Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

INS

Information Systems

INformation Systems

Video on the Semantic Web Experiences with Media Streams

J.P.T.M. Geurts, J.R. van Ossenbruggen, H.L. Hardman, M. Davis

REPORT INS-E0404 APRIL 21, 2004

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3681

Video on the Semantic Web Experiences with Media Streams

ABSTRACT
In this paper, we report our experiences with the use of SemanticWeb technology for annotating digital video material.Web technology is used to transform a large, existing video ontology embedded in an annotation tool into a commonly accessible format. The recombination of existing video material is then used as an example application, in which the video metadata enables the retrieval of video footage based on both content descriptions and cinematographic concepts, such as establishing and reaction shots. The paper focuses on the practical issues of porting ontological information to the Semantic Web, the multimedia-specific issues of video annotation, and requirements for Semantic Web query and access patterns. It thereby explicitly aims at providing input to the two new W3C Semantic Web Working Groups (Best Practices and Deployment; Data Access).

1998 ACM Computing Classification System: H.5.1 H.3.3
Keywords and Phrases: video search query rules porting video, search, query, rules, ontology porting, Media Streams
Note: This work was carried out under project INS2 - NWO NASH

Video on the Semantic Web

Experiences with Media Streams

Joost Geurts[1], Jacco van Ossenbruggen[1], Lynda Hardman[1], Marc Davis[2]

[1] CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
[2] UC Berkeley SIMS, 314 South Hall, Berkeley, CA 94720, USA
Firstname.Lastname@cwi.nl, marc@sims.berkeley.edu

Abstract. In this paper, we report our experiences with the use of Semantic Web technology for annotating digital video material. Web technology is used to transform a large, existing video ontology embedded in an annotation tool into a commonly accessible format. The recombination of existing video material is then used as an example application, in which the video metadata enables the retrieval of video footage based on both content descriptions and cinematographic concepts, such as establishing and reaction shots.
The paper focuses on the practical issues of porting ontological information to the Semantic Web, the multimedia-specific issues of video annotation, and requirements for Semantic Web query and access patterns. It thereby explicitly aims at providing input to the two new W3C Semantic Web Working Groups (Best Practices and Deployment; Data Access).

1 Introduction

While digital video is gradually becoming mainstream, effective processing of video material is still problematic. Video retrieval is still widely recognized as an unsolved problem and for most practical retrieval applications, manual video annotation is a requirement. At first sight, current Semantic Web technology provides a readily applicable set of tools and languages for annotating video material. Our experiences with this task teach us, however, that still much work needs to be done on establishing best practices, tool support and the development of commonly available ontologies for video annotation.

Media Streams [3], a tool developed in the early nineties, provides a platform in which existing material can be annotated and retrieved. Part of our work includes a port of the Media Streams ontology — that was embedded in the tool — to RDF Schema, in order to exploit the Media Streams annotations in a Semantic Web setting.

This paper provides an experience report, discussing the use of current Semantic Web languages and the required tools in the context of video annotation. As an example application we use a prototype system that aims at retrieving video fragments to re-combine them in a new context, often for completely different reasons than the original context in which the material was shot. We use this particular application because it requires high quality markup, that describes both low and high-level features from different perspectives. It is also a very query-intensive application, and thus provides ample opportunity to experiment with Semantic Web queries.

This paper is structured as follows. After discussing related work we first describe an example scenario. We use this scenario to illustrate the discussion in the remainder of this paper. We give a brief overview of the key concepts underlying the Media Streams ontology, and discuss how we ported the ontology and annotations to the Semantic Web. We then describe the development of the prototype, focusing on a discussion of the pros and cons of the Semantic Web technologies used. Finally, we summarize our lessons learned and provide directions for future work.

2 Related Work

The main International Standard on annotation of audio visual material is MPEG-7 [8,9]. IBM's VideoAnnEx tool [7], for example, can be used for annotation video with MPEG-7 descriptions. MPEG-7 is a large and complex standard, that provides its own schema language (the MPEG-7 DDL or Data Description Language) that is based on XML Schema. Using the DDL, MPEG-7 defines many schemata to annotate audio visual material. Note that the choice for XML Schema not only impacts syntax level aspects. Since no formal semantics are provided, one needs to read the English prose in the standard to understand the meaning of the schemata. Also note that unlike W3C's Semantic Web Recommendations, the MPEG-7 standard and its schemata are not freely available. For a more detailed comparison of MPEG-7 and Semantic Web technology, see [16,17].

Previous work on combining MPEG-7 and Semantic Web technology includes a conversion of part of the MPEG-7 schemata into an ontology, as described by Hunter [6]. The conversion is part of a more encompassing architecture that proposes a parallel use of syntactic (XML) schemata and semantic (RDF) schemata. A similar architecture is described by Troncy [15]. It uses OWL for the semantic descriptions and MPEG-7 XML schemata for the structural descriptions.

Previous work by the authors [5,12] described an engine to automatically generate/deduce (narrative) structures out of meta data repositories, in order to represent the information in a meaningful way within multimedia presentations. This is similar to the video generation work here to the extent that it is a knowledge intensive process that requires the integration of multiple ontologies describing diverse topics such as domain, discourse and design knowledge. These are typically heterogeneous sources which need to be able to be easily reused, combined and exchanged with other applications: a weak point of many of the earlier knowledge intensive applications developed in the "pre Semantic Web" AI tradition.

Related work on video annotation for montage is discussed in Section 4 in the context of the Media Streams application.

3 Scenario: a Video Trip Report

Our prototype application explores to what extent it is possible to facilitate the editing process by formalizing the concepts of reaction and establishing shots. The approach is to query an annotated video repository and to find out how the queries need to be formulated to retrieve the desired sequences of the scenario described below. This approach

1. Campanile campus University of California, Berkeley.
2. Protagonist enjoying the sun, looking in the distance.
3. Clock tower showing the time.
4. Protagonist leaving in a hurry.
5. Protagonist entering a university building.
6. Protagonist entering lecture room.
7. Professor is not amused.
8. Protagonist feels ashamed.

Fig. 1. Representative frames of anecdote scenes.

not only provides insights in the required query mechanism, but also in the required granularity and different perspectives of the annotations and the required ontological reasoning. We base our example application around the following scenario.

Our protagonist, a 28 year old Dutch PhD student visits the University of California in Berkeley to learn about automating media production. When he returns to Amsterdam his colleagues are curious about his experiences abroad. Instead of writing a report he decides to make a more engaging video report using his newly acquired cinematographic knowledge and video material he shot during his stay. The report starts off with an anecdote how he nearly missed his first lecture because of the good weather.

A narrative film consists of *scenes* which are characterized by a consistent setting (note that the word "scene" originated from theater where at the end of a scene the curtains closed in order to change the decor and establish a new setting). A *scene* is comprised of one or more *sequences* which are comprised of one, or a series of *shots*. A shot is defined by a single (perceived) recording (turning the camera on and off). Within this paper we consider shots the atomic units of film. In film theory and practice there exists a variety of typologies of shots that help determine their possible syntax and semantics. In common narrative cinematic constructions, a scene often makes use of establishing shots and reaction shots [1].

The simple narrative of our scenario is composed of an establishing shot[3] depicting the student in Berkeley (shot 1 and 2 in figure 1). This is followed by a reaction shot

[3] Note that it is common usage to speak of "an establishing shot", even if it is in fact composed of multiple shots.

which communicates the student being late (shot 3,4). Then another establishing shot of the student entering the lecture room (shot 5,6). Finalized by a reaction shot of the student meeting his professor (shot 7,8).

Establishing shots establish the location of a scene in space and time. A common pattern used in establishing shots is to first show an exterior shot that then is followed by an interior shot. For example, shot 5 shows our protagonist entering a university building, and shot 6 depicts him entering in a lecture room. The viewer will conclude the exterior and interior belong to the same building. Note that in reality this is not necessarily the case [11].

In addition to establishing shots, the scenario uses reaction shots. A reaction shot often depicts an actor's facial expression or emotive gesture in apparent reaction to a preceding shot of an event or action. For example, Shot 7 is a reaction shot depicting the Professor's reaction to the student's entrance to class in Shot 6. Viewers are likely to interpret Shot 7 as causally connected to Shot 6, i.e., as the Professor disapprovingly reacting to the late entrance of the student. Again, this does not need to be the case in reality.

Combining shots into scenes typically requires a level of consistency between the shots. For example, two shots with similar backgrounds are perceived as being at the same location. The first establishing shot in the scenario above used this principle to suggest that our protagonist is sitting at a location near the Campanile Tower at UC Berkeley.

4 Video Annotation Using the Media Streams Ontology

Media Streams [4] was first developed at MIT Medialab to annotate video sequences in order to search and retrieve them for reuse purposes. A video in Media Streams is described by a *Timeline*, which represents multi-layered semantic information about a video arranged in temporal *Streams*, which define descriptive dimensions of a film, such as cinematography[4] and mise-en-scene[5]. Associated with every stream (category) is an ontology which defines the descriptive properties of the respective stream.

The terms in the ontology are graphically represented by icons, called "cascading icon dialog item" or *CIDI*. The individual ontologies of the streams are combined in the Media Streams ontology by a *CIDI* "dramaturgy" as root. Note that some of the categories have partly overlapping class-hierarchies. For example, "car" occurs in the categories "characters"[6], "objects" and "space".

The ontology defines close to 7000 *CIDI's*. The relationships between *CIDI's* are denoted by "subordinates", which is an abstract relation and can be further specified by: "subClassOf", "superClassOf", "partOf", "hasParts", "occursWith" and "looksLike"

An annotator creates a description of a feature in a film, such as a character description, by selecting appropriate icons from the respective stream hierarchy, which is

[4] "General term for all the manipulations of a filmstrip by the camera in the shooting phase and by the laboratory in the development phase." (([1] page 501.)

[5] "All of the elements placed in front of the camera to be photographed: the settings and props, lighting, costumes and make-up, figure behavior." ([1] page 504.)

[6] E.g. Herbie, Kit, Benny the Cab

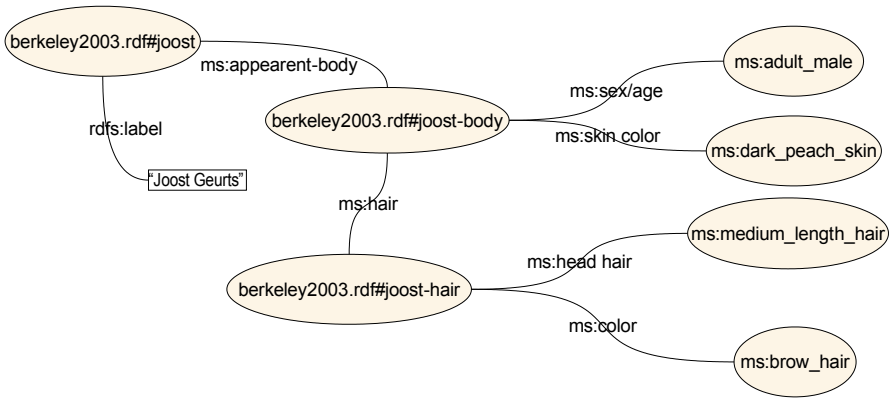

Fig. 2. Part of the annotation graph of the second shot (see Figure 1).

called a *Compound*. A compound contains *Slots* which have a name and a value field. A compound is then dragged upon the timeline to the appropriate start and end frame indicating the *Occurrence* of the compound in the video.

The Streams in Media Streams are:

characters includes terms to describe visual appearance of the characters in the shot (e.g. “adult male”, “brown hair”, “blue shirt”). The protagonist portrayed in the anecdote scene (figure 1) is annotated with the terms “adult male”, “dark peach skin” (figure 2). In addition to atomic terms, compound annotations can be nested. Within the compound which describes the character a nested compound describing the protagonist’s hair is embedded. In this case containing the terms “brown” and “medium length hair”, thus denoting “medium length brown hair”. The advantage of compounds is that they can be referenced. For example, similar occurrences of the character in the video can be annotated by providing a reference to the previously made description. For this reason, *compounds* have no start and end frame associated with them. Instead, they are stored in a Media Streams data structure which is called an *occurrence*.

objects includes terms to describe the visual appearance of objects in the shot. (e.g. “axe”, “knife”). The tower is annotated with the terms “clock tower”, “white”.

space includes terms to describe the visual appearance of the scene (e.g. “bathroom”, “America”) and the vocabulary to describe geographic locations. In general, annotators are advised to be conservative using it because of the possible repurposing of the shot. For example, a shot showing a busy street in Paris (without too obvious street-signs) can just as well serve as a busy street in another French city. A shot of the Eiffel tower, however, can legitimately be annotated with a geographic location “Paris” since it is a distinctive feature. For example, the anecdote scene shot 2 (figure 1) is annotated with the very generic term “outside”.

time includes terms to describe the era, the time of year and time of day (e.g. “eighties”, “summer”, “midnight”).

weather includes terms to describe visual weather conditions (e.g. “rainy”, “sunny”). For example, the weather condition of the first shot displayed in figure 1 is annotated as “clear”, the type of the annotation is “moisture-related weather” (“moisture-related weather” subordinates “clear”).

character actions includes terms to describe an action performed by a character (e.g. “holds”, “walks”). Some streams including character actions, object actions and cinematography can combine annotations from different streams. This is realized by means of a generic template defining slots for “Subject”, “Subject-Part”, “Instrument”, “Action”, “Start-Point”, “End-Point”, “Object” and “Object-Part”. A character gazing into the distance, as shown in shot 2 of Figure 1, is described by a reference to the character for the “Subject” slot and the term “left hand covers forehead”.

objects actions includes terms to describe an action performed by an object (e.g. “contains”, “moves”).

relative position includes terms to denote relative positions (e.g. “above”, “below”).

screen position includes terms to denote screen positions (e.g. “left side of screen”, “upper left corner of screen”).

cinematography includes terms to describe the cinematographic properties of the video (e.g. “camera angle”, “camera distance”, “camera movement”, “framing”).

recording media includes terms to denote the type of material (e.g. “color film”, “70mm”).

transitions include terms to denote the type of transition between two shots (e.g. “cut”, “wipe”, “fade”).

Annotations in Media Streams are relatively complex, consisting of nested structures and cross-references between them. The need for such a complex annotation scheme is due to the nature of video, which in addition to both a spatial and a temporal dimension also has multiple descriptive dimensions (Streams in Media Streams), such as cinematographic terms and content descriptive terms. From a re-use perspective, these dimensions need to be described separately, preferably without high level interpretations since this would constrain possible reuse. The Media Streams ontology thus consists of concrete concepts, which in the original application were represented by graphical icons in an attempt to limit varying interpretations as much as possible.

5 Automatic Video Montage Demands on the Semantic Web

In order to use the annotations and ontologies embedded within Media Streams in our automatic montage environment we first had to extract these from the system and transform them to Semantic Web representations. After our description and analysis of this semantic extraction process we discuss issues surrounding our example application: automatic video montage. In particular, we discuss particular characteristics of the video annotations required when querying for shots and the demands put on rules for assembling a series of shots into a plausible sequence.

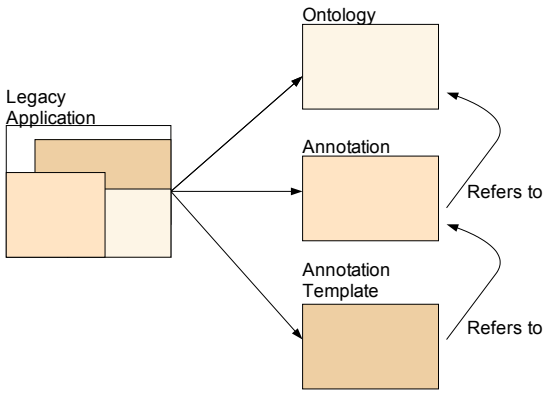

Fig. 3. Porting application knowledge

5.1 Transformation of Media Streams ontology and annotations to RDF(S)

The annotations and the ontology existing within Media Streams were transformed to RDF Schema as illustrated in figure 3. Note that the annotations are instances of an annotation template which describes the structure of an annotation (*Timeline*, *Occurrence*, *Compound*, *Slot* see section 4). Within an annotation instance slot-names and (atomic[7]) slot-values refer to *CIDI's* defined in the ontology. The Media Streams annotation tool is a graphical, icon-based tool, implemented at MIT in Macintosh Common Lisp [13], using the Common Lisp Object System (CLOS) in the Macintosh ToolBox GUI environment. The original implementation dates from the early nineties and relies on a number of machine-dependent binary media libraries. This means that the system runs only on a few Macintosh OS 9 machines. To make things worse, certain parts of the original Lisp source code were not readily available, and are only available in compiled form.

Syntactical dump to XML The underlying ontology used for the annotations, one of Media Streams's key assets, is represented graphically in the annotation tool by a hierarchy of *CIDI's*. The only machine-readable version of the ontology is thus embedded in the code of the annotation tool. We developed a small Lisp script that uses the Media Streams Lisp API to extract as much ontological knowledge as possible from the annotation tool and convert it to a more accessible and platform independent format. Because this extraction script works only on an Macintosh with Media Streams installed, we chose to "dump" the ontological information into a platform-independent XML structure. This mirrored the original structure as faithfully as possible and deliberately postponed the design decisions about how to map this structure to RDF Schema or OWL.

[7] Can be a reference to a *Compound*.

A similar extraction script was developed that could read the (binary) annotation files in the Media Streams repository. This script provided an XML dump for each annotated video fragment. Again, we aimed solely at providing a faithful reproduction of the source and postponed RDF modeling decisions to a later phase.

Semantic transformation to RDF Schema XSLT sheets were used to transform both ontologies into RDF. They generate an RDFS class for each *CIDI*, mapping the Media Streams subClassOf relation to the RDFS equivalent. The other Media Streams subordinate relations, such as partOf, were mapped onto RDF properties with the same name — thereby losing their semantics for a generic RDFS application. Although the Media Streams ontology is technically a single hierarchy, the GUI suggests modularity of streams. The XSLT sheets acknowledge this and transform the ontology for each individual stream into an RDF Schema[8]. The current result is a first step toward a more sophisticated transformation of Media Streams and we are also investigating an OWL version. Wielinga et al. [19] discuss the modeling decisions that need to be made during such transformations in more detail.

We found that a straight port of the Media Streams ontology and annotations was, in most cases, insufficient and we needed to fall back onto our (implicit) working knowledge of the application to interpret the ontology and annotations.

While the added value of Semantic Web technology is that ontologies can be interrelated, pre-Semantic Web applications, such as Media Streams, were often created with a particular use in mind. While originally designed to be extensible, the current implementation of the Media Streams contains a single, closed and not easily extensible ontology. However, for other video annotation tasks such as analysis, more domain specific ontologies might be better suited than the rather generic Media Streams ontology. In the context of a Semantic Web setting linking different ontologies is certainly something to consider.

Once both the ontology and the annotations have been transformed to RDF(S), off-the-shelf tools can be used to query them. To check the correctness of the repository and the retrieved results, Sesame [2] was used to browse the ontology and annotations during the transformation from XML to RDF Schema.

5.2 Querying for shots

The annotations within Media Streams are relatively complex due to the multiple descriptive dimensions and the fact that they can be nested and can contain references. In addition, the annotations need to be quite detailed because this is often required to maintain continuity. There are, however, also cases in which details are less important, for example in a fast action scene. The query engine thus needs to deal with situations in which a query is formulated precisely (e.g. “head shot of an adult male, brown hair, blue eyes”) and where it is more general (e.g. “adult male”). In this section we discuss the requirements surrounding the need for controlling the generalization of the query and

[8] The modularity is non-destructive, since the individual ontologies together define the original Media Streams ontology.

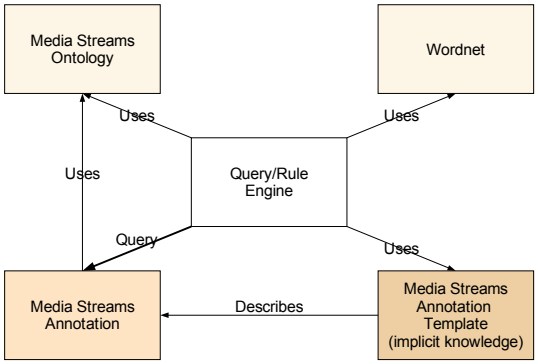

Fig. 4. Query dependencies

how this can be controlled to some extent through the use of keywords in conjunction with extra ontological information.

During the development phase of the automatic montage prototype we used the `semanticweb` package of SWI Prolog [18] for its ability to backtrack. This proved useful within a context where controlled trade-offs need to be made during querying.

The need for controlled generalization A more general query will typically give more results, and therefore the chances of finding a shot are higher. For precisely defined queries, the chances of finding the shot are lower. Over-constrained queries are likely to cause the complete automatic montage process to fail. To prevent this, the engine also needs to be able to relax a precise query when it yields no results. We call this mechanism "controlled generalization". It allows us to use subsumption reasoning when the precise query gives no, or too few results. When the subsumption query produces insufficient results as well the query is allowed to generalize under certain conditions (a parameter sets the level of generalization, we specified only direct superclasses in our application). For example, a query for "pair of blue jeans" is allowed to match with "pants", which is a direct superclass of "blue jeans". In contrast, "piece of clothing", which is not a direct superclass, is not allowed. In addition, generalization is only permitted to take place below a minimum level in the subsumption hierarchy to prevent over-generalization.

A query or rule (see Figure 4) may use different knowledge repositories, such as ontological knowledge about the terms (subClass, looksLike), the annotation template which allows different ways of specifying similar features, and a mapping to WordNet (discussed below) which deals with synonyms in order to increase the number of matched results. All these methods need to be controlled so we can use them in a conditional way.

Label Based Search On a more practical level, every *CIDI* (see 5.1) represents a concept and is associated with a unique keyword. Searching for a synonym, however, will return no results. For example, searching for "spectacles" will give an empty result set since "spectacles" is not defined as a keyword even though *CIDI* "eyeglasses" exists.

To increase the chance of successfully retrieving a relevant shot we mapped[9] *CIDI* labels to WordNet [14] WordForms[10] of which synonym WordForms can be derived (see Figure 4).

Similar WordForms with different meanings such as "spring" which occurs as the verb (to jump) and as a noun (season, spiral elastic device) can result in erroneous retrieval. Some of these errors can be prevented by requiring that the WordNet type (e.g. verb, noun) of a word matches descendants of a particular *CIDI*. For example, WordNet verbs only match with *CIDI* descendant of "character-actions" or "object-actions" and WordNet nouns match with *CIDI* descendants "object" and "locations".

In addition, "spring" also occurs in WordNet as WordForms for nouns with different meanings (water source, season, object). The system is not able to disambiguate these without extra information and, therefore, erroneous mappings can occur. In practice, however, it rarely happens that a shot is retrieved based on a single term. In most cases the query contains a more elaborate description of a relevant shot. When searching for a shot where the weather condition is specified to be "spring", a query on an incorrect synonym for spring, such as water source, will be harmless since nothing will be annotated with "weather: water source".

The WordNet mapping is a hack to make things work. Although it works reasonably well, errors occur which are not easy to prevent, in addition the mapping is not very efficient since it is done at run time. There are a few optimizations, however, which match Media Streams categories to WordNet WordTypes. For example, a *CIDI* "walk" in the category character action is only matched to WordNet words which are of the WordNet type verb, so that unwanted matches, such as the WordNet type noun "walk", are ignored.

5.3 Formalizing Establishing Shot: the need for Semantic Web rules

The automatic montage prototype facilitates the editing of video sequences by formalizing rules of continuity editing as used by expert montage editors. The rules used to generate establishing shots are based on cinematographic principles to maintain continuity, based on content descriptions, such as film material, camera movements and lens settings. Figure 5 gives an example rule that retrieves two shots, which combined, form an establishing shot for a character at a certain building.

The example rule defines an establishing shot as a combination of two sequences of which both shots are in color, the focus of the first has a wider angle then the next shot, and the weather conditions are similar. Note that the rule can also be described as a query that returns all shot combinations that match a certain description: we found that

[9] Using a naive literal string match.

[10] A WordForm in WordNet is a textual representation of a Word. For example "car" and "automobile" are both WordForms(representations) of a Word(Concept).

```
establishing(ObjectDescription,CharacterDescription,ObjectShot,CharacterShot) :-

  % Object is the slotvalue fitting ObjectDescription
   (1)  compound(_Name,ObjectDescription,_Type,Object),

  % ObjectFraming frames object description
   (2)  framing(ObjectParent,ObjectFraming,Object),
   (3)  isa(GeneralObjectFraming,'framing of an object'),

  % Character is the slotvalue fitting CharacterDescription
   (4)  compound(_Name,CharacterDescription,_Type,Character),

  % CharacterFraming frames character description
   (5)  framing(CharacterParent,CharacterFraming,Character),
   (6)  isa(PersonFraming,'person framing'),
   (7)  find_literal_in_compound('cinematography',PersonFraming,_Type,CharacterFraming),

  % Map to a shot (which has url, and frameno)
   (8)  occurrence_compound(ObjectShot,ObjectParent),
   (9)  occurrence_compound(CharacterShot,CharacterParent).

  % not shown: weather, in/out doors, focus
```

Fig. 5. Prolog rule for establishing shot.

in this (and probably many more) application the difference between a rule and query is rather artificial.

The rule takes two input parameters, a description of an architectural object and a description of a character. Note that the author of the establishing shot query is responsible for giving descriptions which fit the location to be established. The shots used in the resulting sequence might not actually be at that location (which is the point of an establishing shot). Line (1) in Figure 5 queries for a compound which fits the description of the building given as an input parameter. The description consist of a list with slot-name, slot-value pairs. An example of a valid description is: `[tower,color:white]`. Line (1) matches a compound which fits this description. Line (2) then queries for the framing of this object. Note that if the matched object in line (1) does not have a specified framing, the system backtracks over line (1) until it both satisfies the description and the framing specified, this backtracking holds for all lines when it cannot satisfy the current line. Line (3) states that the shot's main focus should be that of an object. Line (4-7) query for a character analogue to the query for an object. Finally in line (8 and 9) the compound is mapped to an Occurrence which holds the start frame and end frame and the URL to the clip[11].

Note that this kind of (partial) queries that depend on backtracking are more conveniently specified in Prolog than in higher level languages such as RQL [10] and SeRQL [2]. These language require a complete definition of the query in advance, or a manual split-up in many sub-queries and an external programming environment that mimics the backtrack behavior of Prolog. The advantage of these high level languages is, however, that the syntax is more intuitive. In addition, query optimizations, which are absent in Prolog, make them perform better for complex queries for which backtracking over query fragments is not an necessary.

[11] This is only one example rule for an establishing shot. There are many possible alternatives, including a text overlay denoting the location. Which one to choose is dependent on the available material, their annotations and the context in which it is used.

6 Lessons Learned

In this section, we summarize the lessons learned during the transformation from the original ontology and annotations to RDF Schema and during the development of the automatic montage prototype.

6.1 Lessons learned during transformation

Annotations and ontologies are always developed for a particular goal and in a particular context that needs to be understood. For example, while most of the Media Streams ontology was encoded explicitly in Lisp, parts of it could be better understood after analyzing the hierarchical structure of the corresponding GUI. So the first lesson learned is that ideally, one should not start porting knowledge to the Semantic Web without having a thorough understanding of not only the underlying domain but also the application context for which that knowledge was originally intended.

Having said that, in many practical cases, there is a strong need to convert as quickly as possible. This could be caused by a requirement to show results or benefits early in a project, or, as in our case, the need to be able to develop on platform different from that of the original application. We followed a hybrid approach by doing a quick and dirty, but complete, conversion to an initial XML dump, and used that as a basis for incrementally generating more refined RDF Schema versions as we gained more insights in the domain and the Media Streams application. See Wielinga et al. [19] for a good overview of the types of practical decisions that need to be taken when converting legacy resources to the Semantic Web.

When porting existing ontologies to the Web, the obvious question is how to combine them with other Semantic Web sources. For example, part of the Media Streams ontology contains terminology that is already described by other, more common vocabularies (c.f. Dublin Core, VRA). Second, the Media Streams world model turns out to be incomplete in practice. This requires a facility that allows vocabulary extensions to be defined and plugged-in. Such a modular and open approach requires, however, a more thorough redesign of the annotations, ontology and processing applications than described in this paper. In general, one should be aware that a straightforward port to a Semantic Web language does not necessarily make an ontology open in the Semantic Web "spirit".

Media Streams uses English language labels to identify its concepts. Every concept is associated with exactly one label, although the labels are enriched by explicitly defining that they are in English during the transformation. Plain text labels provide a possible source of error because of language ambiguity and absence of synonym terms. Within the prototype we created a dynamic mapping to WordNet. This mapping has the advantage that it is relatively simple to implement and it scales with improved versions of WordNet. On the downside, the mappings are not necessarily correct. In the prototype example the context of the application provides enough information to filter out most errors. Alternatively a human annotator needs to interpret the Media Streams term and provide the correct mapping.

6.2 Lessons learned during prototyping

Automatic video generation is dependent on the quality of annotation, since subtle differences can break continuity. For many cases, especially those involving recognizable actors, the assumption that the required annotation quality can be practically achieved is unrealistic. In these cases, the assumption that the database will contain material that is shot for other purposes, but fulfilling all requirements that make it reusable in a new context may also be wishful thinking. There are, however, common cinematographic constructs that are less demanding on the quality of the annotations or the material. We used one of these constructs, the establishing shot, to experiment with. While less demanding, it still requires the multifaceted annotations and advanced querying that make it a challenging use case.

Media Streams allows for elaborate descriptions of shots, which make the definition of the inner structure of the annotations a non-trivial task. We want to stress that in this, as in many other applications, the annotations are *not* simply instances of the ontology being used. Instead, the annotations assign properties to the media content, where the property names and values refer to terms in the ontology. The structure of these annotations is, however, independent from the ontology being used.

In Media Streams, this structure was not explicitly defined and coded implicitly in the annotation tool. However, if two applications want to share annotations, they not only need to agree on the ontologies being used, but also on the structure of the annotations. For this purpose, we defined the annotation schema explicitly, currently in RDF Schema[12]. Note that having this schema does not prevent annotators from annotating the same fragment in different ways: a schema that could enforce this would be too rigid to be usable in practice.

Applications such as Media Streams require complex annotations that provide a precise description of the content. Such applications require a query engine that is sufficiently expressive to deal with the associated detailed queries. We also learned, however, that these detailed queries are sometimes over-constrained and need to be relaxed. Strategies for generalizing queries require ontological knowledge but will remain dependent on the domain and application. We found high level query languages such as SeRQL particular suitable for precisely defined queries. Fuzzy queries, which are characterized by alternative descriptions, are better formulated in a lower level query language such as Prolog, where backtracking naturally invokes alternative rules and queries to find a successful match.

7 Conclusions and Future Work

In this paper, we discussed our experiences with the use of Semantic Web technology for video annotation. We used a prototype application that depends on high quality annotations, based on the Media Streams video ontology. The prototype uses common Semantic Web technology as much as possible.

[12] Note that since this schema adds little semantics and is mainly structural, it could as well be defined using XML Schema.

We converted the Media Streams ontology to a Semantic Web format. We used a two step approach that allowed us to incrementally refine the modeling decisions during the conversion of instances and the development of the prototype. We found that taking the right modeling decisions requires detailed knowledge of the application domain. We also found that a straightforward conversion to RDF Schema or OWL might make an ontology syntactically compatible with the Semantic Web, but it does not necessarily make it an "open" ontology than can be easily combined with other Web sources, shared with other applications or used in an international, multilingual setting.

Assuming that no video ontology will be "complete" and well suited for every application, the ability to use multiple ontologies in a single annotation and to able to reuse existing annotations is of key importance. In that context, the ability to use and reuse MPEG-7 annotations will be an important requirement for industrial scale use of video annotations on the Semantic Web.

In the context of query functionality, we found that in addition to high-level, declarative query languages such as RQL and SeRQL, a query mechanism that can make use of subsumption reasoning in a controlled way is also very important. Secondly, we found that in our application, the distinction between queries and rules is quite small, and we recommend a highly integrated approach when developing standardized query and rule languages (c.f. XPath selectors and XSLT rules).

Our research prototype proved to be well suited to experiment with annotations, ontology-based search and rule-based video generation. Practical application of these technologies, however, requires user friendly, Semantic Web enabled video annotation and video editing tools.

The vision is that Media Streams — just as other knowledge intensive, but closed AI applications from the eighties and nineties — will be given a second life on the Semantic Web. Porting such applications, however, requires more than a straightforward conversion. To overcome the limitations of the original systems we require commonly available, open, reusable, interoperable and internationalized ontologies and annotation schemes, along with practical user guidelines.

Acknowledgments

Part of the research described here was carried out during a visit of the first author to the Garage Cinema Research Group in Berkeley, California. It was financed by the Dutch NWO NASH project.

The authors wish to thank their colleagues at CWI for their input to the work, in particular Frank Nack.

References

1. D. Bordwell and K. Thompson. *Film Art: An Introduction.* McGraw-Hill, 7 edition, 2003.
2. J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In I. Horrocks and J. Hendler, editors, *The Semantic Web - ISWC 2002*, number 2342 in Lecture Notes in Computer Science, pages 54–68, Berlin Heidelberg, 2002. Springer.

3. M. Davis. *Media Streams: Representing Video for Retrieval and Repurposing*. PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995.
4. M. Davis. *Readings in Human-Computer Interaction: Toward the Year 2000*, chapter Media Streams: An Iconic Visual Language for Video Representation., pages 854–866. Morgan Kaufmann Publishers, Inc., 1995.
5. J. Geurts, S. Bocconi, J. van Ossenbruggen, and L. Hardman. Towards Ontology-driven Discourse: From Semantic Graphs to Multimedia Presentations. In *Second International Semantic Web Conference (ISWC2003)*, pages 597–612, Sanibel Island, Florida, USA, October 20-23, 2003.
6. J. Hunter. Adding Multimedia to the Semantic Web — Building an MPEG-7 Ontology. In *International Semantic Web Working Symposium (SWWS)*, Stanford University, California, USA, July 30 - August 1, 2001.
7. IBM research. VideoAnnEx - IBM MPEG-7 Annotation Tool, 2002.
8. ISO/IEC. Text of ISO/IEC 15938-5/FDIS Information Technology - Multimedia Content Description Interface - Part 5: Multimedia Description Schemes. ISO/IEC JTC 1/SC 29/WG 11/N4242, Singapore, September 2001.
9. ISO/IEC. Overview of the MPEG-7 Standard (version 8). ISO/IEC JTC1/SC29/WG11/N4980, Klagenfurt, July 2002.
10. G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl. RQL: A Declarative Query Language for RDF. In *Proceedings of the Eleventh International World Wide Web Conference (WWW2002)*, pages 592–603, Honolulu, Hawaii, USA, May 2002. ACM Press.
11. L. Kuleshov. *Kuleshov on film: Writings by Lev Kuleshov*. Univercity of California Press, 1974.
12. S. Little, J. Geurts, and J. Hunter. Dynamic Generation of Intelligent Multimedia Presentations through Semantic Inferencing. In *6th European Conference on Research and Advanced Technology for Digital Libraries*, pages 158–189. Springer, September 2002.
13. Macintosh Common Lisp. http://www.digitool.com/home.html.
14. S. Melnik and S. Decker. Wordnet RDF Representation. http://www.semanticweb.org/library/, 2001.
15. R. Troncy. Integrating Structure and Semantics into Audio-visual Documents. In *Second International Semantic Web Conference (ISWC2003)*, pages 566 – 581, Sanibel Island, Florida, USA, October 20-23, 2003. Springer-Verlag Heidelberg.
16. J. van Ossenbruggen, F. Nack, and L. Hardman. That Obscure Object of Desire: Multimedia Metadata on the Web (Part I). *IEEE Multimedia*, to be published in 2004, based on http://ftp.cwi.nl/CWIreports/INS//INS-E0308.pdf.
17. J. van Ossenbruggen, F. Nack, and L. Hardman. That Obscure Object of Desire: Multimedia Metadata on the Web (Part II). *IEEE Multimedia*, to be published in 2004, based on http://ftp.cwi.nl/CWIreports/INS//INS-E0309.pdf.
18. J. Wielemaker, G. Schreiber, and B. Wielinga. Prolog-Based Infrastructure for RDF: Scalability and Performance. In *The SemanticWeb - ISWC 2003*, pages 644 – 658, Sanibel Island, Florida, USA, October 20-23, 2003. Springer-Verlag Heidelberg.
19. B. Wielinga, J. Wielemaker, G. Schreiber, and M. van Assem. Methods for Porting Resources to the Semantic Web. In *1st European Semantic Web Symposium (ESWS 2004)*, Heraklion, Greece, May 10-12, 2004 (to be published).