many of them are only valid for managed networks. Not only these various monitoring methods are not integrated or operating with each other, but also, in many cases, the end-to-end delivery path will be made of heterogeneous network segments. Therefore breaking down the monitoring of the end-to-end path into these various segments is usually an unpractical approach.
- Active measurement of end-to-end network performance: another approach for modelling end-to-end delivery conditions is to actually monitor on-the-fly the state of end-to-end transport (e.g. TCP) or application (RTP, HTTP steaming) sessions. Rather than monitoring separately network segments (to then infer the end-to-end conditions), one can for instance directly monitor the throughput or packet loss rate that an end-to-end session is experiencing, modelling the network as a single “end-to-end” pipe.
- Managed Network Parameters: in the case of fully managed networks (IPTV) or any network where, by design, we can assume that the necessary resources are reserved (e.g. frequency range for broadcast), the end-to-end network conditions can be provided as predefined system parameters.

4.10.3 Candidate Formats

MPEG-21 UED [Burnett, 2006] format contains a category on network characteristics. It includes both *static* network characteristics (maximum capacity, minimum guaranteed, in sequence delivery, error delivery, and error correction) and dynamic ones (available bandwidth, delay, and error).

However the actual information of sources that allows to model the network conditions can vary in nature and scope.

TCP/IP Stack Monitoring

A first way to look at the state-of-the-art is to overview the tools for generic QoS metrics that have been defined at different layers of the Internet Protocol Suite:

- Link layer (per Ethernet segment or end-to-end)
  - *IEEE 802.3ah* [Beck, 2005] is a protocol which allows to define measurements at end-points and intermediate points in a L2 network, and exchange messages for connectivity fault management.
  - *ITU-T Y1731* extends the use of IEEE 820.3ah to Performance Monitoring (PM), which allows measurements of loss, delay, and delay variation (jitter).
- Internet layer (IP end-to-end to intermediate router)
  - Based on the *Internet Control Message Protocols (ICMP)* messages, some basic statistics on the IP connectivity and transmission delays can be gathered. The best known example is the *Ping* utility which returns statistics on the packet loss and Round-Trip-Time (RTT).
- Transport Layer
  - *Transport Channel Modelling*: Many authors have proposed methods for modelling TCP channels, including the two following classical references. In [Padhye, 1998], the steady state throughput (assuming an unlimited amount of data to send) of a TCP channel is

modelled as a function of the loss rate and round-trip-time of the underlying IP connection. [Cardwell, 2000] focuses on short-lived TCP flows, where the connection establishment and data transfer latency are modelled as a function of the data transfer size, round-trip-time and packet loss rate.

- *Session Monitoring*: Naturally, as transport protocols such as TCP normally operate end-to-end and have feedback mechanisms to ensure lossless data delivery, statistics on throughput, and latency of any active session can in principle be directly gathered by monitoring the state of connection either at the server or the client side.

- Application Layer, where typically monitoring of the end-to-end session can be performed:
  - The Real-time Transport Protocol (RTP) is an application-layer protocol defined by the IETF [IETF-RFC3550, 2003] for the delivery of data with real-time characteristics. In practice, it offers a convenient mechanism to carry audio and video streams over TCP or UDP. An important companion protocol is the *RTP Control Protocol (RTCP)* which in particular defines a format and mechanisms to gather QoS statistics on the end-to-end delivery of RTP streams. One typical problem for the use of RTCP, for video services such as IPTV, is the explosion of the number of RTCP messages issued by the RTP clients. In the FP6 MUSE[26] project, techniques have been developed for the in-network interception of RTP and RTCP packets and derivation of statistics on the packet loss, RTT and jitter [De Vleeschauwer, 2006].
  - HTTP is nowadays one of the most used protocols for delivery of audio and video over the internet. Usually, the technique consists of a progressive download of the content, which is stored in a buffer and played as soon as the buffer reached a given threshold. From the monitoring of the buffer state, one can derive the current throughput of the end-to-end HTTP connection. This idea has been exploited to develop the recent *HTTP adaptive streaming* solutions, where the content is encoded at various bitrates and segmented in chunks of a few seconds. Based on its buffer status, the client is responsible for requesting (via HTTP get command) segments at the most suitable bit rate. Today the actual signalling formats vary greatly between the proprietary implementation (Microsoft Smooth Streaming, Apple HTTP Live Streaming…), although several standardisation efforts are now being launched in several fora (IETF, MPEG, DVB, …).

Management platforms

Many network QoS management platforms exist in the market and may rely either on proprietary or standardised metadata format to represent the various measurements on QoS, It is clearly beyond the scope of this document to make a comprehensive overview of them here. But we can shortly mention some standardised mechanism (typically *not* real-time).

IETF has standardised [IETF-STD0062, 2002] the Simple Network Management Protocol (SNMP), which allows querying and sometimes control the state of managed devices attached to the network. The content and format of the metadata exchange is not defined by SNMP itself but is described by the so-called Management Information Bases (MIBs). Each type of MIB defines a hierarchical format that organizes the information gathered about the managed objects. IETF has defined MIB formats for a large variety of protocols: IP, TCP, UDP, etc.

Note that several IETF protocols for network communications also have to inherently maintain some management information, as part of the delivery mechanism. A notable example is the Internet Group Management Protocol (IGMP) which is required to manage information on group membership in IP multicast. As an example, keeping track of the IGMP messaging can be used to monitor the number of clients “listening” to a given IP multicast channel.

Broadband Forum (previously known as the DSL Forum) has

- TR-69 specifies the CPE WAN Management Protocols, a SOAP/HTTP-based protocol, which allows, among other things, a Service Provider to manage and retrieve data on the status and performance of Customer Premises Equipment (CPE). The set of managed home devices

[26] http://www.ist-muse.org

include the internet gateway device (e.g. an xDSL modem- router), but also other types of networked devices that are typically found in a home network: set-top box (STB), NAS… For each type of managed device, a so-called data model is defined; specifying which type of metadata can be retrieved by the TR-69 mechanism. Note that TR-69 does not put specific requirements on real-time monitoring. For example,

  - TR 98 [BBF-TR98, 2006] specifies the data model for the internet gateway device. It allows in particular a system to retrieve packets and byte counters, information on QoS settings, etc.
  - TR135 [BBF-TR135, 2007] specifies the data model for an TR-69 enabled STB. The available statistics are split into 7 categories: Statistics are broken down into seven categories: De-jittering, RTP, MPEG2-TS, video decoding, audio decoding, video response and high-level metrics. For instance, RTP and MPEG2-TS statistics include some metrics on the transport performance: counters of packet received and lost, buffer status, etc.

- TR-147 specifies the requirements for a Layer2 Control Mechanism in the access and aggregation networks. These requirements are now being used by the IETF work on the Access Node Control Protocol (ANCP) [IETF-RFC5851, 2010]. In those networks, a Broadband Network Gateway (BNG) acts as the gateway between the regional IP network and the Layer 2 aggregated traffic from the access nodes (e.g. the DSL access multiplexers). ANCP allow the BNG to monitor and control the access nodes. As an example, if multicast IPTV channels are being transmitted, the BNG is able to know how much multicast bandwidth are being transmitted on the DSL links and then is able to control how to shape unicast traffic.

4.11 End User Profiles

4.11.1 Definition

Metadata describing preferences of an end user.

4.11.2 Requirements

Besides terminal characteristics and/or social trending, there are user's preferences shown through the user profile, which should be bear in mind when choosing what to offer to the user. According to so far done user studies, it seems that the users want both more interaction and to keep the laid-back medium that TV has always been. Thus, we have to provide tools to create a balance between these aspects.

In achieving the balance, one of the most important issues is enabling different levels of interactivity. For example, with pre-configuration through the user profile, it is possible to have laid back, personalised experience of the live broadcast. Whereas, there are number of factors connected to the user interactivity and the achieved experience:

- *Levels of interactivity***:** Levels range from no interaction at all to extensive interaction.
- *Concept of presets:* If users can customise their viewing preferences before the live event (pre-configuration) that takes some load off the interaction during the event. It is simple, and more importantly it doesn't take a focus from the game/event. Users' preferences can be viewed with the respect to e.g. event type, venue, athlete's performance, nationality/team, statistics, situation, camera view (viewing angle/place) or viewing context.
- *Stream preferences:* It could include:
  - Number of streams to be suggested.
  - Whether suggestion of available streams is done on demand or on change.
  - Sort criteria: priorities, ratings, user interest etc.
  - Dynamic playlist based on the history what was watched before.
- *Alerts preferences***:** The question here is if users want to be notified of automatic scripts (e.g. players begin followed by cameras), replays, producer recommended view etc., in what topics

they are interested and what would be their preferred way of receiving different alarms (e.g. appearance of the small screen with suggested topic, sound alarm etc.). Also, such alerts might be sent by producer or by followed friend.

- *Social preferences:* The way a user wants to interact with the system may in the great extent depend on the social setting. There are different examples of that:
    - *Following a friend or most popular view on the game.* This would include rating of user profiles or/and voting for the most popular stream (possible just by collecting viewing statistics)
    - *Sharing preferences.* Including information about if the user wants to share the view, with whom and what. An example is adding the possibility to share interesting parts (video clips) instead of asking as it is now: did you see a certain situation? Users could share it (as it, in some extent, happens now through e.g. YouTube).
- *Interaction technique preferences:* The choice of a preferred interaction method with TV set (in case of available alternatives).
- *Interface preference:* This would include number of streams to be displayed in parallel, their relative position and size, but also, the way additional streams would be offered, the way alerts would be handled, i.e. users would be allowed to design their own interface to some extent.
- *Hierarchy level:* In the case of more users, a hierarchical control between users could be defined.

An example where such a user profile might be used is in the "sports bar scenario", where a user profile could be oriented towards following certain team. It would enable a group of fans to get the view closer to their preferences and interests with no or little interactivity.

However, the user profile content described here is based on the currently available results from user studies and there is still much more to study.

4.11.3 Candidate Formats

MPEG-7

User interaction tools describe preferences of users pertaining to the consumption of the AV content, as well as usage history. AV content descriptions can be matched to the preference descriptions in order to select and personalize AV content for more efficient and effective access, presentation and consumption.

The description schemes related to describing user information are the focus of the MPEG-7 User Description Profile (UDP) [MPEG-7 P9, 2005]. The description tools from this profile are also used for representation of user profiles in the TV Anytime standard [TVA, 2005].

The MPEG-7 user interaction tools are seamlessly integrated into the MPEG-21 DIA User Characteristics descriptions.

It seems possible to represent preferences specific to the FascinatE system (interaction methods, alerts) using the MPEG-7 user preferences tools together with appropriate classification schemes. The MPEG-7 user preferences DS has a weighting model that could also be used to express the hierarchy of users. If a detailed description of the interface preferences is needed a new type of browsing preferences could be defined for this purpose. Social preferences seem not to be fully covered by MPEG-7, however, it should be possible to cover simple cases. Depending on which complexity of social preferences is needed in the FascinatE system, extensions might be necessary.

4.12 User Interactions

4.12.1 Definition

Metadata describing feedback generated from end user interactions.

The user interactions allowed by the FascinatE systems are limited by the set of interactive commands that the FascinatE system is able to perform. These commands control the audio and video renderers and define the interface between them and the end user interactive system.

A preliminary list of interactive commands is proposed in D5.1.1 and can be summarised as follows:

- *Decides framing*: Selecting a region of the screen as framing
- *Selects objects for tracking*: Selecting an object that will be followed from now
- *Scroll X-Y*: Navigating through menus or moving in an OmniCam view that occupies more space than the interface screen
- *Select / Back*: Used to select or de-select in menus, or navigating in the interface
- *Zoom*: Zooming in camera views
- *Slow motion replays*: Asking for replays on user request
    - *Start of replay*: Defines the starting time of the replay
    - *Duration of replay*: The user can select the duration of the replay
- *Select windows* (multiple cameras on screen): Ability to create compositions in the screen with multiple cameras.
- *Separation foreground / background sound*
- *Set gain on dialogs*: Choosing the volume in dialogs
- *Select channel*

From the above interactive commands, it is clear that the scope of some user interaction metadata will remain within the user terminal. For instance, the commands “setting gain on dialogs” or “select / back in navigation menus” will not need to be transmitted through the network. However, in some scenarios where limited network resources or terminals are used (such as mobile phones) other interactive commands, such as “slow motion replays”, may need to be transmitted. Specifically, the Scripting Engines might have to be informed of the current state at the user terminal.

4.12.2 Requirements

The metadata of interest for the user interaction has the following properties:

- It should not add any new latency to the system. User interaction generally needs low latency when responding to user commands as it is crucial for the system’s acceptance.
- The granularity of the metadata must be enough to be able to a) not add extra delay to the system and b) correctly specify or identify regions in the video image or points in time.
- The correlation with the metadata describing the layered scene representation proposed by the FascinatE project must be high enough to provide all desired functionality.
- A high correlation with the Scripting Engines is needed to be able to provide the user with options to follow tracked objects (such as players in a football match).

In particular, the following description should be supported by the metadata describing user interaction:

- Describe and identify all cameras and camera groups in the layered scene representation (including omni-cameras and personal zoom cameras)
- Describe the exact position in the scene (as regions that are being followed by zoom cameras)

- Consider limitations of the system in the user interface, such as:
  - Maximum or minimum amount of zoom allowed
  - Possibility of framing any part of the omni-camera view or only framing to existing real cameras
  - Maximum or minimum volume of speakers
- Amount of storage space/time so users know if a history of selection of replays is still available
- Describe and identify all audio signals in the scene so users are able to separate foreground and background audio

4.12.3 Candidate Formats

MPEG-7

The MPEG-7 user interaction tools also include a description scheme for recording user actions in a generic way. However, it does not seem to be very appropriate for real-time information.

SMIL

User interface elements and information enabling user interaction can be described using SMIL (spatial composition). See detailed description in section 4.6.3.

5 Synergies and Gaps

This section analyses if all the requirements can be covered by existing formats, identifies possible gaps and discusses how the formats needed for the different types of metadata can be combined, linked and extended.

5.1 Sensor Parameters, Calibration Metadata

There are several established formats for technical metadata that also provide support for embedding metadata into structures of essence containers such as MXF. The formats are similar in scope and partly harmonisation activities between different standards bodies (e.g. EBU and SMPTE) are already taking place.

There is currently a lack of support for several calibration and lens parameters and initiatives to add these parameters are only starting. FascinatE will try to follow these initiatives and contribute where possible.

5.2 Content Description

MPEG-7 seems to cover the requirements well. The main issue is that most tools for generating and processing MPEG-7 descriptions are based on the assumption of a static document describing the complete content. In FascinatE this is not the case, but only a time window is relevant, and outdated descriptions can be discarded while new ones are constantly added.

5.3 Domain/Scene Knowledge

No complete format exists beyond custom XML or OWL. The challenge is that the information to be modelled strongly differs in terms of granularity, abstraction level etc. OWL seems to be well suited for high-level information, but there might also be numerical data, for which the modelling with OWL can become cumbersome.

5.4 Production Rules and Visual Grammar

There is no standard format at the moment. The choice of the format will strongly be influenced by the choice of a rule processing engine. Efficient evaluation of complex structures of rules and principles is a key requirement for the quasi real-time decision making task of the Scripting Engine. Any ECA rule format is a candidate as of today.

5.5 Script Templates

NSL

There are three major issues for using NSL in FascinatE. First, NSL was developed for offline scenarios where the structures were carefully developed by professional film staff with access to the video clips already at production time. Usually, the metadata structure/annotations and the narrative structure were created simultaneously - which is not the case here. Simply put, NSL enables very sophisticated semantic abstract clip descriptions by heavily relying on a good amount of metadata. In the live scenarios of FascinatE, basic metadata will be available from audiovisual analysis and probably more than that through cue processing and semantic inference, but the quantity and quality cannot be expected to match what is possible in offline scenarios where there is more time to annotate.

Secondly, NSL structures are very well suited to build the general structure behind a plot that is roughly known beforehand. As also the video content is available usually when developing the narrative, the author can adjust the level of freedom in the story according to the amount of AV content actually available for a certain part. By defining abstractly which clips are suitable at a certain point in the video, the runtime instantiation is able to produce very different videos dynamically, where certain topics are

more or less prominent, certain subplots are enhanced or missing, or certain parts are skipped completely. Resulting videos may differ in length considerably. NSL is well suited for editing film-type videos such as documentaries or dramas, where usually a story is told from start to end, but that narrative arc can be told by multiple combinations of video clips, and may also vary in length. In contrast to that, the system was never used for broadcasting live events so far. The NM2 scenario "My News Sports My Way" dealt with live news content, however, the technical difference between processing up-to-date annotated news content from a TV broadcast repository and processing live streams is significant, therefore the scenarios cannot be compared in that aspect. While the TA2 MyVideos scenario is about capturing, authoring and sharing clips from live events such as school concerts, it is an asynchronous (offline) scenario, and therefore can't be compared well to FascinatE either.

The third reason is that the specification is simply not publicly available at the moment. We plan to re-investigate the applicability of NSL once it is published.

Isis / Cabbage / Viper

The main problem with those technologies is that, to the best of our knowledge, they are no longer developed and/or supported. While we like the ideas behind those results, it is unlikely that either directly using those components or building on top of them is the best solution for FascinatE. In the years since Isis/Cabbage/Viper were released, research developments along various dimensions have been performed, making the toolset essentially outdated.

5.6 Scripts

RUBENS and I2Vision play-map

The two playmap formats presented above fulfil some of the requirements identified for the FascinatE Delivery Scripts. In particular, they provide the addressing mechanisms to retrieve the AV segments as well as the possibility to define switching points between segments and refer to other playmaps, so as to expose some degrees of interactivity at the delivery level. However there are also two important mismatches with respect to the FascinatE targets.

First, these playmap formats were developed for the transport of interactive Video-on-Demand assets, assuming that a full ingest of the content can performed before the playmaps are released. In addition, the format itself relies on the definition of time segments, which assumes that the end point of a segment is known before the streaming of that segment can be started. This is not in line with the requirements of the project, which focuses mainly to live content delivery.

Second, the playmaps were designed as a single piece of metadata, containing information pertaining to both the transport and the semantic aspects of the content. In FascinatE, we rather advocate an approach where the two aspects are decoupled into two classes of scripts (FRS and FDS).

As a conclusion, the playmap concepts introduced in this document constitutes a good starting point to work on the features required by the FDS mechanisms. But the FDS format does not need to be seen as a formal extension of the RUBENS and I2Visions playmap concepts.

MPEG-7 and MPEG-21 for Content Adaptation – CAIN Framework

The association of MPEG-7 and MPEG-21 is a good example of a framework for content adaptation and, as such, is applicable to the FascinatE context. However, as pointed out in 5.2, the existing tools from MPEG-7 rather provides static description and therefore do not fully match the requirement that scripts need to be “streamable”, i.e. consists of a series of update produced in a “live” fashion. Similarly, the MPEG-21 DIA provides a suitable environment to define adaptation mechanisms, but the existing toolboxes, notably the ones of the CAIN framework, offer little for the FascinatE specificities, namely for on-the-fly content adaptation with the ability to access and adapt content not only temporally but also spatially (with operations requiring e.g. ROI cropping or even A/V rendering). In conclusion, we expect that the implementation of most the FascinatE mechanisms for content adaptation will require brand new features to support the real-time and fine-grained adaptation needed in this project. In the course of the development, we will evaluate whether it is feasible (and of interest) to maintain a compliance with the MPEG-7/21 frameworks.

SMIL

Using the SMIL format throughout the FascinatE workflow makes most sense if the terminal video player is directly supporting it. As it is unlikely that *every* frontend renderer in the FascinatE system is supporting it, it might make little sense to use it that way and transform scripts for all other cases. For cases where SMIL enters the picture, the limits will be checked against FascinatE requirements in detail. It is likely that requirements inherent to FascinatE Rendering Scripts are partially out of scope and further developments have to be made. If SMIL is not chosen as the basic format for scripts in the workflow, it might still be used at certain terminal types. A relevant strength of SMIL is the simplicity of describing the spatial composition (screen composition) of content.

NSL

Already discussed above.

5.7 User Annotations

MPEG-7 seems to cover the requirements. The use of Media Fragment URIs would be a more lightweight solution, but is limited w.r.t. representing moving objects efficiently. In both cases, a knowledge base of relevant objects, persons and events is needed, to which the content descriptions refer. Such a knowledge base could be represented using e.g. OWL or SKOS[27].

5.8 Rights and Licensing

MPEG-21 provides a comprehensive solution for modelling information related to rights and licensing, especially with recent extensions such as the Media Value Chain Ontology. It seems appropriate to choose a subset of tools from those offered by MPEG-21.

5.9 Device Properties and Capabilities

With the host of specifications discussed in Section 4.9 there is a need to exclude certain technological approaches from the FascinatE system to concentrate the research on FascinatE's key technology innovations.

A proposed solution is to

- Exclude JAVA (i.e. MHP, MHEG-6)
- Include JavaScript (i.e. allow the user interaction server to run their own little application on the terminal browser)

This will focus the study on FascinatE terminals initially on systems processing declarative meta data, which provides representation of objects, events and their relationships and states.

FascinatE itself will develop a new media stream representation with a *layered scene description* and an new scripting scheme that later might require to add procedural meta data processing containing control information to use declarative content.

5.10 Network Properties and Capabilities

MPEG-21 UED seems to be a standard that is capable of representing all required network parameters, both static and dynamic ones. For the dynamic parameters, the type of measurement source is important, and in the FascinatE context mainly source capable of reporting real-time information are relevant.

[27] http://www.w3.org/TR/skos-primer/

5.11 User Profiles

The MPEG-7 user interaction tools seem to cover most of the required tools, probably with the exception of more complex social preferences. An advantage of the MPEG-7 user interactions tools is that integrations with both MPEG-21 and TV-Anytime, a standard made for end-user devices, already exist.

5.12 User Interactions

There does not seem to be a candidate format that matches the requirements well. Further investigation is needed to decide whether it makes sense to define extensions for existing formats or design a new format tailored to the FascinatE requirements.

6 Conclusions

For some types of metadata there are obvious candidate formats, which seem to (mostly) cover the requirements from the FascinatE system:

- Content description: MPEG-7
- User annotations: MPEG-7
- Rights and licensing: MPEG-21
- Device properties and capabilities: HTML + JavaScript (+ WebGL)
- Network properties and capabilities: MPEG-21

For other types of metadata, one or more formats exist, that partly cover the requirements. The gaps can be closed by defining extensions for these formats:

- Sensor parameters and calibration metadata: using EBU HIPS-META and/or SMPTE RP210 as basis
- User profiles: based on MPEG-7

Finally, there are types of metadata, for which no obvious candidate format could be identified, and for which an application specific format (or a comprehensive extension of an existing format) will need to be defined:

- Domain/scene knowledge
- Production rules and visual grammar
- User interactions
- Script templates
- Scripts

Any custom FascinatE format for script templates and scripts with options will built on experiences with the formats discussed in this document. Defining proper formats to effectively represent production (cinematographic, aesthetic, semantic, ...) rules is a research challenge on its own that will be driven by the technical implementation itself.

7 References

[Agamanolis, 2001] Agamanolis, S. P. 2001 Isis, Cabbage, and Viper: New Tools and Strategies for Designing Responsive Media. Doctoral Thesis. UMI Order Number: AAI0803398., Massachusetts Institute of Technology.

[BBF-TR69, 2007] Broadband Forum TR-69, “CPE WAN Management Protocol v1.1”, 2007. http://www.broadband-forum.org/technical/download/TR-069_Amendment-2.pdf

[BBF-TR98, 2006] Broadband Forum TR-98, “Internet Gateway Device Data Model for TR-069”, 2006. http://www.broadband-forum.org/technical/download/TR-098_Amendment-1.pdf

[BBF-TR135, 2007] Broadband Forum TR-135, “Data Model for a TR-069 Enabled STB”, 2007. http://www.broadband-forum.org/technical/download/TR-135.pdf

[Beck, 2005] M. Beck, Michael. “Ethernet in the First Mile: The IEEE802.3ah EFM Standard”, McGraw-Hill Professional, 2005.

[Borsum, 2010] M. Borsum, J. Spille, A. Kochale, E. Önnevall, G. Zoric and J. Ruiz: “AV Renderer Specification and Basic Characterisation of Audience Interaction”, FascinatE Deliverable D5.1.1, July 2010.

[Bulterman, 2008] Bulterman, D. and Rutledge, L. SMIL 3.0: Flexible Multimedia for Web, Mobile Devices and Daisy Talking Books. 2nd ed. Springer Publishing Company, Incorporated, 2008.

[Bulterman, 2009] D. Bulterman, P. Cesar., A. Frey, N. Färber, R. Laiola Guimarães, H. Van Herreweghe, J. Jansen, I. Kegel, F. Kuijk, P. Ljungstrand, W. Van Raemdonck, J. Renckens, C. Tuerck, O. Verde: “Definition of Composition Models and Representation Languages”, TA2 Deliverable D5.1, Feb 2009.

[Burnett, 2006] I.S. Burnett, F. Pereira, R.v.d. Walle, R. Koenen, “The MPEG-21 Book”, John Wiley & Sons, 2006.

[Cardwell, 2000] N. Cardwell, S. Savage and T. Anderson, “Modeling TCP latency”, in Prod. 9th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp; 1742-1751, 2000.

[De Vleeschauwer, 2006] Bart De Vleeschauwer et al, “On the Enhancement of QoE for IPTV Services through Knowledge Plane Deployment”, In Proceeding of Broadband Europe, 2006.[Doermann, 2000] D. Doermann and D. Mihalcik: “Tools and techniques for video performance evaluation,” Proc. 15th Intl. Conference on Pattern Recognition, vol. 4, pp. 167-170, 2000.

[Dean, 2004] M. Dean and G. Schreiber: “OWL Web Ontology Language: Reference”. W3C Recommendation, 10 February 2004. http://www.w3.org/TR/owl-ref/.

[DMS1, 2004] DMS-1. Material Exchange Format (MXF) – Descriptive Metadata Scheme-1. SMPTE 380M, 2004.

[EBU3301, 2005] Metadata for non-tape-based camcorders for broadcast production, EBU – TECH 3301, 2005.

[HIPS, 2010] EBU ECM/ META: HIPS-META (Acquisition Metadata), http://tech.ebu.ch/groups/phips_meta, 2010.

[I2Vision, 2010] S. Van den Berghe, “i2vision: Delivering personalized media experiences”, ICT 2010 Exhibits, Brussels, Belgium, September 2010. http://ec.europa.eu/information_society/events/cf/ict2010/item-display.cfm?id=3575

[IETF-RFC3550, 2003] IETF RFC3550, “RTP: A Transport Protocol for Real-Time Applications”, 2003. http://tools.ietf.org/rfc/rfc3550.txt

[IETF-RFC5851, 2010] IETF RFC5851, “Framework and Requirements for an Access Node Control Mechanism in Broadband Multi-Service Networks”, 2010. http://tools.ietf.org/rfc/rfc5851.txt

[IETF-STD0062, 2002] RFC3411-RFC3418 (STD0062), IETF, 2002. http://www.rfc-editor.org/categories/rfc-standard.html

[IPTC, 2007] International Press Telecommunication Council. NewsML 2.0 Specification. Technical Report (2007)

[Jannach, 2006] Jannach, D., Leopold, K., Timmerer, C., Hellwagner, H.: A knowledge-based framework for multimedia adaptation. International Journal on Applied Intelligence 2(24), 109–125 (2006)

[Kohncke, 2007] Köhncke, B., Balke, W.T.: Preference-driven personalization for flexible digital item adaptation. Multimedia Systems 13(2), 119–130 (2007)

[List, 2004] T. List and R. B. Fisher, “CVML—an XML-based computer vision markup language,” Proc. ICPR, vol. 1, pp. 789–792, Cambridge, UK, 2004.

[Lopez, 2006] F. López, J.M. Martínez, V. Valdés, “Multimedia Content Adaptation within the CAIN framework via Constraints Satisfaction and Optimization”, Proceedings of the Fourth International Workshop on Adaptive Multimedia Retrieval-AMR06, Geneva, Switzerland, 27-28 July 2006, 17 pp. (CD-ROM). (to be published also as post-conference LNCS proceedings).

[Lopez, 2007] López, F., Martinez, J.M.: Multimedia Content Adaptation Modelled as a Constraints Matching Problem with Optimisation. In: Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2007, Santorini, Greece, June 2007, pp. 82–85 (2007) ISBN 0-7695-2818-X

[Magalhaes, 2004] J. Magalhaes, F. Pereira, “Using MPEG standards for multimedia customization”, Signal Processing: Image Communications, 19:437-456, 2004.

[Manjunath, 2002] B.S. Manjunath, P. Salembier, T. Sikora, ”Introduction to MPEG-7: Multimedia Content Description Language”, John Wiley & Sons, Ltd., 2002

[Martinez, 2005] J.M. Martínez, V. Valdés, J. Bescós, L. Herranz “Introducing CAIN: A Metadata-Driven content Adaptation Manager Integrating Heterogeneous Content Adaptation tools”. Proceedings of the WIAMIS’2005. Montreux. April 2005.

[MPEG-4 LASeR] MPEG-4 LASeR, ISO/IEC 14496-20, 2006.[MPEG-7, 2001] MPEG-7. Multimedia Content Description Interface. ISO/IEC 15938, 2001.

[MPEG-7 P5, 2003] MPEG-7 Part 5 Multimedia Content description Interface, Technical report, ISO/IEC 15938 (2003)

[MPEG-7 P9, 2005] MPEG-7 Part 9: Profiles and Levels, ISO/IEC 15938-9:2005.

[MPEG-21 P2, 2005] MPEG-21 Part 2: Digital Item Description, TR, ISO/IEC 21000-2 (2005)

[MPEG-21 P7, 2004] MPEG-21 Part 7: Digital Item Adaptation, TR, ISO/IEC 21000-7 (2004)

[MXFXML, 2006] Material Exchange Format – XML Encoding for Metadata and File Structure Information, SMPTE 434-2006.

[Padhye, 1998] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, “Modeling TCP throughput: a simple model and its empirical validation”, ACM SIGCOMM Computer Communication Review, vol. 28(4), pp. 303-314, 1998.

[Prangl, 2007] Prangl, M., Szkaliczki, T., Hellwagner, H.: A Framework for Utility-Based Multimedia Adaptation. IEEE Transactions on Circuits and Systems for Video Technology 17, 719–728 (2007)

[RP210, 2008] Metadata Dictionary Registry of Metadata Element Descriptions. SMPTE RP210.11, 2008.

[RDD18, 2010] Acquisition Metadata Sets for Video Camera Parameters, SMPTE Registered Disclosure Document, SMPTE RDD 18:2010

[RUBENS, 2010] RUBENS Consortium, “Specification of QoE optimizing network features”, Deliverable 3.2, 2010.

[Schreer, 2010] O. Schreer, P. Kauff, J. Spille, J.F. Macq and P. Rondão Alface: “Draft Specification of Generic Data Representation and Coding Scheme”, FascinatE Deliverable D2.1.1, May 2010.

[Smith, 2003] J.R. Smith, “Semantic Universal Multimedia Access”, in Visual Content Processing and Representation-VLBV03, LNCS Vol. 2849, pp.13-14, Springer -Verlag, 2003.

[Troncy, 2010] R. Troncy, E. Mannens, S. Pfeiffer, D. Van Deursen (eds.): Media Fragments URI 1.0, LC Working Draft Jun. 2010, http://www.w3.org/TR/2010/WD-media-frags-20100624/

[Tseng, 2004] B.L. Tseng, C.Y. Lin, J.R. Smith, ”Using MPEG-7 and MPEG-21 for Personalizing Video“, IEEE Multimedia, 11(1), pp. 42-53, Jan-March 2004.

[TVA, 2005] ETSI TS 102 822-3-1 V1.3.1 Technical Specification. Broadcast and online services: Search, select, and rightful use of content on personal storage systems ("tv anytime"); part 3: Metadata; subpart 1: Phase 1 metadata schemas, June 2005.

[Ursu, 2010] Ursu, M.F., 2010, Interactive TV Narrativity, in A. Marcus, A. Cereijo Roibas, R. Sala (Eds): Mobile TV: Customizing Content and Experience, Springer Human-Computer Interaction Series, pp. 121-139, Springer London.

[Ursu, 2008] Ursu, M.F., Kegel, I., Williams, D., Thomas, M., Mayer, H., Zsombori, V., Tuomola M.L., Larsson, H., and Wyver, J., 2008, ShapeShifting TV: Interactive Screen Media Narratives, ACM/Springer Multimedia Systems Journal 14(2), pp. 115–132.

[van Beek, 2003] P. van Beek, J.R. Smith, T. Ebrahimi, T. Suzuki, J. Askelof, “Metadata-driven multimedia access”, IEEE Signal Processing Magazine, 20 (2):40-52, March 2003.

[Valdes, 2006] V. Valdés, J. M. Martínez, “Content Adaptation Capabilities Description Tool for Supporting Extensibility in the CAIN Framework”, in Multimedia Content Representation, Classification and Security-MCRS2006, B.Günsel,A.K.Jain, A.M. Tekalp, B. Sankur (eds.), Lecture Notes in Computer Science, Vol. 4105, pp. 395-402, Springer Verlag 2006.

[Viper, 2003] ViPER XML: A video description format. URL: http://viper-toolkit.sourceforge.net/docs/file/, 2003.

[Wang, 2003] Y. Wang, J. G. Kim, S.F. Chang, “Content-based utility function prediction for real-time MPEG-4 video transcoding”, in Proc. of ICIP 2003, pp 189-192, September 2003.

[X3D, 2008] X3D, ISO/IEC 19775:2004, www.web3d.org

[XPath, 1999] XML Path Language (XPath) Version 1.0, Technical report, WWW Consortium (W3C) (November 1999)

8 Glossary

Terms used within the FascinatE project, sorted alphabetically.

AV	Audiovisual
Close-miced	Recorded with microphone close to the sound source.
DPX	Digital Picture Exchange Format, SMPTE 268M-2003
DTS	Digital Theater Systems, multichannel audio technology owned by DTS Inc.
Essence	Audiovisual data
HIPS	Harmonisation and Interoperability of HDTV Production, http://tech.ebu.ch/groups/phips
HDMI	High-Definition Multimedia Interface (version 1.4 released 5. Jun. 2009), digital audio/video interface to transmit uncompressed data, http://www.hdmi.org
ID	Identity
IP	Integrated Project; or: Internet Protocol
KLV	Key, length, value: encoding of metadata fields and their values by specifying a key, the length of the content and the binary data of the content, defined in SMPTE 336M-2007
LSR	Layered Scene Representation
MXF	Material exchange format: container for audiovisual essence and metadata, specified as a set of SMPTE standards
NSL	Narrative Structure Language
OOI	Object of interest
PCM	Pulse code modulation
QoS	Quality of Service
ROI	Region of interest
SE	Scripting Engine
SLA	Service level agreement
STB	Set-top box
SVG	Scalable Vector Graphics, http://www.w3.org/TR/SVG
UUID	Universally Unique Identifier, specified by the Open Software Foundation for distributed computing and ISO/IEC 11578:1996, ITU-T Rec. X.667 and ISO/IEC 9834-8:2005
WFS	Wave field synthesis
XML	Extensible markup language, http://www.w3.org/XML/

Partner Acronyms

ALU	Alcatel-Lucent Bell NV, BE
ARI	Arnold & Richter Cine Technik GMBH & Co Betriebs KG, DE
BBC	British Broadcasting Corporation
DTO	Technicolor, DE
HHI	Heinrich Hertz Institut, Fraunhofer Gesellschaft zur Förderung der Angewandten Forschung e.V., DE
JRS	JOANNEUM RESEARCH Forschungsgesellschaft mbH, AT
SES	Softeco Sismat S.P.A., IT
TII	The Interactive Institute, SE
TNO	Nederlandse Organisatie voor Toegapast Natuurwetenschappelijk Onderzoek – TNO, NL
UOS	The University of Salford, UK
UPC	Universitat Politecnica de Catalunya, ES