Representation Techniques for Multimedia Objects

Deliverable 3.1.1

IST FP6-027122

Representation Techniques for Multimedia Objects

SALERO Deliverable 3.1.1

SALERO Identifier:	SALERO-D311-JRS-RepTech4MultObj-v13.doc
Deliverable number:	D3.1.1
Authors and companies:	G. Kienast, H. Zeiner, P. Hofmair, R. Schlatte, G. Thallinger (JRS); T. Bürger, D. Denissenko, A. Wahler (LFUI);
Work package/task:	WP03
Document status:	Final
Confidentiality:	Public

DOCUMENT HISTORY

Version	Date	Reason of change
1	2006-05-08	Document created, call for input.
4	2006-06-19	Major input in chapters 3 and 4.
7	2006-06-22	First draft chapter 5, major restructuring of document.
11	2006-06-23	Draft for final comments before internal review.
12	2006-06-26	Final draft for internal review.
13	2006-07-04	Final version including suggestions from internal review.

The work presented in this document was partially supported by the European Community under the Information Society Technologies (IST) priority of the 6th framework programme for R&D.

Table of Contents

1 Executive Summary

SALERO's overall goal is to define and develop 'intelligent content' with context-aware behaviours for self-adaptive use and delivery across different platforms, building on and extending research in media technologies, web semantics to reverse the trend toward ever-increasing cost of creating media.

Work Package 03 "Media Semantics and Ontologies" aims to improve creation, management, retrieval and (re-)use of all types of multimedia objects through their augmentation with semantic information about their meaning. This document serves as motivation and starting point for further developments in this work package, always keeping in mind that within the scope of SALERO a wide range of multimedia objects is used. These comprise traditional still images and video content as well as 3D animated models of worlds in a computer game or other entirely computer generated content including animated characters lip-synched with an audio track.

As a basis for further understanding chapter 3 "Semantic Representation Techniques" gives an introduction about the concept of ontologies and ontology engineering and describes the currently available ontology languages (mainly OWL and WSML) and their associated tools.

Chapter 4 "Multimedia Description Standards" reviews metadata description standards for multimedia content currently in use. Section 4.9 "Semantic Multimedia Description Techniques" lists attempts to combine conventional multimedia standards, which lack of capabilities to describe semantic context with ontologies and their results.

Finally chapter 5 "Research Objectives and Innovation" outlines the goals for SALERO's research within work package 03 where semantic technologies can improve workflows and significantly reduce human resources needed in content creation, annotation and retrieval.

2 Introduction

2.1 Purpose of this Document

This document serves as motivation and starting point for the upcoming developments in WP03 ("Media Semantics and Ontologies") of SALERO and summarized the state of the art review carried out in the first six months of SALERO.

2.2 Scope of this Document

This document gives an overview on the state of the art on multimedia description and semantic technologies as well as an outlook on which benefits are to be expected from a semantic extension of multimedia formats. It describes multimedia description formats and the semantic technologies but not the essence formats (as JPEG, MPEG-2).

It is important to mention that within this document no decision is made to choose one of the existing standards for further developments in SALERO. This is an ongoing process and will happen during further collaborations with work packages WP02, WP04, WP05, WP06, WP07 and WP08.

2.3 Status of this Document

This is the final version of D3.1.1.

2.4 Related Documents

More information can be found at the SALERO website which can be found at http://www.salero.eu.

It is also recommended to read Deliverable 2.1.1 "Media Objects Dictionary" which can also be found on the SALERO website.

3 Semantic Representation Techniques

3.1 Introduction

In current media productions, multimedia objects (MMO) are in almost every case created from scratch. One reason for the limited reusability of MMO is their poor – or entirely absent – self-descriptive ability and the lack of formal semantic representation of the object's state, purpose and intended meaning. Another issue is the absence of contextual information, which is needed for retrieval operations. There is an evident need for algorithms and frameworks to extract and construct semantic metadata out of the essence, ranging from low-level features as dominant colour to high-level statements like 'this picture is about an elephant drinking water'. This is why we amongst others need a formal (machine-processable) representation of the semantic features and the context in which the MMO is to be used. One major challenge is thus to unify the worlds of media metadata on one hand and expressive ontology languages on the other.

In this section ontology standards for the representation of semantics will be presented.

Early standards of representation languages are CycL [Cyc2002] and KIF, OntoLingua, Frame Logic (F-Logic) [Kif1989] and Description logics [Baa2003]. The next era of ontology languages were XOL [Kar1999], OIL [Fen2001], DAML, and the unification of the two called DAML+OIL [Hor2001], which eventually shaped the Web Ontology Language (OWL) [Dea2004].

RDF [Man2004] is a data model that has been augmented by the simple but popular ontology language RDF-S [Bri2004]. RDF-S offers a minimum of expressiveness and supports little reasoning, but has gained some importance due to the availability of scalable repositories and reasoners.

The most well-known ontology languages are the OWL family (OWL Lite, OWL DL, and OWL Full) and RDF-S. They are also the only ones to undergo W3C standardization or pre-standardization activities. However, both suffer from serious limitations: RDF-S lacks expressivity and even OWL Lite as the most constrained variant of the OWL family requires DL-based reasoning

The activities of the WSMO Working group have yielded proposals of new ontology languages, namely WSML, OWL- ("OWL minus") [Bru2005a] and OWL Flight [Bru2005b]. OWL- is a well-founded reduction of OWL that combines efficient reasoning with a high degree of expressiveness. One possible starting point for a new ontology that will be developed in SALERO could be OWL-(minus) or OWL- Flight. An ontology language shall match the requirements of the semantics of media objects and SALERO will develop full tooling support for this language, and eventually prepare a standards proposal.

In section 3.2 we will give a general introduction to Ontologies, in section 3.3.1 the OWL family and in section 3.3.2 the WSML family of ontology languages will be described in more detail as these languages are relevant for SALERO.

3.2 Ontologies as a Means for Semantic Representation

This chapter contains a general introduction to the emerging field of ontologies within the domain of computer science. Our goal is to give readers an understandable overview and to provide additional knowledge about the range of practical application possibilities.

The term ontology has been in use for many years and ontologies are widely used in different areas like Knowledge Engineering or Computer Science in applications that are related to information integration, information retrieval, knowledge management or in the Semantic Web.

An important stage in the evolution of ontologies was marked by the Semantic Web: The Semantic Web as stated in [Ber2001] is "an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation". Ontologies are an integral part of the Semantic Web as they are a means of standardized communication mechanism.

Regarding the domain of Computer Science they aim to provide a common understanding of a domain and capturing this domain knowledge by modelling its basic terms and relations between them and by providing rules stating restrictions on the usage of terms and relations.

The following section is organized as follows: First of all we try to give answers to the basic questions about ontologies to make readers more familiar with what this is all about. We clarify the general and the computer science related meaning of the term *ontology*.

Later on we bring some light into the nature of an ontology. On the one hand we explain potential benefits of the ontology approach and on the other hand, we describe possible application fields.

In the end of this chapter we deal with technology details and elaborate a practically and easily usable, step-by-step "guide" to the process of ontology building, not without to consider the variety of software tools that help us to support this act.

3.2.1 The Term Ontology

The term ontology was taken from the area of Philosophy and many definitions for this term exist in the literature. Probably the most quoted definition is from Tom Gruber, who defined an ontology as follows: *An ontology is an explicit specification of a conceptualization.* [Gru1993]

An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms. [Usc1998]

Ontology (from the Greek *onto = of being* and *logy = study*) was initially a discipline from the subject of philosophy which has later been adopted to computer science. In its elementary meaning ontology is one of the most fundamental branches of metaphysics which concerns itself with the basics of being and existence:

- What is?
- What exists?
- Is existence a property?
- etc.

Ontological theories try to analyse the most basic problems of "finding a subject, a relationship, and an object to talk about" which is also the basic idea of ontologies in terms of computer science.

An ontology in computer science represents a data model which is able to store knowledge about objects and their relations within a specific domain. Is most often occurs in form of an explorable network that also includes reasoning capabilities. Ontologies generally contain information about:

- **Objects:** subjects an ontology is dealing with. Objects can either be general types (classes) or concrete instances:
    - o **Classes:** entities, collections, general object types;
    - o **Instances:** concrete occurrences of one ore many specific classes. Examples:
        - A pig is an instance of the class animals;
        - Four is an instance of the class numbers;
        - H2O is an instance of the class molecules;
- **Attributes:** properties that can be assigned to the objects of an ontology;
- **Relationships**: describe the relations and the relation types between the objects of an ontology;

3.2.2 Why Ontologies?

Basically, the need for ontologies derives from the way people communicate with each other. Usually, every single statement can be seen as a combination of a syntactic and a semantic part. The syntax defines the simple construction and is given by character arrangements that are resulting in a sequence of words. These words and sets of words (called phrases) would not have any meaning without the semantic component which provides the possibility to describe facts and objects to others in an understandable manner.

In the real world, people need to define a specific conversation domain (even for usual small talk) in order to be able to communicate with each other. Within short conversations such agreements are most often done implicitly and unconscious, e.g. by a simple introduction sentence or question. Without an agreement to a topic statements can be ambiguous. That can lead to misunderstandings in the short and to a *poor communication* in the long run.

Actually, the understanding problem can be even more complicated. The previously mentioned *poor communication* can also appear when people are talking about, on principle, the same subject from e.g. different dimensions or points of view, or by using varying (expert) jargons. To be still able to make a reasonable conversation, the concept of a *shared understanding* is essential, because a lack of those would end up in simple confusion in any case.

In context of an IT system the communication problems are quite similar to those in the real world. A lack of a *shared understanding* immensely reduces a system's interoperability. Let us take any system object, for example one that is very well described. The potential for re-use of such an object and its sharing capabilities will be nevertheless dramatically reduced, because its description will almost in any case explain only one single state and only one single dimension of such an object. The description will probably also contain field-specific attributes that might be similar to notions used in other domains, but without a semantic bridge (a *shared understanding*), it will hardly be possible for these attributes to be connected to their equivalents.

The usual way to address this problem is the effort to fill the gap of the *shared understanding.* This can be done by the creation of a unifying framework by extending (objects') descriptions with additional world views. This important approach is also known as the idea of *conceptualisation*.

Generally, a conceptualisation is defined as:

- “a set of formal rules that constraints the structure of a piece of reality” [Gru1993]

The definition is very simple but powerful. It considers the already mentioned world view extension, as a simple expression set of concepts (including their attributes) and their intra-relationships, to formally illustrate a fraction of the real word.

3.2.3 So What Exactly is an Ontology?

The usual and widely accepted definition of an *ontology* is given by Tom Gruber:

- *An ontology is an explicit specification of a conceptualization.* [Gru1993]

For a even better explanation, let us get into the details of the previously mentioned idea of conceptualisation which represents an approach for a formal world view construction, serving as a *shared understanding* layer. On the one hand such a world view can be defined *implicitly*. A frequently mentioned example is some type of financial software that includes world view presumptions about payment entities, processes and their relations (accounts, invoices, transfers, etc.). In that case the world view is an internal part of the software application logic. On the other hand the definition may also be given *explicitly*. Such an explicit elaborated conceptualisation is usually called *ontology*. From this point of view we can consider an ontology as:

- “An explicit account or representation of some part of a conceptualisation” [Uscho97].

Ontologies can be expressed in more or less arbitrary level of formality. The range of definition possibilities includes at least four major milestones [Uscho96]:

- *highly informal*: expressed loosely in natural language, e.g. glossaries;

- *structured informal*: indeed expressed in natural language, but in a restricted and structured form, increasing clarity and reducing ambiguity;
- *semi-formal*: expressed in an artificial formally defined language;
- *rigorously formal*: meticulously defined terms with formal semantics, theorems and proofs of such properties as soundness and completeness

3.2.4 Uses of Ontologies

This section tries to illustrate the varying purposes of ontologies, as well as the potential benefits for each space of use. We will also try to give some significant examples related to multimedia, to be better able to explain not only the general but also the special purpose of ontologies in SALERO.

Communication

As already mentioned, one of the most important motivations for ontologies, is the potential for a global communication improvement. Ontologies are able to reduce conceptual and terminological confusion, by adding a shared understanding layer in form of a unification framework. Several examples for communication improvements have already been given in the previous sections. All these can also be directly applied to multimedia assets, e.g. for semantic retrieval.

Networks of Relationships and Integration of Different User Perspectives

Ontologies can also be used for supervising purposes, by creating networks of relationships that contain information about what is linked together within a system or organization. Every system or organisation implicitly contains such a network, but users often have different perspectives and points of view. In addition, people in different positions have different assumptions about what a system actually does, its primary goals and how these goals can be achieved most efficiently. A shared understanding is essential in order to be able to integrate different perspectives. Making key relationships explicitly visible by constructing a relationship network helps us to identify logical connections and to unify different or opposite points of view. It also helps to integrate global and local views of a system.

Consistency and Disambiguation

Although “consistency and disambiguation” can be seen as a part of the previously mentioned general “improved communication” purpose they play a very important role (especially for the retrieval process) that are worth to be described more precisely. Previously, we already mentioned the sometimes confusing aspects of our language, so ambiguity is certainly one of them. In context of a software system, ambiguity can lead to fatal inconsistencies. Eliminating it, using the “conventional” way by uniforming the objects, a system would definitely loose flexibility. Our goal is to achieve consistency, but staying flexible with the objects at the same time. For that purpose ontologies can provide additional semantics that can be used to reach disambiguation among system objects, without the need for an uniformisation.

Interoperability

A common application of ontologies concerns the issue of interoperability. The use of e.g. different paradigms, languages and software tools causes a gap that makes it impossible to users to easily transfer or share their data. So called “translators” are required in order to achieve interoperability between different approaches. However it is highly inefficient to create a translator for every single approach pair, because an increasing diversity causes an even more increasing demand for translators (n approaches → $O(n^2)$ translators). In that case using ontologies as inter-lingua can reduce the number of required translators significantly. Compared to point-to-point translations the required amount of translator will increase accordingly to the number of different approaches (n approaches → n translators).

This could also be one of the primary applications of ontologies in SALERO. As we will describe in following chapters, our goal is to create a multimedia ontology that should be able to increase sharing capabilities of multimedia objects. Additionally, such an ontology could also be used as a translator, to integrate multiple, heterogeneous data sources.

3.2.5 Building an Ontology

This section concerns itself with a basic guide for building or engineering ontologies. We aim to provide and explain useful methodologies for the construction of “what exists”.

The first hurdle for the ontology constructor is the question for the scope, the actual part of the world the ontology is going to describe. An excellent approach to achieve that purpose is to summarize all possible scenarios that can arise in the application. That should provide a skeletal framework that can be refined gradually using e.g. simple brainstorming.

The next steps concern the actual ontology building process. These can be divided into the processes of capturing and coding.

Capturing the details of an ontology means the identification of the key concepts and their relationships, followed by the recognition of possible terms and constraints. Their integration into the base capture allows the discovery of possible, previously made, misconceptions. Last but no least the capturing process end with the identification of each concept's properties.

Some guidelines for capturing:

- Try to produce a precise, natural language description of the ontology. Keep three things in mind: clarity, clarity and clarity.
- Ensure consistency among the terms by defining a dictionary/glossary and try to avoid the introduction of new words.
- Integrate different points of view and dimensions to be able to avoid circularity and to indicate possible synonyms.
- Go middle-out: when making the decision, what to describe fist, try always to proceed middle-out rather than top-down or bottom-up. Top-down and bottom-up approaches most often deliver a very high detail level with negative side effects that can affect the re-usability and the extensibility of the ontology.
- Try to give examples to undermine the direction and the actual meaning.
- Make reviews and keep notes about the changes

Coding can be seen as the part, where the ontology constructor does the actual representation of the captured knowledge. It always starts with the choice of the representation language. No general advice can be given here, because the decision depends in all cases on the requirements. Once decided, the actual coding (writing down the code) can begin.

After coding the newly created ontology represents a potentially usable knowledge base that needs to be evaluated. This is most often done automated by applying the ontology in use-case scenarios to measure its potential competency.

3.2.6 Methodologies and Methods for Building Ontologies

Methodologies and methods for building ontologies are validated guidelines on how the ontology building process should be structured in order to be a well-organized process which is important because ontologies are often built through a collaboration of many individuals.

These methodologies define how ontologies can be built from scratch by reusing and re-engineering other ontologies, or by merging of pre-existing ontologies.

On of the most prominent of these methodologies is the Uschold and King methodology [Usc1996] which consists of the following steps:

- Identify purpose
- Build the ontology (capture requirements, code the ontology and integrate existing)
- Evaluate the ontology

- Document the ontology

Other methodologies are for example the Cyc method [Len1990], Grüninger and Fox's methodology [Gru1995], METHONTOLOGY [Fer1997] or the On-To-Knowledge Methodology [Sta2001].

In [Gom2004] the authors present a comparison framework to compare some of the methodologies mentioned above: the Cyc method, Uschold and King's method, Grüninger and Fox's methodology, METHONTOLOGY the On-To-Knowledge methodology and, some others. This comparison framework takes into account the ontology construction strategy of the methodologies, their software support, the ontology development process that they propose, and how they were use in reference projects or applications. The most important or interesting conclusions of this survey are:

- None of the approaches covers all the processes involved in ontology building (Ontology management activities, ontology development oriented activities and ontology support activities).
  Scale from the most to the least complete:
    - METHONTOLOGY is the approach that provides the most accurate descriptions of each activity:
        - Ontology management activities: Scheduling, control and quality assurance (all: proposed)
        - Ontology development processes: Specification, conceptualization, formalization and implementation (all: described in detail)
        - Post development processes: Maintenance (proposed)
        - Ontology support activities: Knowledge acquisition (described in detail), evaluation (described in detail), integration (proposed), configuration management (described), documentation (described in detail)
    - The On-To-Knowledge methodology describes more activities than the other approaches
    - The strength of Grüninger and Fox's methodology is its high degree of formality, but the ontology life cycle is not completely specified in it.
    - Uschold and King's method has some missing points and is less detailed than Grüninger and Fox's.
    - The Cyc method was only used to develop the Cyc KB which is a huge knowledge base containing common consensus knowledge.
- Most approaches are focused on development activities and do not pay much attention to aspects related to management, evolution and evaluation of ontologies
- Most of the approaches do not have a specific tool that gives them technology support. However there are tools for the Cyc method (Cyc tools), METHONTOLOGY (ODE, WebODE, OntoEdit, Protégé-2000) and On-To-Knowledge (OntoEdit with plugins).

As these, their benefits and drawbacks and other methodologies are discussed in several books ([Gom2004], [Sur2006]) and state-of-the-art reports ([Sur2004]) the mentioned methodologies will not be discussed more detailed here. The reader is referred to these books and reports instead.

3.3 Ontology languages

The chapter presents an overview on the ontology languages that are considered as possible languages for the ontologies within this deliverable, i.e. on OWL and WSML and other existing ontology languages. We mainly restrict ourselves to these two languages because they are most important for the Semantic Web and for Semantic Web Services, respectively. Below we describe their peculiarities.

An important aspect that will also be considered is the decidability of the language used. Particularly, inferencing in Ontologies expressed in Description Logics style might become undecidable when rules are simply added to the language without restrictions, an example of such an undecidable combination of rules with an ontology language is the rule language SWRL [Hor2003b] as an extension of the Web ontology language OWL (more specifically its OWL DL fragment). OWL Full for example is already undecidable on itself.

It is important to note, that not all of the existing languages have the same expressiveness and do not have the same reasoning support. Also the knowledge representation paradigms underlying ontology implementation languages are diverse (e.g. frames, description logics…). Corcho and Gomez-Perez in [Cor2000] present a framework that allows analyzing and comparing the expressiveness and reasoning capabilities of ontology languages which could be used in the decision process. The process of choosing and selecting the appropriate ontology language includes questions like:

- What expressiveness does an ontology language have?
- What are the inference mechanisms (reasoning capabilities) of it?
- Are there any supporting tools for that language?
- Is the language appropriate for exchanging ontologies between applications?
- Are there translators that transform the ontology implemented in a source into a target language (to enhance reusability, exchangeability or interoperability)?

3.3.1 OWL Family

OWL was designed because of the need for an ontology language that can be used to formally describe the meaning of terminology used in Web documents, thus, making it easier for machines to automatically process and integrate information available on the Web. This language should be layered on top of XML and RDF (W3C's Resource Description Framework, see http://www.w3.org/TR/rdf-primer/) in order to build on XML's ability to define customized tagging schemes and RDF's approach to representing data. OWL is part of the growing stack of W3C recommendations related to the Semantic Web (see [McG2004]) which consists of

- XML to provide the syntax for structured documents,
- XML Schema for restricting the structure of XML documents and to extend XML with data types,
- RDF as a data model for resources and relations between the resources,
- RDF Schema as a vocabulary for describing properties and classes of RDF resources which also allows generalization-hierarchies of properties and classes, and
- OWL finally adds more vocabulary for describing properties and classes.

OWL – as can be seen in [McG2004] – has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S (see below), and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web. OWL is a revision of the DAML+OIL web ontology language [Hor2001] incorporated lessons learned from the design and application of DAML+OIL.

Generally, OWL can be subdivided in the major dialects:

- OWL Lite: small vocabulary, less expressive, easy to apply
- OWL Full: full vocabulary, more expressive, high complexity
- OWL DL: full vocabulary, even more expressive, high complexity

In our approach, we expect to be rather able to abstract from complexity than to loose any expressiveness. Since OWL DL has the highest expression capabilities within the OWL family, we will focus on OWL DL in this section.

OWL DL can be concerned as an alternative notation for the Description Logic language SHOIN(D) [Hor2003a]. From the point of view of an ontology modeller, the basic idea (or maybe the basic difference) of the Description Logics approach is the handling of roles. These are specified independently from concepts. This allows the contemporary determination of the class of an instance, for which the role was initially defined.

OWL has different syntaxes; the most prominent is definitely the RDF/XML [Bec2003] syntax. Furthermore, there exist also a readability improved, abstract syntax [Pat2004] usable for the OWL DL subset of OWL.

OWL DL has a direct model-theoretic semantics [Pat2004]. However, compared with standard Description Logic semantics, it seems to be more confusing and less intuitive. It can also be shown that entailment in OWL DL (checking whether one ontology is logically entailed by another) can be reduced to satisfiability checking in Description Logics [Hor2003a]. Since almost every major reasoning task can be reduced to entailment, it is safe to say that OWL DL is a notational variant of a Description Logic. Furthermore, it can also be explained that Description Logic is a subset of the First-Order Logic [Bor1996], thus OWL DL is also a notational variant for a subset of First-Order Logic.

While basically being definable as a Description Logic variant OWL adds some peculiarities such as for instance the distinction between abstract and concrete values: OWL allows as instance identifiers URLs which represent objects, and literals which represent atomic values of a concrete domain which are no further decomposable.

As for mediation and combination of different ontologies, OWL supports a simple import mechanism. This feature allows for physical modularization of the ontologies but has no consequences on the semantics: The importing ontology is semantically equivalent to the union of all (recursively) imported ontologies. Subclass, class equivalences or sub-property relations as well as equivalence/disjointness between instances can be defined within the importing ontology in order to relate concepts, relations and instances of the imported ontologies. This allows for the formalization of simple mappings.

OWL-(minus)

The overall goal of OWL-$_i$ is not necessarily to serve as an independent ontology language but rather to define a maximal subset of the original OWL variants which does not fall outside a logic programming framework in order to allow for a straight-forward combination with rule languages. The OWL$_i$ subset of OWL could – as stated in [Bruijn2005a] - serve as a starting point for a more usable Web Ontology language and extensions in various directions which tackle some of the limitations of OWL as for example:

- OWL requires equality reasoning
- ABox reasoning in OWL is hard
- Deriving equality in OWL is non-intuitive
- Difference in the treatment of abstract and concrete values

OWL- is also a family of ontology languages: OWL Lite-, OWL DL- and OWL Full-:

- OWL Lite- is a subset OWL Lite that falls within the DHL Description Logic. It enables efficient query answering using deductive databases and does not require complex reasoning with equality. Furthermore, because this language falls within the Horn fragment of First-order logic, rule extensions of the language are straightforward. Also, OWL Lite- does not allow to derive equality and, in order to stay within the Horn fragment, the language does not allow concrete values.
- OWL DL- is an extension of OWL Lite- and a subset of OWL DL. It is the maximal subset of OWL DL that stays within DHL and thus within the Horn fragment. Therefore, OWL DL- does not increase complexity with respect to OWL Lite-, but does include more features in order to give the modeller more flexibility when creating ontologies.
- OWL Full- is a subset of OWL Full which extends OWL DL- by allowing one to treat the same identifier both as a class and an instance. This feature is beyond popular Description Logics.

OWL Full- is semantically layered on top of OWL DL- and thus, OWL- overcomes the problems with inappropriate layering of the species of OWL.

- More details about the OWL- ontology languages can be found in [Bruijn2005a]

OWL Flight

According to [Bru2004] OWL Flight is an extension of OWL- with different kinds of constraints and a form of the local closed-world assumption, as well as support for datatypes based on the OWL-E datatypes extension for OWL.

In [Bru2004] the authors argue that practical applications require a more elaborate datatype support (e.g. datatype predicates) than OWL has to offer. It turns out that the datatype support brings the language outside of plain datalog. More specifically, the language needs the notion of integrity constraints to implement the language in datalog.

OWL- already overcomes some of the drawbacks of OWL, but in some cases the expressivity had to be reduced significantly in order to get the desired effects (mainly the translation to Datalog). Many limitations of OWL still exist in OWL-, such as the lack of property value and cardinality constraints and the lack of negation.

OWL Flight overcomes these limitations: OWL Flight supports the previously mentioned datatypes, as well as cardinality constraints and value constraints. OWL Flight is based on the Logic Programming formalism datalog, extended with support for integrity constraints, inequality (in the body of the rule) and default negation.

The main benefit of OWL Flight is that it allows efficient query answering.

More details about OWL Flight can be found in [Bru2004] and [Bru2005b].

3.3.2 WSML Family

The WSML family of languages for modelling ontologies and Web service descriptions aims (in its ontology modelling facilities) on tackling several problems in OWL DL.

As opposed to OWL, WSML does not commit to the description logics style of modelling but is more grounded in the logic programming paradigm which is an orthogonal decidable fragment of First-Order Logic. Similar to OWL, WSML defines several dialects. WSML-Core determines the least common subset of datalog that is logic programming without function symbols, within the DL world.

WSML-Flight (based on OWL Flight [Bru2004], see above) extends the basic WSML dialect with additional features from logic programming, namely default negation and integrity constraints (analogue to those used in databases).

Due to their construction WSML-Core and WSML Flight both combine naturally with rule extensions.

WSML-Rule which is currently under development will be another extension of WSML-Flight in the direction of logic programming. The language will capture several extensions such as the use of function symbols and possibly extensions based on HiLog [Chen1993] and transaction logic [Bon1998]. Another possible extension for WSML-Rule is disjunction in the head of the rule, as in disjunctive datalog [Eit1997].

WSML-DL is an extension of WSML-Core which fully captures the Description Logic (SHOIN) which underlies the (DL species of the) Web Ontology Language (OWL). The language can be seen as an alternate syntax for OWL DL, based on the WSMO conceptual model.

Finally, WSML Full which hasn't been defined yet should provide a common umbrella for OWL and the other WSML variants which amount to full First-Order Logic with non-monotonic extensions. Note that WSML is intended to serve not only as an ontology language but to describe in general all relevant features of ontologies, web services, and mediators. However, when speaking about WSML here, we only restrict ourselves to its ontology modelling facilities.

The normative syntax for WSML is a human readable syntax as defined in [Bru2005c]. It is planned to also provide an RDF/XML exchange syntax for all WSML variants like for OWL, as well as a more concise native XML syntax which is closer to its human-readable syntax.

According to OWL, WSML also provides facilities for importing other ontologies. Additionally WSML includes the concept of mediators which define the interoperability between different WSMO components. As for interoperability between ontologies, in WSML it is also allowed to import mediators which link other ontologies not directly but via some conversion, defined in the capability of a so-called mediation service.

Relation to OWL:

OWL has several drawbacks, as pointed out in [Bru2005b]:

- Abstract vs. concrete values are treated differently, in the sense that OWL does use the unique name assumption for concrete values (i.e. literals), whereas it does not for abstract values.
- Restrictions vs. constraints: Given a property of a certain class in an ontology, OWL allows only for restrictions of the range to another class in the sense that membership of the range class is inferred for any filler of the respective property. Similarly, if the cardinality of a property is restricted, whenever more role fillers are specified equality between two or more of therm is inferred. This can often lead to unintuitive results. Particularly, in OWL is not possible to define constraints on the instance data of an ontology with respect to a local knowledge base. This feature would be particularly useful when importing legacy data or mapping between ontologies.
- Datatypes: OWL has a very limited support for datatypes and pre-defined predicates over simple datatypes such as Strings or Numbers. As mentioned above pre-defined predicates and datatypes, (for instance for string concatenation or extraction of substrings) might be necessary for sufficiently describing mappings involving value conversions resolving heterogeneities between different ontologies. Extensions of OWL such as OWL-E [Pan2004] and the pre-defined predicates suggested for SWRL [Hor2003b] take this into account, but have not yet found their way into the OWL proposal. WSML includes a set of pre-defined predicates listed in [Bru2005c] which shall be supported by any implementation.

The main motivation for developing the WSML language was to have a language for WSMO (Web Service Modelling Ontology) that resolves the issues mentioned above. Having an ontology language that resolves these issues like described in [Bru2005b] where OWL-Flight is introduced, amongst others enables efficient query answering which is next to subsumption reasoning one of the primary reasoning task on the Semantic Web.

3.3.3 Other Existing Ontology Languages

In this chapter some other existing ontology languages will be described shortly.

KIF and Ontolingua

KIF (Knowledge Interchange Format) [Gen1992] was the first (serious) effort to standardize a knowledge representation language. KIF provides a Lisp-syntax for expressing sentences of first order predicate logic and also provides extensions for representing definitions and meta-knowledge. KIF is highly expressive but does not provide direct representation primitives for representing ontologies. This can be done with Ontolingua [Gru1992]. Ontolingua is an ontology language based on KIF and on the Frame Ontology [Gru1993a] Ontolingua allows users to build KIF ontologies using ontology primitives such as relations, classes (and subclasses), functions and sets. KIF is a prefix notation of first order predicate calculus with some extensions. It has a declarative semantics and permits the representation of meta-knowledge, reifying functions and relations, and non-monotonic reasoning rules. The Frame Ontology which is built on top of KIF is a knowledge representation ontology for modelling ontologies under a frame based approach. It provides primitives such as Class, Binary-Relation, Named-Axiom and so on.

KIF is a proposed ANSI standard, has limited tool support (there is a theorem prover for KIF expressions and the Ontolingua Server to store Ontolingua ontologies) and it not used by a big community.

F-Logic

F-Logic (short for Frame Logic) [Kif1989] was created in 1995 and initially developed as an object oriented approach to first order logic. It was specifically used for deductive and object-oriented databases and was adapted and used for implementing ontologies. F-Logic is an extension of First-Order Logics by primitives to describe classes, objects and methods. It integrates features from object-oriented programming, frame based knowledge representation and first order logic. Its most significant features are object identity, complex objects, inheritance, polymorphic tags. F-Logic has a model-theoretic semantics and a sound, complete resolution-based proof theory [Kif1995].

F-Logic is not widely used to describe ontologies and there is small tool support as not many implementations of F-Logic exist. On of the extensions of F-Logic is FLORA (see http://flora.sourceforge.net).

Topic Maps

Topic Maps were standardized by ISO in 1999 (ISO/IEC 13250). Topic Maps is a standard for describing knowledge structures and associating them with information resources.

Topic Maps are not an ontology language in the classical sense – rather a language aiming to simplify the browsing of document collections using topics. Modelling primitives are available to define Topics, occurrences of Topics and Aggregations.

A topic is the most fundamental concept in the Topic Maps standard: A topic can be any "thing" – a person, an entity, a concept, etc. – regardless of whether it exists. It is basically the subject of something. Topics have names and types. The second central concept is occurrences: A topic may be linked to one or more information resources that are deemed to be relevant to the topic in some way. Such resources are called occurrences of the topic. Occurrences can have roles. The concepts of "topic", "topic type", "name", "occurrence" and "occurrence role" allow to organize information resources according to topic (or subject), and to create simple indexes. The third central concept – Associations – enables the description of *relationships* between topics.

Topic maps have big tool support and a big community.

Further information on Topic Maps can be found in [Pep2002], [Par2002] and online at http://www.topicmaps.org and http://www.isotopicmaps.org.

RDF(S)

RDF Schema was the first W3C recommendation of a schema language for RDF and is the predecessor of OWL. RDF stands for Resource Description Framework and was developed by the W3C in order to be able to create metadata for describing Web resources. RDF is based on the idea of identifying things using Web identifiers (called Uniform Resource Identifiers, or URIs), and describing resources in terms of simple properties and property values. This enables RDF to represent simple statements about resources as a graph of nodes and arcs representing the resources, and their properties and values [Man2004]. For the definition of vocabularies for RDF, the RDF-Schema (RDFS) specification [Bri2004] was developed. RDF(S) should not be considered as an ontology language, but rather as a general language for describing metadata on the Web. In fact it can be used to describe simple ontologies.

As said above, RDF Schema extends RDF with some basic (frame-based) ontological modelling primitives like classes, properties and instances. The combination of RDF and RDF Schema is not very expressive, just allowing the representation of concepts, concept taxonomies and binary relations.

RDF(S) is widely used as a representation format and there is great tool support for browsing, validating, querying and storing.

3.4 Ontology Tools

There is a wide range of tools available for the management and the dealing with ontologies. These tools can be – according to [Gom2002] – into the following categories:

- Ontology development tools
- Ontology evaluation tools
- Ontology merge and alignment tools
- Ontology based annotation tools
- Ontology querying tools and inference engines
- Ontology learning tools

Ontology development tools can be used to build ontologies from scratch and they also can be used to edit and browse ontologies. These tools also often give support to import/export ontologies or to manage ontology libraries. Ontology evaluation tools are used to evaluate the content and related technologies of ontologies. Ontology merge and alignment tools are used to solve problems of merging and aligning different ontologies. With ontology based annotation tools users can (semi)automatically annotate content of for example web pages. Ontology querying tools and inference engines can be used to query ontologies and perform inferences. And finally ontology learning tools can be used to derive (semi)automatically ontologies from natural language texts or databases.

An extensive review of tools falling in these categories can be found in [Gom2002]. Additionally there is a survey which covers software tools that have ontology editing capabilities [Den2004].

Additional to tools falling into the single categories mentioned above and that are covered partly in [Gom2002] and [Den2004] there are tools or suites that cover several aspects of dealing with ontologies, widely known as Ontology Management Tools.

Ontology Management is how ontologies are stored and retrieved in a system. Ontology Management should thus make ontology data available as the knowledge base of a system, in a scalable, accurate, and secure manner. According to [Mar2004] ontology management tools include the following as the most important aspects:

- Ontology Storage and Retrieval: This includes devices and mechanisms for storing ontology data. This should also include mechanism to efficiently retrieve ontology data
- Ontology Versioning: An ontology versioning mechanism serves to support the evolution of ontologies over time.
- Ontology Libraries: This function refers to general management of ontologies, including tool support for editing, browsing, searching, and maintaining of ontologies.

The requirements and a state of the art analysis of available ontology management tools from 2004 is available here: [Mar2004].

Because of the availability of several surveys and state of the art reports ([Gom2002], [Den2004] and [Mar2004] we only want to point out to some newer ontology management suites in the next sections: DOME (DERI Ontology Management Environment), WSMO Studio and the DIP Ontology Management Suite.

Additionally to the mentioned surveys and reports at http://esw.w3.org/topic/CommercialProducts a list of commercial products that utilize W3C Semantic Web technologies is available and at http://esw.w3.org/mt/esw/archives/cat_applications_and_demos.html you can find a list of various applications and demonstrations of open source tools and services using Semantic Web technologies.

3.4.1 DOME

The DERI Ontology Management Environment (DOME) is an open-source tool, developed by the Ontology Management Working Group (OMWG) which consists of members from DERI, Ontoprise, OntoText and StarLab. The aim of DOME is to provide robust & complete ontology management functionality in form of an easy to use software tool. DOME is offered as a set of freely combinable plug-ins for the Eclipse IDE [Eclipse2006]. Together, they cover all main aspects of ontology management including:

- browsing and editing (dome module)
- merging and mapping (mediation module)
- versioning (versioning module)
- and additionally, ORDI framework integration [Ordi2006] (ordi module)

3.4.2 WSMO Studio

Although there exist some visual similarities between DOME and WSMO-Studio (both open-source Eclipse plug-in based applications) the goals of WSMO Studio are slightly different. Its main objective covers the idea to provide a complete and fully integrated service environment especially for – as the name already says – WSMO.

WSMO Studio includes:

- Editor for WSMO elements (ontologies, services, goals, mediators)
- WSML text editor with syntax colouring
- Import/export filters for WSML, OWL-DL (subset only), RDF and WSML
- Choreography designer, for WSMO based choreographies
- Browser for ontology / service / goal repositories
- Axiom editor
- Integrated WSML Validator
- Integrated WSML Reasoner for consistency checks of ontologies

3.4.3 DIP Ontology Management Suite

DIP Ontology Management Suite is an integrated set for tools for the effective management of ontologies, designed especially for underlying data model use of ontologies in context of Semantic Web applications and services.

The whole suite consists of six major components that are developed (again) as Eclipse plug-ins:

- Browsing & Editing: standard browsing & editing functionality;
- Mapping & Merging: allows to map and/or merge multiple ontologies;
- Versioning: controls the history of an ontology;
- Reporting: a graphical user interface for creating different types of diagram reports;
- Repository: for persistent storage and retrieval of ontology relevant data
- Representation & Data Integration Framework: middle layer & central API that provides transparent access to suite's components;

4 Multimedia Description Standards

4.1 Introduction

An essential requirement for information exchange between different systems is a common representation of data. In the area of description of audiovisual material a range of metadata standards has been developed. A detailed survey of these standards has been made in [Gil2004].

A possible structure categorisation of metadata is:

- Data about the medium: Creation and production information: e.g. creation date and location, title, genre, meta information about the description: e.g. author, version, creation date, etc. Meta information related to the usage of the content: e.g. right holders, access rights. This information (of the points above) typically has to be entered manually, unless it is available in an already existing electronic catalogue. Technical description of the storage media: file and coding formats, image size, image rate, audio quality, etc. Depending on the encoding format, these attributes can be detected automatically.
- Structural description: Structural units: e.g. for video are shot, clips, key frames.
- Content description: Low-level features (colour, texture, shape, motion etc.) and high level features: (content summary, events, scenes, objects)

There are different sources of the feature descriptions to be used for the browsing and searching infrastructure. Sources of the feature descriptions are:

- Existing metadata description are already available descriptions which are either included in the technical metadata of the content or added automatically during production (e.g. GPS device on a camera) or annotated manually. This includes for example production date, location, time, and title.
- Automatic analysis of audiovisual content is capable of extracting a number of features from the content.

4.2 Multimedia Objects Relevant in SALERO

In SALERO a wide range of multimedia objects is used by the project partners. This has to be considered when choosing or developing applications to create, manage or use descript

- Audio objects: recorded sound (mainly speech) as well as synthesized speech.
- Image sequences: TV recordings, computer generated content, film sequences (up to 4K resolution).
- 3D Objects: Animated 3D objects in the application domains of interactive games, special effects for film, 3D objects can range from simple objects to complete “worlds” (designed by 3D software or procedurally generated).
- Any combination of the above, e.g. a computer animated character in a film sequence which has to be lip-synched to a pre-existing or synthesized audio track.

4.3 MPEG-7 (Multimedia Content Description Interface)

4.3.1 Overview

The Multimedia Content Description Interface MPEG-7 ([Mar2002], [Man2002]) has been defined to address a wide range of multimedia content description. MPEG-7 aims to be an overall standard for audiovisual content. Standardization has taken from 1996 to 2002. Possible applications are in the areas of digital audiovisual libraries, electronic news media and interactive TV. MPEG-7 provides standardized description schemes that allow creating descriptions of material that are directly linked with the essence to support efficient retrieval. The audiovisual information can be represented in various forms of media, such as pictures, 2D/3D models, audio, speech, and video. Nowadays, there are an increasing number of cases where the audiovisual information is created, exchanged, retrieved, and re-used by computational systems.

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. As shown in Figure 1, the definition of description schemes and descriptors use the data definition language (DDL) which is largely based on XML schema. MPEG-7 descriptions can thus be serialized as XML (TeM). Apart from this a binary representation of all MPEG-7 descriptors has been defined (BiM).

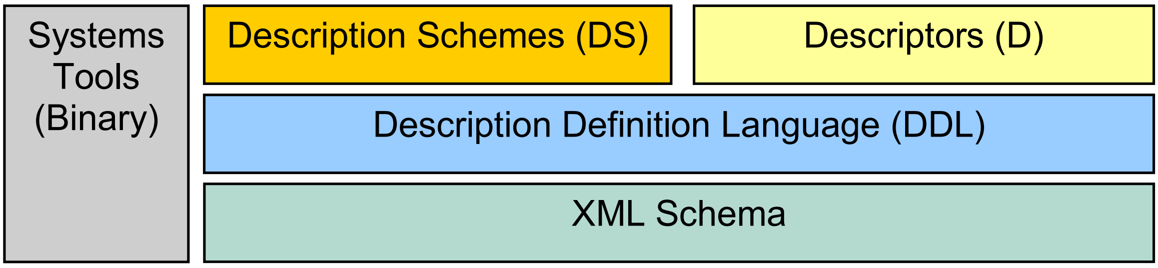

Figure 1: Definition of MPEG-7 description tools.

4.3.2 Structure of MPEG-7

Version 1 of the standard is subdivided into eight parts:

- Part 1 – Systems: specifies the tools (in the sense of description schemas) for preparing descriptions for efficient transport and storage, compressing descriptions, and allowing synchronization between content and descriptions. Part 1 just considers delivery of descriptions, not modifications or updates.
- Part 2 – Description Definition Language: specifies the language for defining the standard set of description tools (Description Schemes, Descriptors, and Data Types).
- Part 3 – Visual: specifies the description tools pertaining to visual content.
- Part 4 – Audio: specifies the description tools pertaining to audio content.
- Part 5 – Multimedia Description Schemes: specifies the generic description tools pertaining to multimedia including audio and visual content.
- Part 6 – Reference Software: provides a software implementation of the standard.
- Part 7 – Conformance: specifies the guidelines and procedures for testing conformance of implementations of the standard.
- Part 8 – Extraction and use of MPEG-7 descriptions: provides guidelines and examples of the extraction and use of descriptions.

Version 2 of the standard will add further parts:

- Part 9 – Profiles and Levels: definition of the adopted profiles and levels
- Part 10 – Schema definition: the XML Schema definition of parts 2, 3, 4, and 5
- Part 11 – MPEG-7 profile schemas: the XML Schema definitions of the adopted profiles

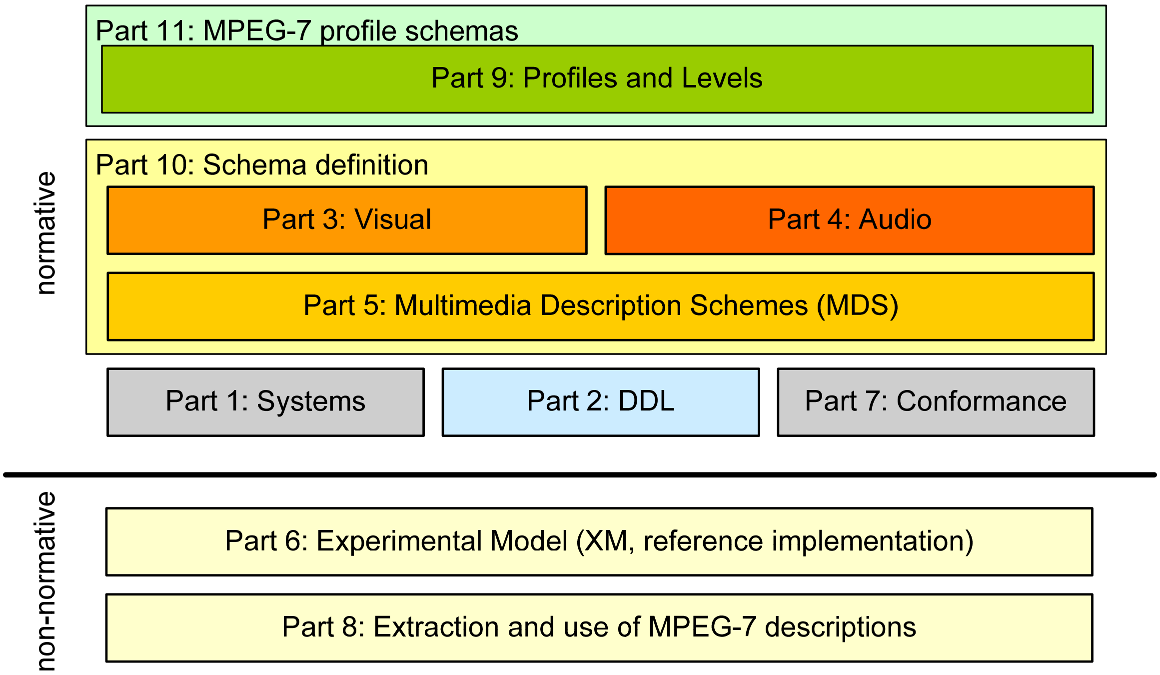

Figure 2: Structure of the MPEG-7 standard.

4.3.3 Description Tools

This section gives a brief overview of the available description tools on the MPEG-7 standard, with a special focus to those that may be relevant in a digital cinema workflow.

Global description

Descriptions on the level of whole media items:

- Technical metadata (e.g. encoding, resolution, sampling rate), similar scope as P/Meta Technical Metadata
- Creation (production) metadata (e.g. title, production place / date, rating, classification, keywords), similar scope as MXF DMS-1 Production Framework.

Summarization Tools: Allow a compact, reduced description of a media item. Summaries can be hierarchical and use clips of the original media items to represent larger segments.

Analytic description

- Decomposition into temporal segments which may be organized hierarchically (e.g. video, scene, shot). The decomposition can be done by any criteria. Depending on the criteria, a temporal segment corresponds to a clip or scene in MXF DMS-1.
- Decomposition into objects, i.e. spatial or spatiotemporal segments of the content.
- Description of low-level visual and audio features of a segment, such as dominant colours, dominant spectral components.

Speech transcription is possible on both word and phoneme level, by using the temporal segment description capabilities temporal alignment is possible. Additional information, such as recognized speakers, can be annotated.

Descriptions that can be applied both globally and to segments

- Classification of content

- Text annotations: Both free text annotation and structured annotations (e.g. lists of keywords, who/what/when/where) are supported. All descriptors that involve text are designed to be multilingual. Structured text annotations allow the use of controlled vocabularies (thesauri, ontologies)

Semantics: Allows describing objects, agents, events, concepts and their relations.

Other tools

- MPEG-7 can be extended by new descriptors under the conformance rules defined in Part 7 of the standard. This allows customizing descriptions for application specific metadata, while the rest of the description is still MPEG-7 compliant. The XML representation is defined using XML Schema, so that these extensions can be formalized as extensions of the original XML Schema.
- Collections: Allow to describe sets of related content. A collection may be organized hierarchically and consists of elements of different granularity (e.g. whole programmes and clips).
- MPEG-7 allows to describe user preferences (e.g. for search and browsing) and usage histories which can be used for personalization of access systems.

4.3.4 Limitations

- The capabilities to describe ontologies are very limited in MPEG-7, but it is possible to reference external ontologies and thesauri from within the MPEG-7 description.
- Limited capabilities to describe usage information.
- Limited capabilities to describe rights information.

Because of the richness and complexity of MPEG-7, it is necessary to agree on a subset for a certain application in order to simplify exchange of descriptions. Version 2 of MPEG-7 will introduce the concept of Profiles which are defined subsets for certain application domains.

The Dublin Core metadata standard ([Dcm2006]) was originally developed to describe electronic text documents but has later been extended to also cover audiovisual material. Focusing on simplicity, it contains a flat list of fifteen attributes belonging to three groups (content, version and intellectual property). Some of these elements can be refined using qualifiers to narrow down the meaning. This is called Qualified Dublin Core. The Scandinavian Audiovisual Metadata Group has also evaluated the Dublin Core set and has proposed a metadata set based on Qualified Dublin Core. This basic set of metadata descriptors is considered optimal for an elementary common set of descriptors that can be shared by multiple archives and is hence used in the proposed system. Moreover, Dublin Core recommends using controlled vocabularies for providing the values for these elements. The elements can be refined by using DCMI terms.

4.3.5 Software Tools

- JOANNEUM RESEARCH provides an API for MPEG-7 Part 3 (audio), Part 4 (visual) and Part 5 (multimedia description schemes) which can be used free of charge [Joa2006]. The MPEG-7 library is a set of C++ classes, implementing the MPEG-7 standard (ISO/IEC 15938:2001). With this library application developers are able to create multimedia content descriptions, manipulate them, serialize them to XML and de-serialize them – with validation – from XML. Target operating system is currently Windows; however, the source code was designed in such a way that a compilation for UNIX is possible. One major design goal was to simplify extending single classes to allow the developer to enrich interface functionality for certain descriptors. Furthermore documentation on concept and source code level improves the learning curve for the programmer. All application developers who want to deal with MPEG-7 metadata, and who do not want to struggle with complex XML DOM programming will benefit from our solution.
- The **ViTooKi** - Video-Toolkit ([ViT2006]) is an operating system independent, high-level C++ multimedia library developed to simplify the implementation of multimedia applications. Within the scope of ViTooKi classes for parsing and creating MPEG-7 files have been released under

GPL. Several libraries have been developed with the objective of describing audio content with some descriptors of the MPEG-7 standard, e.g. Java MPEG-7 Audio Encoder (http://mpeg7audioenc.sourceforge.net/), Extract Audio Spectrum Evenelope (http://sourceforge.net/projects/mpeg7ease), SoundSpotter MPEG-7 Audio Software (http://mpeg7.doc.gold.ac.uk/mirror/index.html). Implementations are available in Java, C++ or Matlab source under GPL or LGPL.

- MPEG-7. The free-use tool **VideoAnnex** ([AlphaWorks2006],) assists authors in the task of annotating video sequences with MPEG-7 metadata and automatically performs shot detection on a video, i.e. detecting the scene cuts, dissolves, and fades.

4.4 P/Meta

The EBU P/Meta ([PMe2006]) working group has designed this standard as a metadata vocabulary for programme exchange in the professional broadcast industry. It is not intended as an internal representation of a broadcaster's system but as an exchange format for programme-related information in a business-to-business use case. P/Meta consists of a number of attributes (some of them with a controlled list of values) which are organized into sets. P/Meta is technology independent; currently it can be represented in KLV (key, length, value) format or as XML.

4.5 Material Exchange Format (MXF)

The Material Exchange Format is now an established SMPTE standard (SMPTE standards 377M-394M), defining the specification of a file format for the wrapping and transport of essence and metadata in a single container. Material eXchange Format is an open binary file format targeted at the interchange of captured, ingested, finished or "almost finished" audio-visual material with associated data and metadata. It was designed and implemented with the aim of improving file-based interoperability between servers, workstations and other content devices. MXF are efficiently stored on various types of media and transported on several transportation links. MXF has been a topic in several European research projects such as Cougar, Atlantic, G-FORS, Metavision, and Nuggets. In the meantime MXF has got strong support in the industry which can be seen by a number of available software SDKs described in this short summary.

The **MXF file format** follows a common structure and defines the following basic structure. Simple MXF File looks like:

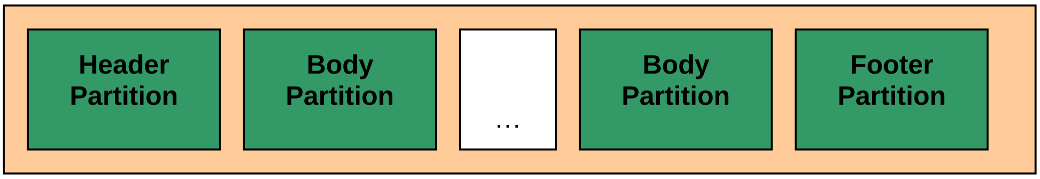

Figure 3: MXF file format

An MXF file starts with a Header Partition (comprising the header partition pack and header metadata). This followed by a Body Partition (which can be described as file body with essence container(s) and index table(s)) and is completed by a Footer Partition (which consists of a Footer Partition Pack at least). For streaming purposes, copies of the header metadata can be spread out in several body partitions. Metadata can be updated this way too! Finally the Footer Partition can embed the completed metadata.

MXF files use **key-length-value (KLV) coding**. This mechanism supports the encapsulation of the individual elements. KLV-coding is a data-encoding protocol (SMPTE 336M) that complies with International Standards Organization rules for Object Identifier data and SMPTE Universal Label (SMPTE 298M). MXF decoders can ignore information if the key (K) of a KLV triple is unknown and can jump to the next triple using the length information (L) of the current value object (V). However, decoders may optionally report such keys to the user for diagnostic purposes.

MXF files are intended for sequential writing and for sequential and random access reading. The MXF standard specifies an MXF file format that is readily convertible to a common streaming format with low overhead and without loss of data.

MXF and AAF share a **common data model**. The same data model for the representation of timelines, description of essence, and metadata is used. The main difference is that MXF does not include transition and layering functionality. Raw essence (i.e. original source material) wrapped in MXF can be post produced in AAF and the final content can be generated as a MXF file again.

4.5.1 Metadata Support

This section describes shortly different types of metadata in a MXF file. There are different types of metadata: structural, descriptive, and dark metadata.

Structural Metadata: This type of metadata describes the capabilities and the structure of the file. For example this describes the synchronisation of different tracks. It can also describe the picture size, rate, aspect ratio and other essence description parameters. The MXF specification defines the structural metadata schema and metadata sets as a single schema.

Descriptive Metadata: Content production facilities have methods for handling descriptive metadata which describe the editorial aspects of the file. The audio-visual content exchange moves from a tape-based to file-based operations which require a method to embed descriptive metadata in the file itself. Detailed information about the descriptive metadata user requirement and implementation guidelines can be found in SMPTE EG-42 document [Jon2000].

User requirements for the descriptive metadata are:

- Must be easy to understand & apply and standardized
- Low implementation overhead
- Must be open
- Must provide identification of the metadata scheme
- Must provide for normative template
- Must be extensible in header and body
- Scalability (small file/single frame to large file)
- Must provide synchronisation for multiple essence types e.g. audio, video and data essence and certain metadata
- Must be usable on major platforms and/or operating systems
- Must be transport and storage mechanism independent
- Simple and complex template (backward-forward compatibility)
- Extensibility to include non-predefined data (e.g. dark metadata)

This standard is the specification of a collection of descriptive metadata sets (called frameworks) suitable for being plugged into an MXF file with the purpose of describing the material at different levels of granularity (Programme, Scene, Clip). **A very simple plug-in mechanism** is described in the document SMPTE 377M (see [Smp2006b]). Such a metadata plug-in mechanism is a strong user requirement. It is clear that no single metadata schema will be appropriate for each application purpose. Therefore it is essential that there are possibilities to integrate new schemas without a re-development of the applications itself.

The MXF format includes a standardized mechanism to link descriptive metadata to individual essences tracks, a defined group of essence tracks or all the essence tracks. This mechanism describes the start and stop points along the essence track(s) where the descriptive metadata applies.

The description of the descriptive metadata can be found in the structural header metadata. Each item has associated with it a descriptor of the essence. Descriptive metadata tracks may be related to a timeline, an event or a static track.

Descriptive metadata schemes are identified by **universal labels** in the value field of the DM schemas batch properties. This is recorded in the preface set of the structural metadata. In this ways decoders can detect the existence of descriptive metadata on an early stage and can react, if it is a known schema.

This mechanism allows different metadata sets. It is possible that an MXF file may contain several descriptive metadata (DM) schemas. These schemas may or may be not a MXF DM schema. There are some **constraints for descriptive metadata** schemas e.g. the schema is defined by a single-inheritance class hierarchy or a SMPTE UL is defined and it is present it the preface of the structural metadata.

The MXF-DMS-1 (formerly know as Geneva Scheme) is the most important schema in this context. In the DMS-1specification, several parties agreed upon the terms „production“, „clip“, and „scene“ after long discussions (see [Smp2004a], p168).

The terms can be summarized as follows:

- Production information provides identification, or a label, and other type of metadata. It is connected with the entire A/V content as a production entity
- Clip information: This term describes the information from the creation or capture aspect. The data is most likely to be automatically collected and recorded at the point of capture and creation (e.g. metadata of the camera, GPS information etc.)
- Scene information provides metadata to describe actions and events in an editorial context (e.g. location). Editorial staff usually creates the data for logging purposes.

Dark Metadata: This type of metadata describes metadata which are unknown by an application. It can be for example metadata which are privately defined and generated.

The processing of dark metadata depends on the application. MXF decoders can ignore the local tags, but they should preserve it in the file for other applications and processes which require on a later stage in the workflow.

4.5.2 Software Tools

This section gives a short overview about MXF products and toolkits. Key functions for MXF tools are: Able to read (parse and de-multiplex) and write (create and update) MXF content, handle MXF and KLV elements, support of a plug-in mechanism for future extension and independent storage on different medias (files, cartridges, network)

Available tools are:

- MOG Solutions ([Mog2006]) offers a diverse and powerful family of MXF tools that enable accelerated development and integration of the MXF format in multimedia content applications. The MXF::SDK was developed in co-operation with the Institut für Rundfunktechnik GmbH ([IR2006]).

  For end-users:

  - “theScribe” is a full-featured MXF creation and metadata editing tool. It enables you to create new MXF files with full metadata descriptions.
  - MXFExploer is a full-featured MXF browser tool. It enables browsing of your MXF files, playing the content and navigating through the metadata.

For developers:

- The MXFComponentSuite is a set of MXF-enabled software components with a flexible API and extensive support for multiple OS and programming environments. These MXF components expose simple programming interfaces, designed to specifically target the most common MXF use cases (e.g. read, generate MXF file, add or update descriptive metadata, and play MXF files in Windows Media Player).
- The MXF::SDK is a software development kit that implements the SMPTE MXF standard. With MXF::SDK you get a robust implementation and full professional support. The MXF::SDK is a robust C++ implementation of an easy to use API, with detailed interface documentation. The toolkit provides a plug-in mechanism for new storage media, new essence type, and new operation patterns.

- OpenCube ([Cub2006]).
    - Playback all your MXF files with the OpenCube's MXFTk Reader and experience really fast access to MXF Dv, Mpeg2, Mpeg4 content.
    - MXFTk® Standard v1.2 reads and writes any type of essence into MXF-OP1a in the easiest and quickest way. Descriptive metadata are not supported in this version. MXFTk®
    - Advanced v1.2 reads and writes MXF OP1a and OPAtom, and any upper OPs as they are released. It combines most of the MXF capabilities with special mention for its full descriptive metadata support, native access directly into MXF files and MXF file streaming.
    - FreeMXF ([Fre2006]). Main freeMXF.org projects consists of several open source based MXF tools:
- MXFLib:
    - A multi-platform C++ library for reading and writing MXF files. MXFLib is an Open Source project and can be found of SourceForge at http://sourceforge.net/projects/mxflib. It is platform independent. It supports: Windows, MacOS X, SUSE Linux, Redhat Linux, FreeBSD and Solaris.
    - KLVLib: A very lightweight C library for reading and writing MXF files. KLVLib is an Open Source project and can be found of SourceForge at http://sourceforge.net/projects/klvlib. It is designed for use in applications where minimal processing power is available.
    - MXF-GUI is a simple GUI front-end for the main MXFLib command-line utilities.
- MetaGlue ([Glu2006]): Diffuser is a reference and management tool for several type of metadata (e.g. private, public, static, and dynamic form). Diffuser enables the comparison and management of public and private dictionaries and solves the problem of managing dynamic metadata. Diffuser supports MXF with MXFixer™ which is a tool to manipulate MXF Files. The tool works with both binary and XML representations of the file. MXFixer plugs in to the Diffuser system for integration of metadata.

4.6 SMIL

The Synchronizes Multimedia Integration Language (SMIL, pronounced “smile”) [SMIL2006] is an XML-based standardized markup-language for synchronized multimedia objects developed by the World Wide Web Consortium (W3C).

SMIL enables to create multimedia presentations in a text editor without any programming. As a simple example, SMIL can be used to display an image after a video is played. SMIL-objects can communicate with JAVA Servlets, Applets or CGI scripts

The main features of SMIL are:

- The presentation is created from one or more components that all are accessible via URIs, e.g. they may be accessible via a Web server.

- SMIL supports Familiar looking control buttons such as stop, fast-forward and rewind allow the user to interrupt the presentation and to move forwards or backwards to another point in the presentation.
- The individual component may be of different media type (image, video, audio, text). Start and end time of these components are given in relation to times in other objects or event therein. The begin and the end time of different components are specified according to events in other media components. For example, in a slide show, a slide is shown when the narrator in the audio track begins talking about it.
- Media composition and transformation capabilities include: visual transparency (alpha-blending), visual deformations, colour effects, chroma-key effects, audio track manipulation, sound effects, sound level control
- ”Random access”, e.g. the presentation can be started at any point or could also be shown in ”slow motion”, ,i.e. running at half its predefined speed.
- Hyperlinks can be embedded in the presentation and the user can follow these while the presentation is shown.

4.7 AAF (Advanced Authoring Format)

AAF (Advanced Authoring Format, [Aaf2006]) describes standardized metadata definitions and a file format that are used to exchange metadata and essence between creative content workstations.

AAF metadata can be seen as a super-EDL, but this metadata format can contain much more information than that description implies. Nevertheless, this open standard "format" has been created primarily for postproduction use.

An AAF file can contain the essence itself along with the metadata or can contain only the metadata referencing the essence. In order to reference material internally or externally AAF assigns to each peace of material unique identifiers. They identify the metadata and essence parts globally uniquely. Like MXF, AAF uses Unique Material Identifier (UMID) defined in SMPTE 330M for this purpose. The advantage of an SMPTE UMID is that it may be generated locally but yet is globally unique. Whenever AAF is referencing material, it does it by using this unique identifier. Tape names or file names may be included in the AAF metadata but are not used for identification. With the KLV (Key-Length-Value) mechanism, as known from MXF, arbitrary data can be stored.

There are some built-in codecs for RGBA and YUV (JPEG) picture essences and WAVE, AIFC and BWF for audio essences.

Timelines can consist of several tracks, each with transitions and operators between clips inside a track and between tracks. Several timelines with varying priorities are possible.

AAF is still under development but it can be used as exchange format for project files of editing applications for example.

A Protocol is a set of specifications for using AAF in particular application domains. The protocol defines rules how AAF files should be created or read. The Edit Protocol is intended to meet the requirements of audio-visual editorial interchange applications supporting interchange of metadata that describes edit decisions, audio and visual effects.

The AAF Association provides an open-source software reference implementation of the AAF specification also known as the AAF SDK. The SDK is available on several common computer platforms and is available from the development web site http://aaf.sourceforge.net.

The current API is based on the Component Object Model (COM). Included in the examples section of the SDK is a C++ wrapper for the API which eases development. Yet it does not cover the whole API.

Microsoft Structured Storage is used as a storage container, which has several limitations; other storage containers from SchemaSoft or the GNOME project are considered as a replacement.

At the moment the AAF-X specification is under way which will define the AAF storage format as XML.

For the future it is planned that the AAF SDK can be used to read and write MXF files.

4.8 Extensible Metadata Platform (XMP)

XMP ([Ado2006a]) is a means and format for expressing and embedding metadata in various multimedia file formats. XMP was developed by Adobe and is supported across Adobe's product range. Adobe also offers an open-source cross-platform Software Development Kit for software developers who want to work with XMP.

XMP uses RDF (The W3C's Resource Description Framework, see http://www.w3.org/TR/rdf-primer/) as data model. In brief, RDF makes it possible to make statements about a resource – in XMP this is typically the file the XMP data is embedded in. The statements that can be made are collected in *schemas* (roughly analogous to XML namespaces).

The XMP specification defines the following schemas:

- Dublin Core Schema
- XMP Basic Schema
- XMP Rights Management Schema
- XMP Media Management Schema
- XMP Basic Job Ticket Schema
- XMP Paged-Text Schema
- XMP Dynamic Media Schema

XMP was designed to be easily extensible, particularly for the addition of custom schemas. XMP supports extension by addition of standard schemas, by the addition of private or application specific schemas, and by updating schemas to new versions.

XMP data is normally embedded into the media file. The XMP specification describes that process for various file formats, among them TIFF, JPEG, GIF, PNG, HTML, PDF, SVG/XML, Postscript/EPS and DNG. There is also the possibility of storing the XMP data alongside the essence file, in a "sidecar" XMP file.

4.8.1 Tools

Available tools are:

- The XMP Software Development Kit, a C++ toolkit to enable application programmers to implement XMP support in applications (see [Ado2006b]).
- The full list of Adobe products currently supporting XMP is at [Ado2006c].
- MetaGrove by Pound Hill Software features an XMP schema and dialog editor as well as plug-ins for Adobe products and Quark Xpress ([Pou2006]).
- The Perl module Image::Exiftool has read/write support for XMP data in images [Exi2006]).

4.9 Semantic Multimedia Description Techniques

There are currently many approaches aiming to design and build ontologies for different purposes but all regarding multimedia content: First there are ontologies modelling and describing the content structure, second there are specific domain ontologies and third there are core or upper level ontologies. The following section provides an overview about different approaches and projects.

4.9.1 (Multimedia) Core Foundational Ontologies

The role of core ontologies is to serve as a starting point for the construction of new ontologies, to provide a reference point for comparisons among different ontological approaches and to serve as a bridge between existing ontologies. Core ontologies are typically conceptualizations that contain specifications of domain independent concepts and relations based on formal principles derived from philosophy, mathematics, linguistics and psychology.

ABC

The ABC ontology was intentionally developed “as a basic model and ontology that provides the notional basis for developing domain, role, or community specific ontologies, the ABC model incorporates a number of basic entities and relationships common across other metadata ontologies including time and object modification, agency, places, concepts, and tangible objects.”. [Lag2001]. It has been designed “to model physical, digital and analogue objects held in libraries, archives, and museums and on the Internet. This includes objects of all media types – text, image, video, audio, web pages, and multimedia. It can also be used to model abstract concepts such as intellectual content and temporal entities such as performances or lifecycle events that happen to an object. In addition the model can be used to describe other fundamental entities that occur across many domains such as: agents (people, organizations, instruments), places and times.” [Lag2001]

CIDOC-CRM

CIDOC CRM [Doe2003] is a core ontology for the semantic integration of cultural information. It aims at enabling information exchange and integration between heterogeneous sources of cultural heritage information. It concentrates on the definition of relationships in order to capture the underlying semantics of multiple data and metadata structures. Jane Hunter did already merge CIDOC-CRM with MPEG-7 and Jane Hunter and others did a merge of the ABC [Lag2001] ontology with CIDOC-CRM [Hun2002].

DOLCE

The Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) is an upper level ontology that provides a cognitive basis for ontology development in the sense that it aims at capturing the ontological categories underlying natural language and human commonsense [Gan2003].

WordNet

WordNet can – in a broader sense - be seen as a foundational ontology. WordNet (http://wordnet.princeton.edu) is an online lexical reference system (or a semantic lexicon) whose design is inspired by current psycholinguistic theories of human lexical memory. WordNet groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets.

General purpose ontologies are often based on WordNet and concepts from domain ontologies or other ontologies are often linked to concepts of these general purpose ontologies.

The default language of WordNet is English but there exist German, French or Spanish versions, too.

A working group at the W3C recently started to develop a RDF/OWL representation of the whole Wordnet lexicon [Ass2006].

4.9.2 Multimedia Upper Ontologies

Upper Level Ontologies are intended for more general use and describe higher level concepts that can be refined by domain ontologies

Enhanced Multimedia Meta Objects (EMMO)

EMMOs [Sch2003] encapsulate meaningful relationships between multimedia objects and maps them into a navigable structure. An EMMO is a self-contained unit of multimedia content that encompasses three aspects:

- The media aspect: An EMMO aggregates the media objects of which the multimedia content consists.
- The semantic aspect: An EMMO further encapsulates semantic associations between its contained media objects by means of a graph-based model similar to conceptual graphs. Hence, an EMMO constitutes a unit of expert knowledge concerning the multimedia content.
- The functional aspect: An EMMO offers operations for dealing with its content which applications can invoke. [Sch2003]

The encapsulated media objects are described using MPEG-7.

Knowledge Content Objects (KCO)

KCOs are based on the DOLCE foundational ontology and have so-called semantic facets that form modular entities to describe the properties of KCOs, including the 'raw' content object or media item, metadata and knowledge specific to the content object and knowledge about the topics of the content (its meaning). Through these facets all the information is modelled, that is necessary to deal with KCOs in different situations. [Beh2005]

This semantic information that is included in a KCO

- Content description: Information on how to access the content (Content Format / Encoding, Storage Location…)
- Propositional description: Information on what the content means or is about (Semantic Description and Content Classification)
- Presentation description: How to present the content (Rendering, Interaction…)
- Community description: How the content is typically used (Plans, tasks… defined by the community)
- Business description: How to trade the content (Negotiation protocol, price…)
- Trust and security description: How to protect the content
- Self description: What is the structure of the KCO (active facets, ontologies used…)

SWeMPs ontology

The SWeMPs framework/ontology is a conceptual model to express the multimedia generation task.

ZyX ontology

The ZyX ontology provides an ontological description of an abstract multimedia presentation model and is based on the ZyX model by Boll/Klas [Bol2000].

The XyZ model describes complete or fragments of multimedia documents by the means of a tree. The nodes of the tree are called presentation elements. Each presentation element has got a binding point associated with it. Such a binding point can be bound to one variable of another presentation element, thus creating the edges of the tree. The presentation elements are the generic elements of the model. They can represent atomic media elements (e.g., videos, images and text) or operator elements which combine presentation elements with certain semantics. There are operator elements that allow for temporal synchronization, definition of interaction, adaptation, and for the spatial, audible, and visible layout (the so-called projector elements) of the document.

4.9.3 Multimedia Ontologies Describing the Content and Semantic Structure

In the area of media semantics/intelligent content handling, the last few years of research in the domain of MPEG-7 and ontology languages (e.g. OWL, RDF-S, WSML) have produced notable results. Starting with a critical review of both MPEG-7 and OWL (as well as an analysis of their respective interoperability) in [OSSENETAL2003], we find there is some freedom in choosing the proper representation technique for any purpose. In the area of media semantics/intelligent content handling, the last few years of research in the domain of MPEG-7 and ontologies languages (e.g. OWL, RDF-S) have produced notable results. The following approaches may be taken to handle smart media content:

- The purist approach, where either an MPEG-7- or an OWL-based environment is assumed to fulfil the task.
- The integration approach which tries to embed one representation technique into the other [Hun2005].
- The layer approach, or “principle of subsidiarity” which uses each vocabulary in its appropriate realm. [Tron2003].
- Finally, it is possible to invent new representation techniques – as proposed in [HUJIAN1999].

In this case it is important to differentiate between rigid syntactical representation and knowledge-based representation. The invention of new syntactical representation formats (e.g. XML-based serializations) would lose the advantages of existing standards without necessarily gaining expressiveness. On the other hand, there is still significant development in the area of standardized ontology languages (e.g. OWL vs. WSML), because of several severe weaknesses.

MPEG-7 Upper MDS Ontology by Hunter

One of the first approaches to create an MPEG-7 ontology was initiated by Jane Hunter from the University of Queensland [Hun2005], firstly developed in RDFS, then converted into DAML+OIL, and recently also available in OWL. The ontology covers the upper part of the Multimedia Description Scheme (MDS) part of the MPEG-7 standard. It consists in about 60 classes and 40 properties. This is an OWL Full ontology.

MPEG-7 MDS Ontology by Tsinaraki

Based on Jane Hunter's work Tsinaraki et al from the University of Crete/Greece created another extended MPEG-7 ontology which covers the full Multimedia Description Scheme (MDS) part of the MPEG-7 standard [Tsi2004]. This work describes a software engineering framework aiming to facilitate knowledge-based MPEG-7 multimedia application development. It consists of an ontological infrastructure (following the MPEG-7 and TV-Anytime way of describing AV content), methodologies for interoperability between MPEG-7 and OWL and metadata in an underlying MPEG-7 repository. The methodology of OWL ontology integration and tools are based on

- The core OWL ontology covers the semantic part of the MPEG-7 MDS,
- Definition of domain specific ontology that extend the core ontology, and
- Set of rules to transform the semantic metadata to MPEG-7 and TV-Anytime compliant XML documents.

It contains 420 classes and 175 properties and is formulated in as OWL DL ontology

MPEG-7 Ontology by DMAG

Another notable effort was recently made by the DMAG (Distributed Multimedia Applications Group) from Universitat Pempeu Fabra in Barcelona/Spain. Their aim was to completely move MPEG-7 to Semantic Web by making its internal XML Schema semantics explicitly visible in form of an OWL-document, primarily to extend MPEG-7's flexibility. According to the authors [Gar2005], the result is an automatically generated MPEG-7 Ontology which is able to cover all internal semantics of the MPEG-7 standard (1182 elements, 417 attributes, 377 complex types) and results in additional reasoning capabilities for multimedia management.

INA Ontology

This ontology is not really an MPEG-7 ontology since it does not cover the whole standard. It is rather a core audio-visual ontology inspired by several terminologies, either standardized (like MPEG-7 and TV Anytime) or still under development (ProgramGuideML). Furthermore, this ontology – represented in OWL Full – strongly benefits from the practices of the French INA institute, the English BBC and the

Italian RAI channels which have also developed a complete terminology for describing radio and TV programs [Tro2003].

4.9.4 Multimedia Ontologies from EU and National Project Efforts

ACEMedia

ACEMedia extends and enriches ontologies to include low level audiovisual features, descriptors and behavioural models in order to support automatic annotation; they developed a core ontology based on extensions of the DOLCE core ontology and the multimedia-specific infrastructure components, the Visual Descriptor Ontology, based on an RDFS representation of the MPEG-7 Visual Descriptors and the Multimedia Structure Ontology based on MPEG-7 MDS. Its main aims are the support of audio-visual content analysis and object/event recognition, the creation of knowledge beyond object and scene recognition through reasoning processes and at enabling user-friendly and intelligent search and retrieval.

AIM@SHAPE

AIM@SHAPE focuses on the development of ontologies for the description of resources relevant in the context of Computer Graphics and Vision derived by ontologies. They develop a common ontology that includes features of all their modelled domains and several domain ontologies for the modelling of (mainly) Virtual Humans, Product Design and Acquisition and Reconstruction Processes. The knowledge these ontologies all share is about shape models, structures and tools. They use OWL and MPEG-7.

BOEMIE

BOEMIE tries to combine multimedia extraction and ontology evolution in a bootstrapping process involving extraction of semantic information from multimedia content in order to populate and enrich the ontologies. The project uses MPEG-7 for the description of multimedia content and its properties, domain specific ontologies and a geographic information ontology. The project didn't decide at present which ontology language to use and consider using DOLCE as a starting point for the conceptualization of spatio-temporal relations.

K-Space

K-Space (Knowledge Space of Shared Technology and Integrative Research to Bridge the Semantic Gap) is a network of excellence of research teams from academia and industry to conduct integrative research and dissemination activities in semantic inference for both automatic and semi-automatic annotation and retrieval of multimedia content. A goal is to close the gap between the low-level and high level content descriptions.

K-Space focuses on three research areas:

- **Content-based multimedia analysis:** Tools and methodologies for low-level signal processing, object segmentation, audio/speech processing and text analysis, and audiovisual content structuring and description.
- **Knowledge extraction:** Specification of a multimedia ontology infrastructure, knowledge acquisition from multimedia content, knowledge-assisted multimedia analysis, and context based multimedia mining and intelligent exploitation of user relevance feedback.
- **Semantic multimedia:** knowledge representation for multimedia (e.g. extension of RDF, OWL, SWRL and reasoning with extended standards), distributed semantic management of multimedia data (e.g. storage of the metadata and a distributed version of the metadata store, collaborative distributed reference framework), semantics-based interaction with multimedia and multimodal media analysis (e.g. semantic retrieval and presentation of multimedia content).

Mammography ontology

The aim of this ontology is to support the formalization of knowledge mainly to reduce the subjectivism of the description of mammography images. They use a domain ontology based on Breast Image

Reporting and Data System (BI-RADS) and the ACEMEDIA Visual Descriptor Ontology. Both ontologies are being developed in OWL-DL.

METOKIS

Metokis provided a deeper understanding of 'knowledge types' and thus contributed to the definition of self-describing knowledge- and media content objects. One of the main results of METOKIS was the development of a model for knowledge content objects (KCOs) that is based on the DOLCE foundational ontology.

MediaCampaign

The main goal of MEPCO is the cross-relation of media campaigns over the media TV, press and Internet and furthermore the ambitious goal to cross link media campaigns also over different countries. What makes a media campaign unique from others is not completely straight forward; however, there are rules how a human can determine whether a media campaign is new. These heuristic rules will be formally encoded as to describe media campaigns in a generic way. The MediaCampaign Ontology (MEPCO) will be based on the upper-level ontology PROTON and will be aligned to media-related standards. It provides consistent formal definitions of about 300 general concepts and 100 relations and attributes. MEPCO will also be aligned with existing standards for media-related metadata, such as NewsML and News Codes from IPTC.

MUSCLE NOE

The muscle network of excellence works on the definition of an ontology for multimedia (primarily image) understanding. They want to make use of MPEG-7.

NM2

NM2 is developing compelling new media forms which take advantage of the unique characteristics of broadband networks. To achieve this goal NM2 will create tools for the cost efficient production of non-linear interactive narratives that are integrated with emerging end-to-end digital production environments. As the NM2 system will be able to create personalised versions of narratives on an individual basis, it has to understand the content and the structure of the narrative. Describing intrinsic low-level features ranging from shape, colour, as well as shot-border, key-frames, to activity region, and object/face tracking in NM2 is done using MPEG-7. The ontology layer comprises a core ontology formalising all generic concepts and relations and production-specific ontologies are defined per production to capture concepts, relations, and instances depending on the domain of the production (news item, historical element, etc.). To define ontologies, add contextual information and embed the below described Narrative Structure Language, OWL is utilised. The Narrative Structure Language (developed within NM2) is a language for expressing non-linear narratives. Global narratives contain references to media objects and specify rules that are used to create a specific narrative on-the-fly based on context information. A specific narrative is a set of representations of media objects arranged into a play-list that is delivered to a NM2 end-user, hence is an instantiation of a global narrative.

4.9.5 Multimedia Annotation Tools

This section lists semantic image and video annotation tools. Most references and descriptions of image annotation tools are taken from [SWB2006].

Semantic Image Annotation Tools

- flickr2rdf [Flickr2RDF2006]: flickr2rdf is a web-based service which uses the Flickr API [FlickrAPI2006] to extract metadata from Flickr's photo repository [FlickrPhoto2006] As result it is able to generate adequate RDF descriptions. Flickr is a, well known, major an online photo management and sharing tool, where users can upload as also annotate their photos with simple keywords.
- JpegRDF [JPegRDF2006]: jpegRDF is an open source tool by Norman Walsh, written in Java. On the one hand, jpegRDF is a simple editor for the RDF metadata which is stored in the comment section of JPEG image files. On the other hand, it can also be used for retrieving purposes. Another application of jpegRDF concerns the conversion from EXIF to RDF.

- M-OntoMat-Annotizer [MontoAnn2006] (M stands for Multimedia) has been initially developed within the ACEMEDIA project. The idea behind this very user-friendly tool is to allow the semantic annotation of images and videos for multimedia analysis and retrieval. M-OntoMat-Annotizer is based on the CREAM (CREAting Metadata for the Semantic Web) framework [Han2002]. The core component, a Visual Descriptor Extraction Tool (VDE) was developed as a plug-in to the OntoMat-Annotizer. The purpose of the plug-in is to extension of OntoMat-Annotizer's capabilities and supporting the initialization and linking of RDF(S) domain ontologies with low-level MPEG-7 visual descriptors.
- PhotoStuff [PhotoStuff2006]: This is an image annotation tool for the Semantic Web. It allows annotating images and image parts according to several OWL ontologies of any domain. Another application of PhotoStuff concerns the conversion of JPEG internal metadata to RDF. The resulting annotations can then be published and shared on the Web.
- SWAD [SWAD2006]: This tool is written in JavaScript and uses the RESTful web services to access remote information. It is designed to be a quick and easy means of creating structured information about images, including who or what is depicted in the image; where and when it was created; creator and licensing information. The aim is to create and enable the reuse of alternative formats for both text and images for use in an accessibility context, although the potential application is much wider
- SCHEMA [SCHEMA2006] This tool integrates a non-normative part of the MPEG-7 standard and the MPEG-7 XM software, for extracting, coding and storing in the database standardized descriptors based on the output of the analysis modules. The system can also support high level (semantic) descriptors and the integration of visual media indexing and retrieval with other modalities (like text and audio based indexing and retrieval).
- AKTive Media [AktiveMedia2006]: AKtive Media provides an ontology based multimedia annotation system. It aims to automate the process of annotation by presenting interactive suggesting knowledge, while the user-driven annotating process. The whole system actively works in the background, interacting with web services and queries a central annotational store to look for context specific knowledge.

Semantic Audio/Video Annotation Tools

- Advene [Advene2006]: Advene (Annotate Digital Video, Exchange on the NEt) is an ongoing project in the LIRIS laboratory (UMR 5205 CNRS) at University Claude Bernard Lyon 1. It aims at providing a model and a format to share annotations about digital video documents (movies, courses, conferences...), as well as tools to edit and visualize the hypervideos generated from both the annotation and the audiovisual document.
- AlphaWorks [AlphaWorks2006], IBM MPEG-7 Annotation Tool. The IBM MPEG-7 Annotation Tool can be used to simplify the process of annotation of MPEG video data. The tool allows the user to describe each shot with static scene descriptions, key object descriptions, event descriptions and other lexicon sets. All annotations are stored as MPEG-7 descriptions in an XML file.

5 Research Objectives and Innovation

The previous sections have outlined the state-of-the-art in multimedia description standards and semantic technologies. These base technologies serve as a starting point for the research work to be carried out within the SALERO project.

This section provides a short summarization of key research objectives which use or will use multimedia ontologies, especially for the annotation and search & retrieval tasks within a production and post production environment.

5.1 Motivation

Our goal is to describe multimedia objects in a way that makes them usable for a broad field of software and different areas of application.

The challenge according to the goal of generalization is that among most of the standards no compatibility is guaranteed since most of them are implemented in a narrow domain where the syntax and structure is specific to a particular organisation [Jef2006]. If we want to make use of existing standards a high level description has to be developed to bridge the semantic gap between them. A feasible way of a generalized description is the use of an ontology [Hov2003].

This chapter addresses several different duties and responsibilities of semantic multimedia description techniques:

1. Create a multi-purpose description
2. Identify the language of description and so have control over further processing like importing and exporting metadata from and into a variety of standards
3. Mapping and translating from low level features to concepts of the domain ontology

5.2 Objectives and Innovation

We aim to create a metadata description schema that can be used for high-level features, production-specific features and low-level features alike. Challenges include:

- Mapping the various existing metadata formats into this schema.
- Respecting and preserving semantic differences between similar features in the existing metadata formats, while still unifying these features as far as possible in our description schema.

One result of the project will be to show the feasibility of using ontologies for this purpose (**unifying disparate metadata formats**). Using ontologies is a promising approach because of their modelling power, formal background and strong semantics compared to ad-hoc and informally specified data models. We try to use the arising multimedia ontology for enhancing metadata integration and as a further step improve the reusability of Multimedia Objects (MMOs).

Another advantage of ontologies could be the **semantic search** (e.g. enhanced recall and precision of search results for multimedia objects). SALERO's WP05 "Context Based Information Retrieval" will show the value of ontologies for that purpose. Semantic annotated content enables indexing all the content/multimedia objects of the repositories and perform requests which combine structured queries, reasoning, and IR. Ontologies are used in the retrieval process and to present the result. For example: The multimedia objects should be open for context-sensitive retrieval ('given the sound of a galloping horse, find an image of a galloping horse'). The associated contextual information should be clearer and more useful, helping the production company to find examples of 'horse' and 'horse with rider' from other productions with similar actions. From the context of the scenario, it should be possible to infer the underlying need and look for 'rider objects' created before which could be adapted for re-use.

Furthermore a concept browser can support the user in viewing the search results. It provides the user with a high-level way to explore a collection. The visual view of concepts (e.g. labelled video segments) and relationships can be useful to find similar objects. The concept browser can be designed around a tree (hierarchical structure) or graph based visualisation, where users could navigate to concepts of interests.

A recurring problem in an industrial setting is **creating metadata and keeping it up to date**. We aim to develop ontology-based ways of supporting the user with these important tasks. This support includes the implementation of a suite of ontology tools used in daily work. An important point is the integration of the ontology-enhancements into the work process with a minimum amount of disturbing the workflow.

The research tasks will focus on finding suitable representation techniques for the efficient and expressive representation of semantic aspects of media components. One major challenge is to unifying the worlds of media metadata on one hand and expressive ontology languages on the other. In detail, we will aim at applying work in the design of ontology languages that overcome limitations of the OWL family of languages for the rich representation of media semantics.

A special focus will be on **alignment of the ontology** language and the computational costs of full **reasoning support**. One must pay attention to the fact that most complex description logic reasoners do not scale well when handling huge amount of data. There are two primary reasoning tasks on the Semantic Web: subsumption and query answering. For the subsumption problem efficient description logics algorithms can be used but for query answering, the main task in SALERO, only few optimizations exist that do this task efficiently. Efficient query answering could be done for example with OWL Flight or OWL(minus) which however have less expressive power than OWL DL.

Hence, we shall evaluate what is needed in terms of expressiveness and reasoning for the specific requirements of SALERO before choosing an ontology representation language. The basic outcome of the research in WP3 will be a methodology for handling SALERO's multimedia objects with semantic representation tools and techniques. We are looking forward to the added value in using semantic methods (e.g. ontologies, rules etc.) but we will also keep other technologies in perspective if necessary.

6 References

[Aaf2006] Advanced Authoring Format. URL: http://www.aafassociation.org/. [last visited: 20/06/2006]

[Ado2006a] Adobe XMP Specification. URL: http://partners.adobe.com/public/developer/en/xmp/sdk/xmpspecification.pdf. [last visited: 30/05/2006]

[Ado2006b] Adobe XMP Software Development Kit. URL: http://partners.adobe.com/public/developer/xmp/sdk/index.html. [last visited: 30/05/2006]

[Ado2006c] Adobe XMP Partners. URL: http://www.adobeopenoptions.com/products/xmp/partners.html. [last visited: 30/05/2006]

[Advene2006] Advene: http://liris.cnrs.fr/advene/, Last access: 22/06/2006

[AktiveMedia2006] AKTive Media: http://www.dcs.shef.ac.uk/~ajay/html/cresearch.html, Last access: 22/06/2006

[AlphaWorks2006] alphaWorks IBM MPEG-7 Annotation Tool: URL: http://www.alphaworks.ibm.com/tech/videoannex, Last access: 22/06/2006

[Ass2006] Assem M., Gangemi A. and Schreiber G. (eds.) (2006) RDF/OWL representation of WordNet. W3C Editor's Draft 23 April 2006, http://www.w3.org/2001/sw/BestPractices/WNET/wn-conversion.html [last visited: 01.06.2006]

[Baa2003] F. Baader, D. Calvanese, D. L. McGuiness, D. Nardi, P. F. Patel-Schneider (2003): The Description Logic Handbook: Theory, Implementation, Applications. Cambridge University Press, Cambridge, UK, 2003.

[Bec2003] Beckett D. RDF/XML syntax speci¯cation (revised). Recommendation 10 February 2004, W3C, 2003.

[Beh2005] Behrendt, W. Gangemi, A., Maass, W., and Westenthaler, R. (2005): Towards an Ontology-Based Distributed Architecture for Paid Content. In: A. Gómez-Pérez and J. Euzenat (Eds.): ESWC 2005, LNCS 3532, pp. 257–271, 2005. Springer-Verlag Berlin Heidelberg 2005

[Ber2001] Berners-Lee T,Hendler j, and Lassila O: The Semantic Web. Scientific American, May 2001.

[Bol2000] Boll,S., Klas, W. and Westermann, U. Multimedia Document Formats - Sealed Fate or Setting Out for New Shores? In: Multimedia - Tools and Applications 11(3), August 2000, 11(3): 267-279, 2000

[Bon1998] Bonner A. J. and Kifer M. A logic for programming database trans actions. pp. 117-166, 1998.

[Bor1996] Borgida A. On the relative expressiveness of description logics and predicate logics. Artificial Intelligence, 82(1-2): pp. 353-367, 1996.

[Bri2004] Brickley D, Guha RV (2004) RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, http://www.w3.org/TR/rdf-schema/ [last visited: 20.05.2006]

[Bru2004] de Bruijn J., Polleres A., Lara R., and Fensel D. OWL Flight. Deliverable 20.3v0.1, WSML, 2004. Available from http://www.wsmo.org/2004/d20/d20.3/v0.1/.

[Bru2005a] de Bruijn J, Polleres A (eds) (2005): OWL-.WSML Deliverable D20.1 v0.2. WSML Working Draft May, 15 2005. http://www.wsmo.org/TR/d20/d20.1/v0.2/20050515/ (last visited: 20.05.2006]

[Bru2005b] de Bruijn J, Polleres A, Lara R, and Fensel D (2005), OWL DL vs. OWL Flight: Conceptual Modeling and Reasoning for the Semantic Web, Proceedings of the 14th International World Wide Web Conference (WWW2005), Chiba, Japan.

[Bru2005c] de Bruijn (ed) (2005). The Web Service Modeling Language WSML WSML Final Draft 20 March 2005 Deliverable D16.1v0.2, Available from http://www.wsmo.org/TR/d16/d16.1/v0.2/. [Last visited: 01.06.2006]

[Chen1993] Chen W., Kifer M., and Scott Warren D. (1993) HILOG: A foundation for higher-order logic programming. Journal of Logic Programming, 15(3): pp.187-230, 1993.

[Cor2000] Corcho O. and Gomez-Perez A. (2000) A Roadmap to Ontology Specification Languages. In: Dieng R., Corby O. (eds) Proceedings of the 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW'00) LNAI 1937, Springer Verlag, Berlin, Germany.

[Cub2006] OpenCube. URL: http://www.opencube.fr/. [last visited: 30/05/2006]

Cyc2002] CycCorp (2002): Ontological Engineers Handbook, http://www.cyc.com/doc/handbook/oe/oe-handbook-toc-opencyc.html [last visited: 20.05.2006]

[DCM2006] Dublin Core Metadata Initiative, URL: http://dublincore.org/. [last visited: 30/05/2006]

[Dea2004] Dean M, Schreiber G (eds) (2004) OWL Web Ontology Language Reference. W3C Recommendation. http://www.w3.org/TR/owl-ref/ [last visited: 20.05.2006]

[Den2004] Denny M. (2004): Ontology Tools Survey, Revisited, 2004. available online: http://www.xml.com/pub/a/2004/07/14/onto.html [last visited: 01.06.2006]

[Doe2003] Doerr, M. (2003) The CIDOC CRM-an Ontological Approach to Semantic Interoperability of Metadata AI Magazine-Special Issue on Ontologies, 2003

[Eclipse2006] Eclipse project. URL: http://www.eclipse.org/org/#about, [Last visited: 23/06/2006]

[Eit1997] Eiter T., Gottlob G., and Mannila H. Disjunctive Datalog. ACM Transactions on Database Systems, 22(3): pp. 364-418, 1997.

[Exi2006] ExifTool (by P. Harvey). URL: http://www.sno.phy.queensu.ca/~phil/exiftool/. [last visited: 30/05/2006]

[Fen2001] Fensel D, van Harmelen F, Horrocks I, McGuinness DL, Patel-Schneider PF (2001) OIL: An ontology infrastructure for the Semantic Web. IEEE Intelligent Systems & their applications 16(2): pp. 38-44

[Fer1997] Fernandez-Lopez M., Gomez-Perez A. and Juristo N. (1997) METHONTOLOGY: from Ontological Art Towards Ontological Engineering. Spring Symposium on Ontolological Engineering of AAAI, 1997.

[Flickr2RDF2006] Flickr2RDF. URL: http://purl.org/net/kanzaki/flickr2rdf/, Last access: 22/06/2006

[FlickrAPI2006] Flickr API. URL: http://www.flickr.com/services/api/, Last access: 22/06/2006

[FlickrPhoto2006] Flickr Photo Repository. URL: http://www.flickr.com/, Last access: 22/06/2006

[Fre2006] Free MXF. URL: http://www.freemxf.org/. [last visited: 30/05/2006]Elsevier. Focal Press. 2004.

[Gan2002] Gangemi, A.; Guarino, N.; Masolo, C.; Oltramari, A. & Schneider, L. (2002) Sweetening ontologies with DOLCE. In: Proceedings of 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), pp. 166 ff, 2002.

[Gar2005] Garcia, R.; Celma, O. (2005): Semantic Integration and Retrieval of Multimedia Metadata; Distributed Multimedia Applications Group (DMAG), Universitat Pompeu Fabra

[Gen1992] Genesereth M. R. and Fikes R. E. (1992) Knowledge Interchange Format. Version 3.0. Reference Manual. Technical Report Logic-92-1. Computer Science Department. Stanford University, California.

[Gil2004] Ed. Brad Gilmer. File Interchange Handbook for images, audio and metadata.

[Glu2006] MetaGlue 2006. MetaGlue. URL: http://www.metaglue.com. [last visited: 30/05/2006]

[Gom2002] Gomez-Perez A. (2002): A Survey on Ontology Tools, OntoWeb deliverable D1.3, http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D13_v1-0.zip

[Gom2004] Gomez-Perez A, Fernandez-Lopez M. And Corcho O. (2004) Ontological Engineering, Springer, 2004.

[Gru1992] gruber, T.R. (1992): Ontolingua: A mechanism to support portable ontologies. Technical Report KSL-91-66, Knowledge Systems Laboratory, Stanford Universiy, California.

[Gru1993] Gruber, T.R. (1993): A translation approach to portable ontology specification. Knowledge Acquistion 5(2) pp. 199-200.

[Gru1995] Grüninger M. and Fox M. S. (1995) Methodology for the design and evaluation of ontologes In Skuce D. (ed) IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, 1995

[Han2002] Handschuh Siegfried, Staab, Steffen, and Maedche, Alexander CREAting relational Metadata (CREAM) -- a framework for semantic annotation In I. Cruz, S. Decker, J. Euzenat (eds.), The Emerging Semantic Web. IOS Press, Amsterdam, 2002.

[Hop2006] Hopper R., EBU project Group P/Meta – Metadata Exchange Scheme, V1.0 EBU Technical review, 2002. URL: http://www.ebu.ch/en/technical/trev/trev_290-hopper.pdf. [last visited: 30/05/2006]

[Hor2001] Horrocks I, van Harmelen F (eds) (2001) Reference Description of the DAML+OIL (March 2001) Ontology Markup Language. Technical report. http://www.daml.org/2001/03/reference [last visited: 20.05.2006]

[Hor2003a] Horrocks I. and Patel-Schneider P. F. Reducing OWL entailment to description logic satisfiability. In Proc. of the 2003 International Semantic Web Conference (ISWC 2003), Sanibel Island, Florida, 2003.

[Hor2003b] Horrocks I., Patel-Schneider P. F., Boley H., Tabet S., Grosof B., and Dean M. SWRL: A semantic web rule language combining OWL and RuleML. Daml draft v0.5, DAML, 2003. Available from http://www.daml.org/2003/11/swrl/. [Last visited: 01.06.2006]

[Hov2003] Eduard Hovy, Using an ontology to simplify data access, Communications of the ACM, v.46 n.1, January 2003

[Hun2002] Hunter, J. (2002) Combining the CIDOC CRM and MPEG-7 to Describe Multimedia in Museums In Proceedings of Museums and the Web, 2002

[Hun2005] Hunter J.: Adding multimedia to the Semantic Web: Building and applying an MPEG-7 ontology: In Stamou, G and Kollias S. (eds): Multimedia Content and the Semantic Web, Wiley, 2005.

[IR2006] Institut für Rundfunktechnik. URL: http://www.irt.de/. [last visited: 30/05/2006]

[Jef2006] Keith G Jeffery, Metadata: The Future of Information Systems. World Meteorological Organization, 2000. URL: http:// wmo.ch/web/www/WDM/ET-IDM/Doc-2-3.pdf. Last access: 30/05/2006

[JPegRDF2006] jpegRDF. URL:http://sourceforge.net/projects/jpegrdf/, Last access: 22/06/2006

[Jon2000] de Jong, A.: Metadata in the audiovisual production environment. International Federation of Television Archives FIAT/IFTA

[Joa2006] Joanneum Research MPEG-7 Library. URL: http://iis.joanneum.at/MPEG-7/overview.htm. [last visited: 30/05/2006]

[Kar1999] Karp, P.D. Chaudhri, V. K. and Thomere, J. (1999) XOL: An XML-Based Ontology Exchange Language, http://www.ai.sri.com/~pkarp/xol/xol.html [last visited: 20.05.2006]

[Kif1989] Kifer, M. and Lausen, G. (1989). F-logic: a higher-order language for reasoning about objects, inheritance, and scheme. In Proceedings of the 1989 ACM SIGMOD international Conference on Management of Data. J. Clifford, B. Lindsay, and D. Maier, Eds. SIGMOD '89. ACM Press, New York, NY, pp. 134-146

[Kif1995] Kifer M., Lausen G. and Wu J. (1995) Logical Foundations of Object-Oriented and Frame-based Languages. Journal of ACM 42(4), pp. 741-843, 1995.

[Lag2001] Lagoze, C. and Hunter, J. (2001), "The ABC Ontology and Model," Journal of Digital Information, 2 (2), 2001.

[Len1990] Lenat D. B. and Guha R. V. (1990) Building Large Knowledge Based Systems: Representatoin and Inference in the Cyc Project. Addision Wesley, 1990.

[Lev1998] Levy A. Y. and Rousset M.-C. Combining horn rules and description logics in CARIN. Artificial Intelligence 104, pp. 165-209, 1998.

[Man2002] Ed. B.S.Manjunath, Philippe Salembier, Thomas Sikora, Introduction to MPEG-7, Wiley 2002

[Man2004] Manola F and Miller E (2004) RDF Primer. W3C Recommendation. http://www.w3.org/TR/REC-rdf-syntax/ [last visited: 20.05.2006]

[Mar2002] José M. Martínez, “MPEG-7 Overview (version 8)”, ISO/IEC JTC 1/SC 29/WG 11/N4980, 2002-07-22/26.

[Mar2004] Martín-Recuerda F., Harth A., Decker S., Zhdanova A., Ding Y., Stollberg M. and Arroyo S. (2004) DIP Deliverable D2.1 Report on Requirements Analysis and State of the Art, 2004. Available online: http://dip.semanticweb.org/documents/DIP-D21-v1_Public.pdf [last visited: 01.06.2006]

[McG2004] McGuiness D. and van Harmelen, F. (eds.): OWL Web Ontology Language Overview, W3C Recommendation 10 Februar 2004, http://www.w3.org/TR/owl-features/ [last visited: 01.06.2006]

[Mog2006] MOG Solution. URL: http://www.mog-solutions.com/. [last visited: 30/05/2006]

[MontoAnn2006] M-OntoMat-Annotizer. URL: http://www.acemedia.org/aceMedia/results/software/m-ontomat-annotizer.html, Last access: 22/06/2006

[Ordi2006] Ontology Representation and Data Integration (ORDI) Framework. URL: http://www.omwg.org/tools/ordi/v0.21/FactSheet.html, [Last access: 22/06/2006]

[Pan2004] Pan J. Z. and Horrocks I. OWL-E: Extending OWL with expressive datatype expressions. IMG Technical Report IMG/2004/KR-SW-01/v1.0, Victoria University of Manchester, 2004. Available from http://dl-web.man.ac.uk/Doc/IMGTR-OWL-E.pdf. [last visited: 01.06.2006]

[Par2002] Park J. and Hunting S. (2002): *XML Topic Maps: Creating and Using Topic Maps for the Web.* Addison-Wesley, 2002.

[Pat2004] Patel-Schneider P. F., Hayes P., and Horrocks I. OWL web ontology language semantics and abstract syntax. Recommendation 10 February 2004, W3C, 2004. Available from: http://www.w3.org/TR/owl-semantics/ [last visited: 01.06.2006]

[Pep2002] Pepper, S (2002): The TAO of Topic Maps, available online: http://www.ontopia.net/topicmaps/materials/tao.html [last visited: 01.06.2006]

[PhotoStuff2006] PhotoStuff. URL: http://www.mindswap.org/2003/PhotoStuff/. [last visited: 22/06/2006]

[PMe2006] P/Meta Tools. URL: http://www.crit.rai.it/eng/attivita/pmeta/pmeta_more.html. [last visited: 30/05/2006]

[Pou2006] Poundhill. URL: http://www.poundhill.com/products.html. [last visited: 30/05/2006]

[Sch2003] Schellner Karin, Westermann Gerd Utz, Zillner Sonja and Klas Wolfgang CULTOS: Towards a World-wide Digital Collection of Exchangeable Units of Multimedia Content for Intertextual Studies. In Conference of Distributed Multimedia Systems (DMS2003), 2003.

[Sta2001] Staab S., Schnurr H. P. and Sure Y. (2001) Knowledge Processes and Ontologies. IEEE Intelligent Systems 16(1), 2001.

[SCHEMA2006] SCHEMA. URL:http://www.schema-ist.org/SCHEMA/. [last visited: 22/06/2006]

[SMIL2006] SMIL specificatioin. URL: http://www.w3.org/TR/SMIL2/. [last visited: 22/06/2006]

[SWB2006] Semantic Web Best Practices Multimedia Task Force (2006): Image Annotation on the Semantic Web: Tools Overview, http://www.w3.org/2001/sw/BestPractices/MM/resources/Tools.html, 2006.

[Sur2004] Sure Y. and Tempich C. (2004) State of the Art in Ontology Engineering Methodologies Technical Report , Institute AIFB, University of Karlsruhe. November 2004. SEKT Deliverable D7.1.1.c, http://www.sekt-project.com/internal/informal-deliverables/WP07a/D7.1.1.c_v0-8.pdf

[Sur2006] Sure Y., Tempich C. and Vrandecic D. (2006) Ontology Engineering Methodologies In John Davies and Rudi Studer and Paul Warren, Semantic Web Technologies: Trends and Research in Ontology-based Systems, chapter 9. Wiley, UK, 2006. ollaboration of a multiplicity of individuals.

[SWAD2006] SWAD. URL:http://swordfish.rdfweb.org/discovery/2004/03/w3photo/annotate.html, Last access: 22/06/2006

[Tsi2004] C. Tsinaraki, P. Polydoros and S. Christodoulakis (2004)A Integration of OWL ontologies in MPEG-7 and TVAnytime compliant Semantic Indexing. In Proc. of the 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004), Riga, Latvia, June 2004.

[Tro2003] Troncy, R. (2003) Integrating Structure and Semantics into Audio-visual Documents. In: D. Fensel, K.S. & Mylopoulos, J. (ed.) The Semantic Web – Proceedings of ISWC, LNCS 2870, 2003.

[Usc1996] Uschold, M (1996) Building Ontologies: Towards a Unified Methodology In: Watson I. (ed) 16th annual Conference of the British Computer Society Specialist Group on Expert Systems. Cambdirge, UK.

[Usc1998] M. Uschold, M. King, S. Moralee & Y. Zorgios (1998): The Enterprise Ontology. Knowledge Eng. Rev. 13(1) pp. 32-89

[ViT2006] **ViTooKi,** Video-Toolkit. URL: http://vitooki.sourceforge.net/. [last visited: 30/06/2006]

7 Glossary

Terms used within the SALERO project sorted alphabetically.

AAF	Advanced Authoring Format

Closed world assumption: a logical presumption that everything that is not known to be true is false

Datalog	a query and rule language for deductive databases
DHL	Description Horn Logic (DHL) an expressive subset of First order logic
First-order logic	a system of a mathematical logic with high expressive power
Horn clause	a clause with at most one positive literal, e.g.: $\neg A \vee \neg B \vee \neg C \vee D$
MMO	Multimedia Object
MPEG-7	Multimedia Content Description Interface
MXF	Material Exchange Format
SMIL	Synchronised Multimedia Markup Language

Partner Acronyms

AM	Activa Multimedia, ES
BLITZ	Blitz Games, UK
CINESITE	Cinesite Europe Ltd., UK
DIT	Dublin Institute of Technology, IE
DTS	Digital Theatre Systems, UK
FBM-UPF	Fundació Universitat Pompeu Fabra, ES
GVG	Grass Valley Germany, DE
JRS	JOANNEUM RESEARCH Forschungsgesellschaft mbH, AT
LFUI	Leopold-Franzenzs Universtät Innsbruck, AT
PGP	Pepper's Ghost Productions Ltd., UK
TAIK	Taideteollinen Korkeakoulu, FI
URL	Universitat Ramon Llull, ES
UG	University of Glasgow, UK

