Ontology-based Semantic Indexing for MPEG-7 and TV-Anytime Audiovisual Content

Chrisa Tsinaraki[*] Panagiotis Polydoros[*] Fotis Kazasis[*] Stavros Christodoulakis[*]

chrisa@ced.tuc.gr panpolyd@ced.tuc.gr fotis@ced.tuc.gr stavros@ced.tuc.gr

Abstract

In this paper, we describe a framework that we have developed for the support of ontology-based semantic indexing and retrieval of audiovisual content following the MPEG-7 and TV-Anytime standard specifications for metadata descriptions. Our work aims to provide a methodology to enhance the retrieval effectiveness of audiovisual content, while maintaining compatibility with the international multimedia standards.

In our framework, domain-specific ontologies guide the definition of both the application-specific metadata and the instance-description metadata that describe the contents of audiovisual programs and/or their segments. The ontologies are also used in order to provide compatible descriptions in both audiovisual content standards (MPEG-7 and TV-Anytime) for the same content. This approach allows indexing compatibility and interoperability of TV-Anytime and digital library applications.

We describe the design and implementation of a system supporting this framework. The components of the system include a segmentation tool for segmenting audiovisual information, which also provides ontology-based semantic indexing capabilities, and an appropriate API for semantic query support. An application testbed for the domain of soccer games has been developed on top of this system. An ontology for soccer games has been defined and used for indexing and retrieval of soccer games that have been stored in the system database.

The methodology we developed opens up a wide opportunity for the creation of MPEG-7 and TV-Anytime services offering structured domain-specific ontologies that can be integrated to these standards for enhancing audiovisual content retrieval performance.[†]

Keywords: Semantic Indexing, MPEG-7, TV-Anytime, Ontologies, Segmentation

1 Introduction

The increasing cooperation, and sometimes merging, of computer, home electronics, software, broadcasting and audiovisual information provision industries that take place in the last few years, result in offering advanced audiovisual services to users. These services include audiovisual digital libraries [9], [13], [18], video on demand services [3] [20] [22] [25] and recently TV-Anytime services [18]. *MPEG-7* [23] and *TV-Anytime* [32] are well-accepted standards for the description of audiovisual content. They both provide capabilities for specifying structured metadata descriptions of both the audiovisual content and the user preferences regarding audiovisual content.

The MPEG-7 descriptions can be used in any audiovisual digital library environment or any other audiovisual web application environment to describe audiovisual content and user preferences. They can be used for example by search engines or information dissemination systems to deliver to the users multimedia content matching their interests. The TV-Anytime specifications on the other hand are promoted by the TV-Anytime Forum, and are targeting broadcasting environments and interactive TV services. The simplest TV-Anytime scenario foresees TVs equipped with a processor and large, inexpensive disk storage devices, which will select, store and replay broadcasted audiovisual content that matches the user interests. The TV-Anytime metadata describe both the broadcasted audiovisual content as well as the user profiles. More advanced TV-Anytime scenarios foresee TV-Anytime servers that connect to the user's client with last mile connections to provide larger selection and parallel capturing of programs broadcasted from thousands of digital TV channels. It is clear that the TV-Anytime servers can provide value added services including the provision of additional audiovisual content, which is of interest to the user.

---

[*] Lab. of Distributed Multimedia Information Systems and Applications (MUSIC/TUC), Technical University of Crete Campus, 73100 Kounoupidiana, Chania, Greece, http://www.music.tuc.gr/

[†] The work presented in this paper was partially funded in the scope of the EU-funded project (IST-1999-20751) UP-TV: "Ubiquitous personalized interactive Multimedia TV Systems and Services" [33] and the DELOS II Network of Excellence in Digital Libraries [5]

It has been discussed in many scientific forums [24][5] that the use of domain-specific ontologies improves information retrieval performance (in terms of precision/recall). It also facilitates the work of information indexers and the correctness of indexing in manual, semi-automatic or automatic indexing environments. In addition, it facilitates the users to pose queries or profiles, and the system to build correct user profiles. The above are also true for multimedia information. Domain-specific ontologies for indexing multimedia content may be highly structured. It is important to understand that these structured ontologies have to be smoothly integrated with the multimedia content description standards (MPEG-7 and TV-Anytime) using the structuring mechanisms provided by the standards, so that the integration is transparent to the applications and tools using these standards. This would allow those applications and tools to work without any knowledge of the domain-specific ontologies. If this is not the case, the tools and applications should be aware of the ontologies used beforehand and the ontologies should be components of the tools and applications, thus reducing software interoperability. Our aim in this paper is to provide a methodology for transparent integration of structured ontologies with the multimedia content description standards.

In more advanced application environments it is also important to be able to use the same ontologies across different multimedia standards. For example, a sports reporter might search among soccer game videos in a digital video library in order to retrieve videos or segments of videos of soccer games where a certain player, e.g. Ronaldo, scores. On the other hand, an end-user might denote in his preference profile in a personalized TV environment that he would like all the goals of a soccer game to be recorded for him. While viewing them he may request to see some extracts of the goals of Ronaldo against the same team two years ago from the digital library of the server, as well as articles, comments, critics from reporters, pictures or other audiovisual material about him.

A system that satisfies the requirements of the above use cases must store semantic descriptions – essentially semantic metadata – for the audiovisual content and possibly the user profiles, and provide semantic-based search capabilities on the content. The descriptors may refer to the whole multimedia content (programs, videos etc.) or to parts of the content (segments). To guarantee interoperability in involved application environments as the aforementioned ones, it is of outmost importance to adhere to international standards for describing audiovisual content, such as MPEG-7 and TV-Anytime. In addition, the more advanced TV-Anytime scenarios impose requirements that can be satisfied only with the MPEG-7 semantic description capabilities. Thus, a framework that permits the integration and interoperability of standards is needed for successful service provision.

Our research focus in this paper includes two aspects: First to provide enhanced content-based retrieval and dissemination capabilities for audiovisual content through the use of domain-specific ontologies for the knowledge domain of specific audiovisual content, and second to provide this extended functionality in a way that is compatible with the existing audiovisual content standards (MPEG-7 and TV-Anytime), thus also facilitating the interoperability across tools and applications that use these standards.

As far as we know, there are no specific proposals for the integration of domain-specific ontologies to the MPEG-7 and TV-Anytime standards also supporting interoperability of those standards. The domain-specific ontologies permit better support for specific application domains, without ad-hoc extensions of the standards.

Research on the semantic description of audiovisual data has been carried out, before the establishment of standards, using different models [1] [2] [10] [17] [18] [3], some of which have been implemented in either prototype or real-world systems. While the standards were under review, some research groups used the draft standards in order to investigate their applicability in system development [31] [30] [14]. Other systems have been based on specific audiovisual standards, after their finalization, but there exists no interoperability support [14] [18].

In this paper we describe the framework and the software infrastructure that we have developed for the support of ontology-based semantic indexing and retrieval of audiovisual content following the MPEG-7 standard specifications for metadata descriptions. The framework is also applicable for content metadata which have been structured according to the TV-Anytime standard and the same ontologies can be used to guide the TV-Anytime content and user profile metadata derivations and retrieval processes. In our framework, domain-specific ontologies – that is, sets of entities and relationships among them that capture the knowledge on specific application domains [11] – guide the definition of both the application-specific metadata and the instance-description metadata that describe the contents of audiovisual programs and/or segments. The ontologies are also used in order to provide compatible descriptions in both standards (MPEG-7 and TV-Anytime) for the same content.

We describe the design and implementation of a system supporting our framework. The components of the system include a segmentation tool for segmenting audiovisual information, which also provides ontology-based semantic indexing capabilities, and an appropriate Application Programming Interface (API) for semantic query support. We demonstrate the use of this framework through the development of a complete ontology for soccer games, its capturing in the segmentation tool, and its use for segmentation, indexing and retrieval of soccer games using the MPEG-7 and the TV-Anytime standards.

The rest of this paper is organized as follows: In section 2 we provide an overview of the system architecture we have developed, while in section 3 we discuss our methodology for the integration of domain-specific ontologies with MPEG-7 and TV-Anytime. The semantic base, where semantic metadata are stored, is described in section 4, while in section 5 we describe our segmentation and semantic indexing tool. The support provided for semantic queries and the application of our framework ideas and implementation in the soccer domain are discussed in sections 6 and 7 respectively. In section 8 we present the conclusions of our work and a brief description of our future research in this area.

2 Architectural Overview

In this section we provide an overview of the system architecture that supports ontology-based semantic indexing for audiovisual content. An outline of the system architecture is shown in Figure 1. As shown in Figure 1, our approach provides end-user search interfaces, which are based on both MPEG-7 and TV-Anytime standard specifications. The major issue here is that any domain-specific ontology can be imported in the system in a systematic way, and that search and profiling software that has been developed by a third-party to be compatible to MPEG-7 and TV-Anytime can still be used to access the audiovisual content. The use of the ontologies enhances the retrieval effectiveness but does not affect the software that uses MPEG-7 and TV-Anytime constructs only. This is achieved through the provision of appropriate search APIs and compatible semantic indexing for both standards. The major architectural components are:

- The *Segmentation & Semantic Indexing Tool*, which is used during the audiovisual content segmentation. The segmentation process includes semantic indexing, which is carried out using the *Semantic Indexing Component* of the tool. The Semantic Indexing Component is responsible for the import of *domain-specific ontologies* and *application-specific metadata* (in both cases whenever needed), the definition of both application specific metadata and *instance description metadata*, and their storage in the system database. For example, in a soccer tournament application the application-specific metadata may be instances of the players of the teams that participate in the tournament, instances of the referees, coaches etc. These instances are reusable both in one game (e.g. they are related with several events in the video program of the specific game), as well as across several games in the tournament. The part of the application specific metadata appearing in the specific audiovisual program are the *instance specific metadata*. For example, the players that participate in a soccer game are parts of the instance specific metadata for this particular game. These players are also a subset of all the players that participate in the tournament (application specific metadata).The *instance description metadata* describe in general which events of the real world take place in which parts of the video. The event types, the participants of the events, the time and the place that the events take place are described by the ontologies and the application-specific metadata. During the segmentation, MPEG-7 compliant semantic metadata as well as TV-Anytime metadata (keywords), which are validated against the ontologies and the application-specific metadata are produced and used for the indexing of programs and segments. During querying or user profile formation, the specific queries or the user profile are validated against the ontologies and the application-specific metadata. The system currently supports ontologies in MPEG-7 compliant XML format, but the support of OWL [12] ontologies is under way.
- A relational database, where audiovisual segments and the metadata that describe them are stored. The relational database contains an *MPEG-7 compliant Semantic Base*, where the semantic metadata are stored and a *TV-Anytime compliant Database* where TV-Anytime metadata for audiovisual programs and segments are stored. References from the Semantic Base and the TV-Anytime database to the corresponding programs and/or segments are maintained.
- The appropriate interfaces, which permit the end users to pose both simple, to a large extent keyword-based queries (according to TV-Anytime) and more powerful, semantic queries (supported by MPEG-7). These interfaces are based on the existence of appropriate *query APIs (Application Programming Interfaces)*. Thus, an API for *simple keyword queries* along with other TV-Anytime metadata is used for TV-Anytime queries. An API that permits *semantic-based queries* has also been implemented for retrieval from MPEG-7 content descriptions. The later makes use of the semantic metadata stored in the semantic base, resulting in more accurate queries. Both interfaces validate

their queries against the domain-specific ontologies and the application specific metadata before submitting the query.

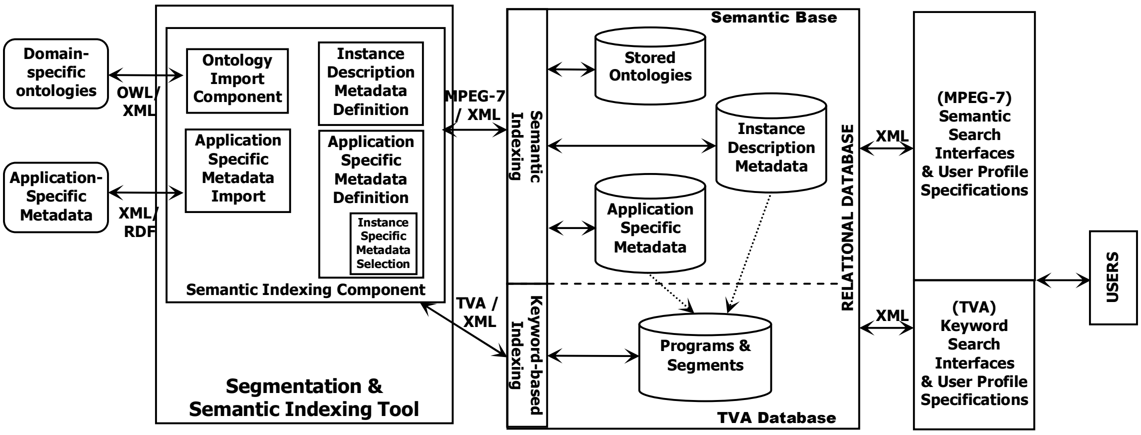

Figure 1: Architectural Overview. The segmentation and semantic indexing tool is also responsible for importing domain-specific ontologies and application-specific metadata, which are stored in the database as is the case with the metadata instances describing programs and segments. The search tools validate query instances against the ontologies and the application-specific metadata and proceed to matching queries against metadata descriptions.

The above-described framework has been implemented in the Linux OS environment. MySQL was the platform selected for the relational database (that contains both the Semantic Base and the TV-Anytime compliant database). XML data management in the relational database is carried out by a binder capable of fully representing XML documents as objects or objects as XML documents. The Segmentation & Semantic Indexing Tool has been implemented in java.swing and the query API functions were implemented using JDBC on top of the relational database.

3 Integrating Ontologies with MPEG-7 and TV-Anytime

In this section, we discuss the integration of domain-specific ontologies with the standards for metadata that describe multimedia content. The integration is needed, as our approach for the structure of content-description metadata is that it must be compliant with a two-layered model: A set of core entities, capable of describing any content, based on the well-accepted standards exists, while any domain-specific extension should be done using appropriate ontologies [30] [31]. This ontology-based approach allows compatible extension of the standards based on the same ontologies and allows for standard interoperability.

Before proceeding to the details of ontology integration with the standards (discussed in subsection 3.3), we provide an overview of the relevant constructs of MPEG-7 and TV-Anytime, on which we have based our work, for audiovisual content description (discussed in subsections 3.1 and 3.2 respectively).

3.1 MPEG-7 based Audiovisual Content Descriptions

We will refer here to the MPEG-7 constructs for the representation of metadata used for audiovisual content description and how ontology support is provided in MPEG-7. The complex data types needed for the semantic description of audiovisual content are defined in the *Semantic* part of the MPEG-7 MDS [15] specification, in a set of *Description Schemes (DSs)* rooted in the SemanticBase DS. The more important among the description schemes are:

- *SemanticBase DS:* The abstract type *SemanticBaseType* is defined here. SemanticBaseType is the base type extended by other description schemes according to the needs for the description of semantic entities of specific types in a narrative world.
- *SemanticBag DS* and *Semantic DS:* Description schemes used for the description of collections of semantic entities. *SemanticBagType* is an abstract type, defined in the SemanticBag DS, which extends SemanticBaseType, while *SemanticType* is defined in the Semantic DS. SemanticType is a concrete type, thus its instances are used for the representation of semantic entity collections.
- *AgentObject DS:* The actors that appear in an audiovisual segment are described here using the extensions of the abstract type *AgentObjectType* that extends SemanticBaseType. Actors in general are represented using the *AgentType*, an abstract type extending AgentObjectType defined in the *Agent DS*. *PersonType*, *OrganizationType* and *PersonGroupType* extend AgentType, are defined respectively

in the *Person DS*, the *Organization DS* and the *PersonGroup DS* and are used for the representation of persons (e.g. a football player), organizations (e.g. a football team) and groups of persons.

- *Object DS:* The *ObjectType* defined here extends SemanticBaseType and is used for the description of objects and object abstractions in the material world (e.g. a table).
- *Event DS:* The *EventType* defined here extends SemanticBaseType and is used for the description of events that take place in a semantic world (e.g. a goal).
- *SemanticState DS:* The *SemanticStateType* defined here extends SemanticBaseType and is used for the description of a state of the world described in an audiovisual segment and the parametric description of its features (e.g. the score in a soccer game before and after a goal).
- *SemanticPlace DS:* The *SemanticPlaceType* defined here extends SemanticBaseType and is used for the description of a place in a semantic world (e.g. Athens).
- *SemanticTime DS:* The *SemanticTimeType* defined here extends SemanticBaseType and is used for the description of semantic time (e.g. Christmas).

We have already mentioned that we currently use MPEG-7 syntax and constructs for ontology definition. Ontology definition support is not directly provided in MPEG-7, so we have to decide about a strategy (framework) for its integration. We could use subtyping of the basic MPEG-7 types in order to describe the ontology types and hierarchies. However, this approach could make our content descriptions unusable by standard MPEG-7 software. So we have decided to completely incorporate the ontology structures within the provided MPEG-7 structures. Fortunately, MPEG-7 provides a mechanism that can be used for the definition of abstract semantic entities. This mechanism is implemented through an attribute defined in SemanticBaseType (and thus inherited in all its extensions), *AbstractionLevel*, of type *AbstractionLevelType*. It has only one attribute, Dimension, of non-negative integer type. When AbstractionLevel is not present in a semantic description, the description refers to specific audiovisual material. When AbstractionLevel is present, abstraction exists in the description. When the Dimension of AbstractionLevel is 0, there exists a description of a semantic entity (e.g. the football player Ronaldo) that is related to every audiovisual segment where the entity appears. When AbstractionLevel has a non-zero Dimension, it can specify classes for the description of abstract semantic entities. The bigger the Dimension of AbstractionLevel, the more extended the abstraction in the description (e.g. Ronaldo is an instance of the "football player" class with AbstractionLevel.Dimension=1, while "football player" is an instance of the "person" class with AbstractionLevel.Dimension=2).

3.2 TV-Anytime based Audiovisual Content Descriptions

In this subsection we describe the constructs provided by the TV-Anytime metadata specification for the definition of content-description metadata for audiovisual information. TV-Anytime provides, through the TV-Anytime metadata series, the TV-Anytime metadata system to enable the end users to find, navigate and manage content from a variety of internal and external sources including, for example, enhanced broadcast, interactive TV, Internet and local storage. It defines a standard way to describe user profiles including search preferences to facilitate automatic filtering and acquisition of content by agents on behalf of the user.

The semantic metadata that describe audiovisual programs and segments are related to the types BasicContentDescriptionType and BasicSegmentDescriptionType respectively. These types include the following common description elements:

- *Title*: A title given to the segment or program. There can be multiple titles for a segment or program.
- *Synopsis*: A textual description of the segment or program content.
- *Keywords*: Important words or phrases, describing the content of the segment or program.
- *RelatedMaterial*: References to external material, related to the content of the segment or program.

Referring to the above it is obvious that an important mechanism in the TV-Anytime metadata system that could be used for semantic content description is the use of keywords or keyword phrases. Taking advantage of this mechanism, it would be possible to support a keyword-based information retrieval mechanism, similar to a Web search engine. The experience from the search engines shows that such a mechanism has limited performance in terms of both precision and recall. In the digital TV environment additional problems appear due to the interface limitations (lack of keyboard, limited resolution). In addition, TV-Anytime provides no explicit support for the definition of domain-specific ontologies.

3.3 Domain-Specific Ontology Integration with MPEG-7 and TV-Anytime

A major emphasis in the work we describe here is in the integration of ontologies with the audiovisual standards. In audiovisual content management environments several tasks are based on the domain-specific ontologies, while the later are the mechanism used for achieving interoperability among standards. We focus here on the role of the domain-specific ontologies and discuss their integration with the MPEG-7 and TV-Anytime standards.

In the framework described here, we provide dual support, for both MPEG-7 and TV-Anytime, through the maintenance of two repositories, on top of a relational database, each of which is compliant with one of these standards. The repositories are indexed consistently, using the same ontology-based tool. This is achieved through the definition of domain-specific ontologies, which are integrated in the repositories. The indexers (people responsible for providing metadata for indexing the segments of the audiovisual content) use terms and relationships from the controlled vocabulary with the precise semantics that the ontology contains to do the indexing. In this work they may use the PC interfaces. The users on the other hand, may often utilize TV-based interfaces. However these interfaces guide the users to select from an existing set of items and relationships through templates and selection mechanisms that guarantee faster interaction and better retrieval performance. Instead of an interface that allows the user to select letters to form keywords, an interface that provides a set of application-specific search templates has been implemented.

In the following we focus our description on the integration of ontologies in MPEG-7 since the steps for integration in the TV-Anytime are similar and the TV-Anytime constructs allowed are just keywords (and therefore the integration becomes even simpler).

As we have explained in section 3.1, we use some basic types and constructs of MPEG-7 to provide ontology support so that we stay always compatible with existing MPEG-7 retrieval and dissemination software. A set of MPEG-7 compliant XML documents contains the entities of interest that have been defined. An extract of an ontology-definition document for soccer games, containing the description of the goalkeeper entity is shown in Example 1.

```
<SemanticBase id='goalkeeper-obj' xsi:type='PersonType'>
     <AbstractionLevel dimension='1'/>
     <Label>
          <Name preferred='true'>Goalkeeper</Name>
          <Name preferred='false'>GK</Name>
          <Name preferred='false'>Goalie</Name>
     </Label>
     <Definition>
          <FreeTextAnnotation>The player positioned directly in front of the goal who tries to stop opposing players to score; the only player allowed to use his hands and arms, though only within the penalty area.</FreeTextAnnotation>
     </Definition>
     <Relation type='urn:...:specializationOf' target='player-obj'/>
</SemanticBase>
```

Example 1: Description of the Goalkeeper entity

In Example 1 a definition of the goalkeeper entity is shown, as well as 3 possible names for the entity, one of which is set as "preferred". The goalkeeper can be related to other entities of the ontology.

After the definition of an entity, it can be instantiated, as is shown in Example 2 for the goalkeeper entity and the entity instances can be associated with specific audiovisual segments.

```
<SemanticBase xsi:type='PersonType'>
     <AbstractionLevel dimension='0'/>
     <Relation xsi:type='urn:...:isExemplifiedBy' target='goalkeeper-obj'/>
     <Agent xsi:type='PersonType'>
          <Name>
               <FamilyName>Kahn</FamilyName>
          </Name>
     </Agent>
</SemanticBase>
```

Example 2: Association of a video segment with the goalkeeper entity

A domain-specific ontology contains entities for the description of both *Application-Specific Metadata* (e.g. the teams and the players taking part in a soccer game) and *Instance-Description Metadata* (e.g. the goals scored in a soccer game). Thus, the contents and structure of a domain-specific ontology are stored in the Semantic Base as the *Stored Ontology*. The ontology also guides the definition of the entity instances corresponding to both the Application Specific Metadata and the Domain Specific Metadata, as shown in Figure 2.

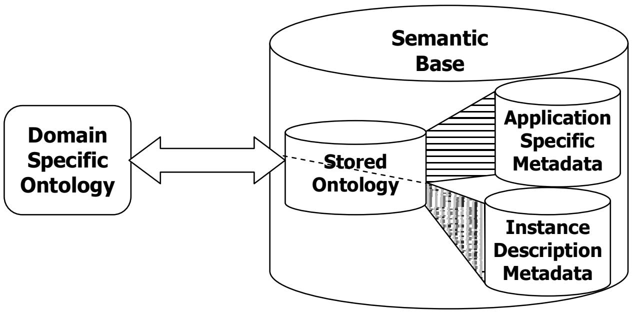

Figure 2: Ontology – Semantic Base Integration

As the various framework components (e.g. the tool used for compatible Semantic Indexing) may interact in different ways with the Application Specific Metadata and the Instance Description Metadata, we have decided to make explicit this separation among the entities defined in the domain-specific ontologies. In order to achieve this, we define for each ontology an ontology-configuration file, where we store:

1. Information for the files containing the definitions of the entities that represent the Application Specific Metadata and the Instance Description Metadata.
2. The type of the entities (Application Specific Metadata vs. Instance Description Metadata) that are defined in each file.
3. Information about the ontology (e.g. ontology specification language, ontology name etc.).

The structure of the ontology configuration file is defined in XML Schema [6] syntax and is shown in Schema 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:element name="OntologyConfiguration">
      <xs:annotation>
         <xs:documentation>Ontology configuration</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="ASMFiles"/>
            <xs:element ref="IDFiles"/>
         </xs:sequence>
         <xs:attribute name="language" type="xs:string" use="optional"
default='MPEG7'/>
         <xs:attribute name="name" type="xs:string" use="required"
form="unqualified"/>
      </xs:complexType>
   </xs:element>
   <xs:element name="ASMFiles">
      <xs:annotation>
         <xs:documentation>Files describing the structure of the Application Specific
Metadata</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="File" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="IDFiles">
      <xs:annotation>
         <xs:documentation>Files describing the structure of the Instance Description
Metadata</xs:documentation>
      </xs:annotation>
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="File" minOccurs="0" maxOccurs="unbounded"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="File">
      <xs:annotation>
         <xs:documentation>File describing the structure of a Metadata category
</xs:documentation>
      </xs:annotation>
      <xs:complexType>
```

```
            <xs:attribute name="filename" type="xs:anyURI"/>
            <xs:attribute name="name" type="xs:string"/>
            <xs:attribute name="type" default="static">
               <xs:simpleType>
                  <xs:list itemType="xs:string"/>
               </xs:simpleType>
            </xs:attribute>
         </xs:complexType>
      </xs:element>
</xs:schema>
```

Schema 1: Ontology configuration Structure

All the files defining entities that describe Application Specific Metadata and Instance Description Metadata are currently using MPEG-7 syntax, so the default value for the ontology specification language is MPEG-7, but as our system is being extended in order to support ontology description languages like OWL, this value may change.

4 Semantic Base

In this section we describe the *Semantic Base* maintained in our framework for the storage of semantic metadata that describe audiovisual content. It is implemented as a part of the framework relational database on top of MySQL. The schema of the underlying database depicts our MPEG-7 compliant core model. The entities comprising this model are of the MPEG-7 types discussed in subsection 3.1. The schema of this part of the Semantic Base is depicted in Figure 3.

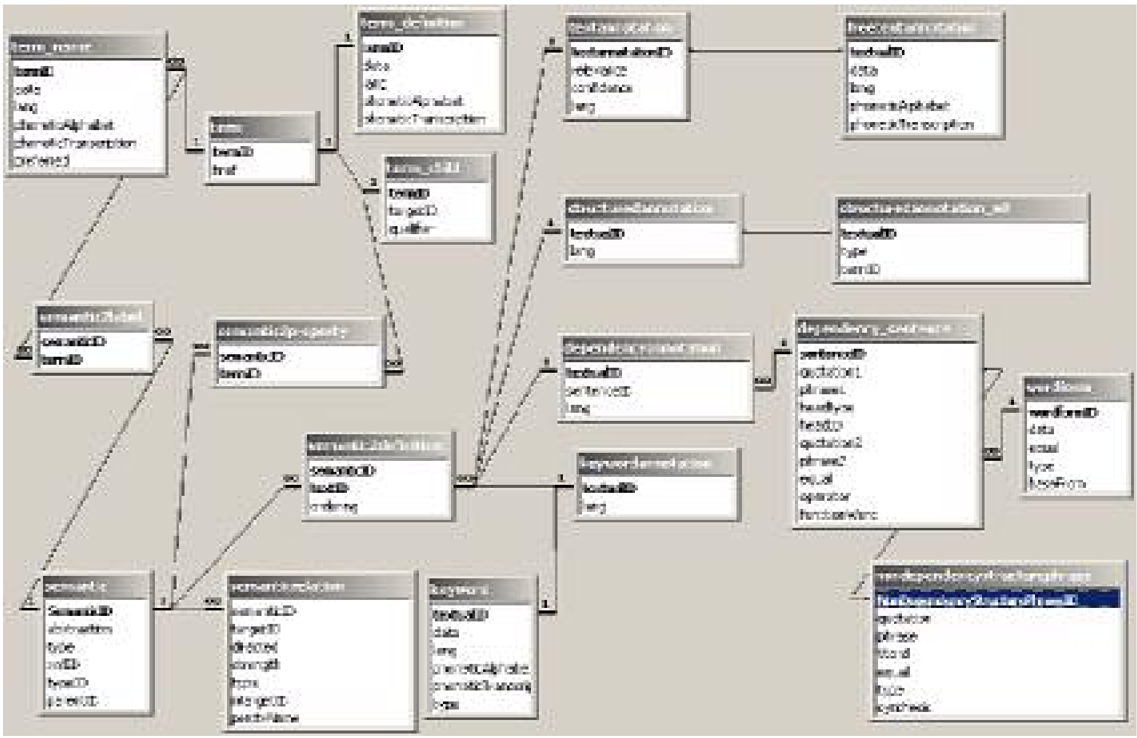

Figure 3: The MPEG-7 compliant part of the Semantic Base schema

In addition to the MPEG-7 fully compliant part of the Semantic Base schema, there exists another part where information related to the audiovisual content but not contained in it (e.g. the age of a player) is represented. This part of the schema, where the Person, Organization, PersonGroup, State, Place and Time entities are defined, is shown in brief in Figure 4.

Due to its semantic content, the Semantic Base can support much more powerful semantic queries than those supported by a keyword only based annotation system. This is achieved through the use of an appropriate query API, which makes use of the Semantic Base contents. The query API we have implemented in order to support semantic queries is described in section 6.

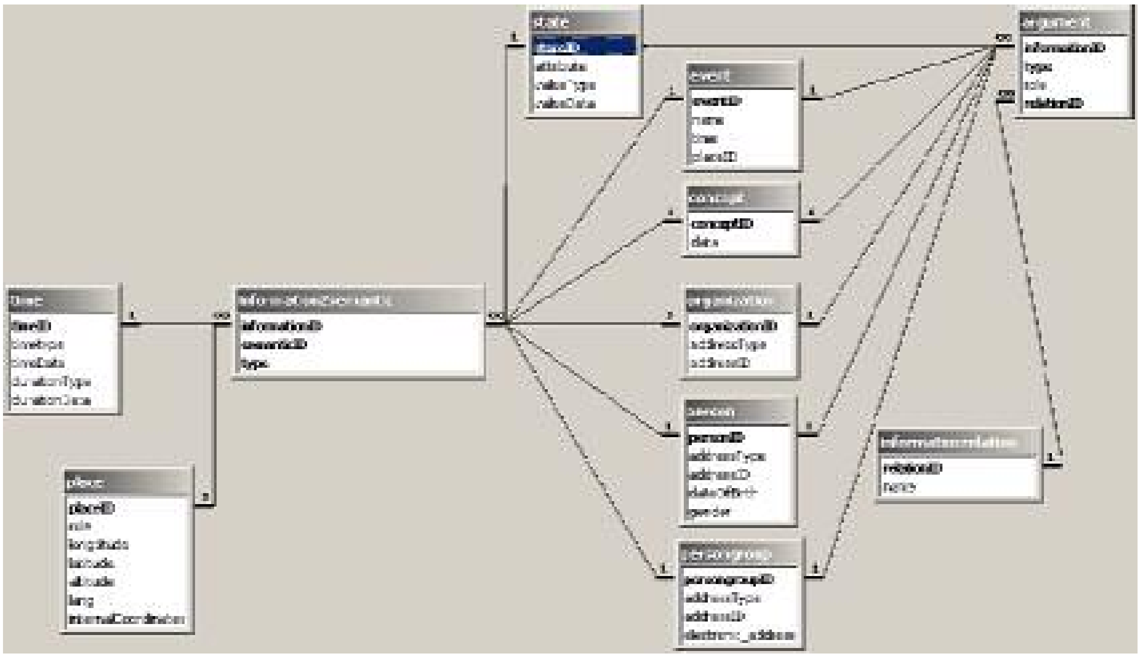

Figure 4: Representation of the additional information

5 Segmentation & Semantic Indexing Tool

In this section we describe our *Segmentation and Semantic Indexing Tool*, used for the segmentation of audiovisual programs – that is, definition of audiovisual program segments – and the semantic description of the defined segments.

The Segmentation and Semantic Indexing Tool is implemented in java.swing, on top of the relational system database. XML data management in the relational database is carried out by a binder capable of fully representing XML documents as objects or objects as XML documents (the serialization of the object tree to XML document is encapsulated in class (un)marshal methods). The data binder uses a SAX-based parser and the corresponding validator can be used to ensure that incoming and outgoing XML documents conform to the TV-Anytime and MPEG-7 XML schemas.

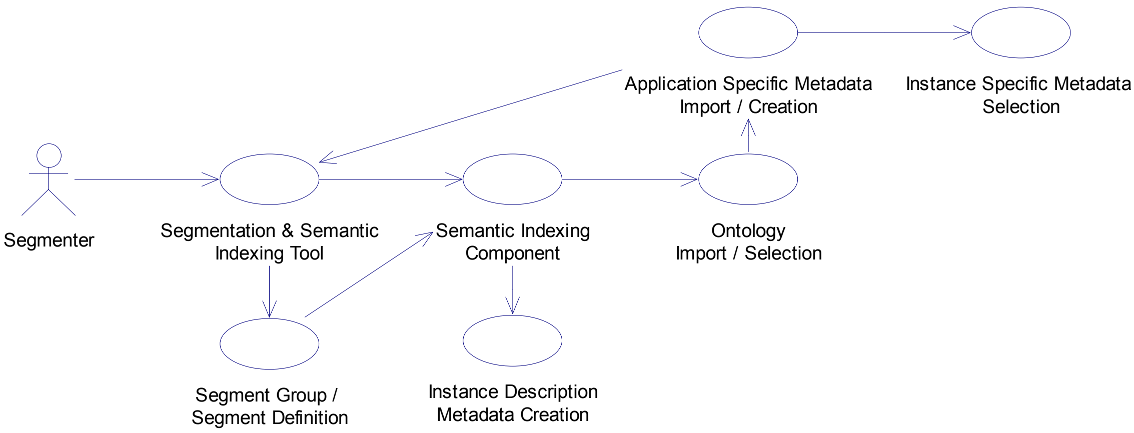

Figure 5: Use Case diagram of the Segmentation & Semantic Indexing Tool

The use-case diagram of Figure 5 illustrates the usage scenario of the Segmentation and Semantic Indexing Tool: The person responsible for audiovisual program segmentation, who is also responsible for program and segment annotation, is the *Segmenter* and uses the Segmentation and Semantic Indexing Tool. At the beginning of a segmentation session, the Segmenter opens the file containing the audiovisual program and is asked to specify the ontology (s)he will use from the available ontologies or to specify the ontology configuration of the ontology (s)he will use, if it isn't already supported by the system. The ontology import is carried out by the *Semantic Indexing Component*, a component of the Segmentation and Semantic Indexing Tool responsible for ontology-based semantic indexing support. Whenever an ontology is imported, the ontology configuration file is parsed and the Application Specific Metadata and

Instance Description Metadata structure definitions are imported from the corresponding files. Then, the ontology structure and contents are stored in the Semantic Base.

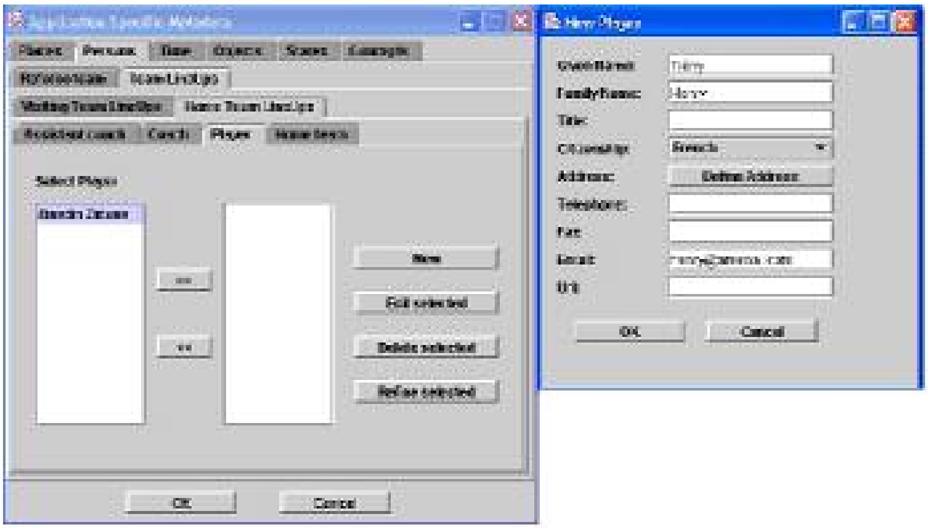

Figure 6: Interface for the definition of Application Specific Metadata for soccer games

After ontology selection/import, according to the structure of the Application Specific Metadata defined in the ontology, an appropriate interface is automatically generated, based on the ontology, which permits the definition of Application Specific Metadata items for the specific audiovisual program or the import of existing ones. The interface generated for the definition of Application Specific Metadata for soccer games is shown in Figure 6. If the user wishes to import existing Application Specific Metadata items, he specifies the corresponding files and the Application Specific Metadata are stored, if they don't already exist, in the Semantic Base, in the same way with the ones defined using the graphical user interface. Then, the segmenter selects, among the Application Specific Metadata, the instance specific metadata (the part of the application specific metadata related with the specific video program).

After the Application Specific Metadata import/definition, the Segmenter proceeds with the definition of audiovisual segment groups and segments. Although the segmentation tool does not perform automatic segmentation, we should note here that there exists support for the segmentation process, through a speech transcription search component. This component takes as input the output of a voice recogniser, which has analyzed and recognized the audio stream of the audiovisual information and has stored the resulting text together with the corresponding timing information. The Segmenter can use the component in order to locate the video frames where a word or phrase is spoken (e.g. "penalty" in a soccer game) and thus to be assisted in the segmentation process.

The transcription tool (shown in Figure 7) assists the user in locating relevant locations in the audiovisual program with the help of the ontologies and the instance specific metadata. The transcription file is automatically processed to identify locations within the transcription, where the transcription contains constructs (words, sets of words in phrases) which may satisfy an ontology instance. Those instances are highlighted in the transcription so that the user can easily browse the transcription file and check if the particular location of the audiovisual material should be segmented and indexed. Alternatively, the user may specify a filter with words that appear in the ontology in order to locate frames in the audiovisual file where potential events of interest happen. Note that the tool allows the user to browse within the transcription file with a filter that contains words that are not part of the ontology; in this case the system will give the user a warning message.

The transcription file is cross-referenced with the audiovisual file so that from any location of the transcription file the corresponding location of the audiovisual file can be accessed and vice versa. This makes it easy to eliminate the "false drops" of the filtering search via direct inspection. The false drops may be the result of the imperfect voice recognition, or of the fact that the filters may be specified in a more "loose" fashion (as a set of keywords instead of concrete semantic constructs). This is done in order to facilitate the indexer to explore areas of the transcription that are of potential interest given that there is no high retrieval effectiveness in plain text search.

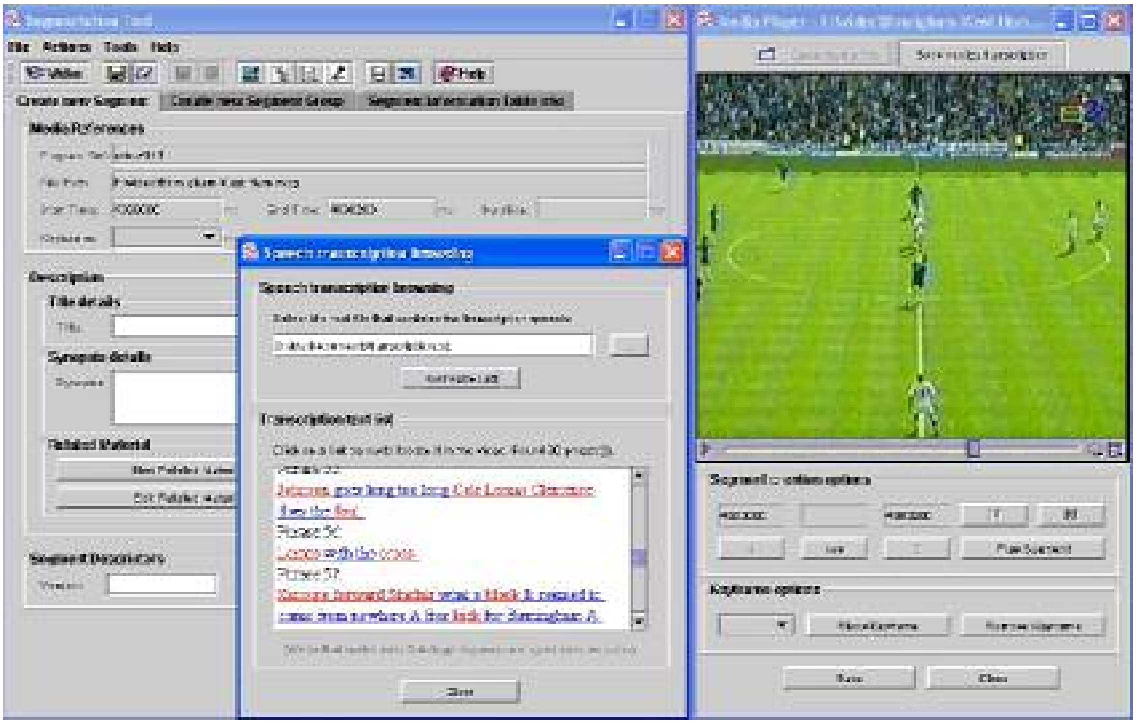

Figure 7: The Transcription Tool

The Segmenter can annotate the audiovisual segments using the interface shown in Figure 8. In order to provide compatible descriptions for the TV-Anytime compliant database, (where simple keywords from the ontology and/or keyword phrases are needed), and the MPEG-7 compliant Semantic Base, segment annotation is carried out using the Semantic Indexing Component. The Semantic Indexing Component is now used for the creation and the storage of Instance Description Metadata items. The Semantic Indexing Component interfaces for the creation of Instance Description Metadata are also automatically generated from the ontology definition files, where the structure of the Instance Description Metadata is described. The definition of an Instance Description Metadata item (a goal in a soccer game) is shown in Figure 8. The segmentation tool allows browsing among the segments specified so far using the instance description metadata items that have been used for the indexing of the segments. The filtering mechanism to locate available segments that are associated with instance description metadata of interest again makes use of the ontologies to facilitate the user.

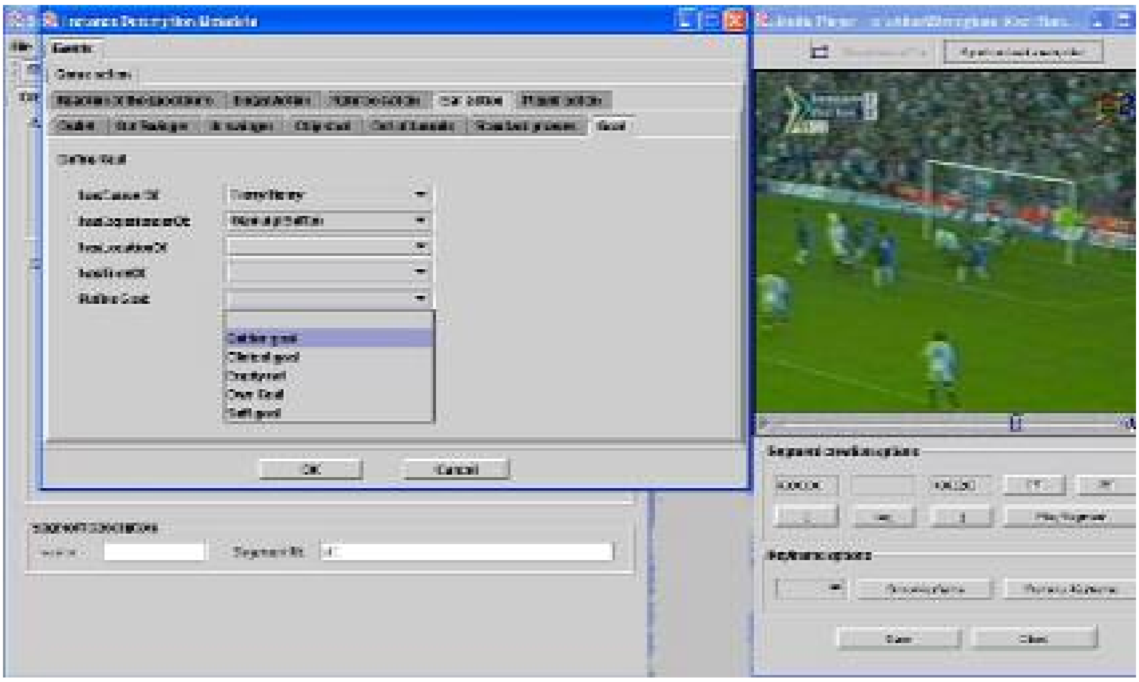

Figure 8: Definition of a goal in a Soccer Game

When the audiovisual program segmentation finishes, a set of Instance Description Metadata items that describe the content of the segmented audiovisual program are stored in the Semantic Base, while the program together with a set of segments and segment groups, annotated with keywords and/or keyword phrases, are stored in the TV-Anytime compliant database.

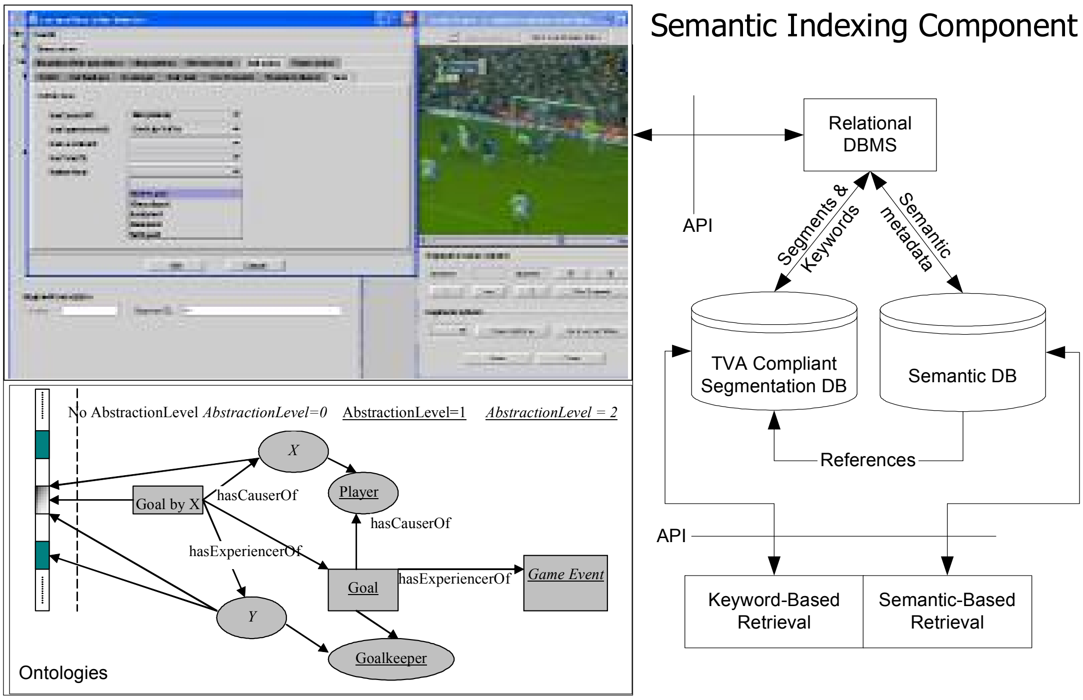

Figure 9: The operation of the Semantic Indexing Component

References from the metadata items stored in the Semantic Base to the related segments stored in the TV-Anytime compliant database are maintained. The keywords in the TV-Anytime database have been inferred from the semantic metadata stored in the Semantic Base. The inference is based on the knowledge encapsulated in domain-specific ontologies and a set of rules stored in the Semantic Base. Each rule is a template, which produces a keyword phrase for each relationship (e.g. for a goal scored by Ronaldo is produced the keyword phrase "Ronaldo scores"). In addition, independent keywords corresponding to the metadata items related to the video segment are stored in the TV-Anytime database (e.g. "Ronaldo" and "goal" for a goal scored by Ronaldo).

The operation of the Semantic Indexing Component of the Segmentation and Semantic Indexing Tool is shown in Figure 9.

6 Semantic Query Support

In this section we discuss the support provided by our system for semantic queries, in addition to the simple keyword-based queries provided by the TV-Anytime compliant part of the framework.

The API does not make any assumptions on the kind of the end-user equipment and interaction capabilities. The application through which the end-user interacts with the rest of the system uses a set of SQL queries that retrieve the IDs of the entities specified in the interface (e.g. retrieve the ID of the Person named "Ronaldo"). Thus, the search API can be used in different environments (e.g. TV, Web), with different user interaction capabilities (e.g. remote control vs. mouse and keyboard): In a TV interface a search template is presented, where only the missing words need to be filled from a list presented in the TV. In this case, the proper items are selected using the remote control. On the other hand, in the Web, a Java application that permits full user interaction will be provided.

The same API and set of interfaces can be used to interact and retrieve audiovisual contents described in both the Semantic Base, which supports the MPEG-7 standard, and the database that supports the TV-Anytime standard, taking into account that both databases are aware of domain-specific, ontology-based extensions of the standards.

The TV-Anytime allows, as mentioned above, indexing based on keywords or keyword phrases mainly. However, the pure TV-Anytime approach to specifying the content utilizing keywords and keyword phrases not only limits the power of the retrieval language, but has also interoperability limitations that become apparent with the new evolutions in technology, standards and business models. The use of domain specific ontologies to limit the possible values of keywords in conjunction with the TV-Anytime metadata alleviates two significant problems:

- Recent TV-Anytime Forum trends emphasize the existence of servers that are connected with high capacity lines with homes and provide additional digital audiovisual channels, but also digital library functionality with multimedia content. This content may be structured according to the MPEG-7 standard to give high discriminatory power to users.
- It is highly possible that digital audiovisual information stored in the TV-Anytime server will also be used by other consumption channels, for example PCs or mobile devices (the later is a scenario also supported by the TV-Anytime Forum). It is also possible that digital audiovisual content will be transferred between servers to support users during travelling. In all the above scenarios high precision and recall of the retrieval mechanism are needed and the limitations of the TV interface do not exist. A more powerful than keyword based retrieval mechanism presents advantages.

Thus, the limitations of the TV-Anytime approach are important and, in order to overcome them, we have developed an API for semantic queries that contains functions allowing the user to pose powerful MPEG-7 queries if he wishes.

The API has been implemented in Java, using JDBC on top of a relational database. It is comprised of two functions that are capable of supporting queries that contain one or more query terms combined with the logical operators AND and OR. The API functions are listed below:

1. *GetSegment((AND|OR)? (informationID informationType semanticID)+)*, where:
   - The first argument denotes the logical operator (AND/OR) used for the combination of the query terms in the rest of the query. The argument can appear at most once and may be omitted if there is only one query term.
   - The second argument is a set of triplets that describe the query terms, that is the metadata items that should be related to the query results: *informationID* is the id of a metadata item, *informationType* is the type of the metadata item and *semanticID* is the role of the metadata item. There must appear at least one triplet in a query. The "null" value should be used when the user doesn't want to set a criterion for any part of the triplet.

   The GetSegment function permits queries that retrieve audiovisual segments related to specific metadata items. All the metadata items are treated as equivalent query terms. No relationships among the metadata items can be defined, while the query terms may be combined using the AND and OR logical operators.

   In Table 1 we present some examples of queries expressed using GetSegment together with their free text descriptions. We assume that the informationID of the person named "Ronaldo" is 1 and the informationID of the person named "Beckham" is 2.

Query	**Description in Free Text**
`GetSegment(1 Person null)`	"Give me the segments where Ronaldo appears" (not only as a player!)
`GetSegment(1 Person player-obj)`	"Give me the segments where the player Ronaldo appears"
`GetSegment(AND, (1 Person player-obj) (null Event goal-event))`	"Give me the segments where the player Ronaldo appears and a goal is scored" (the scorer may be any player)
`GetSegment(OR, (1 Person player-obj) (2 Person player-obj))`	"Give me the segments where the players Ronaldo or Beckham appear"

Table 1: Examples of queries expressed using GetSegment

2. *GetSegmentMQT(informationID informationType semanticID (AND|OR) (informationID informationType semanticID relationType)*)*, where:
   - The first argument is a triplet that describes the main query term, which is a metadata item that is related to the other query terms: *informationID* is the id of the main query term, *informationType* is the type of the main query term and *semanticID* is the role of the main query term.

- The second argument denotes the logical operator (AND/OR) used for the combination of the query terms expressed in the rest of the query. The argument can appear at most once and may be omitted if no query terms follow.
- The third argument is a set of quadruplets that describe the query terms that are related to the main query term: *informationID* is the id of the metadata item, *informationType* is the type of the metadata item, *semanticID* is the role of the metadata item and *relationType* is the type of the relationship of the current metadata item with the main query term. There can appear any number of quadruplets in a query. The "null" value should be used when the user doesn't want to set a criterion for any part of the quadruplet.

The GetSegmentMQT function permits queries that retrieve audiovisual segments that relate to a specific metadata item that is the main query term. The main query term may be related, through specific relationships, to the other metadata items (including the information related to them that is not present in the described audiovisual segment, a feature of special interest in Digital Library environments). The metadata items related to the main query term are combined using the AND and OR logical operators.

In Table 2 we present examples of queries expressed using GetSegment, where we assume that the informationID of the person named "Ronaldo" is 1 and the informationID of the person named "Kahn" is 3. We also assume that the Greek citizenship has citizenshipID 1.

Query	Description in Free Text
`GetSegmentMQT(null null goal-event AND 1 Person player-obj hasCauserOf)`	"Give me the segments where the player Ronaldo scores"
`GetSegmentMQT(null null goal-event AND (1 Person player-obj hasCauserOf) (3 Person player-obj hasPatientOf))`	"Give me the segments where the player Ronaldo scores against the goalkeeper Kahn"
`GetSegmentMQT((null Person player-obj) AND (1 citizenship person2citizenship))`	"Give me the segments where a player with Greek citizenship appears"

Table 2: Examples of queries expressed using GetSegmentMQT

The combination of the API functions (through the union or intersection of the results of more than one queries) can support even more complex queries. Thus, powerful queries in several platforms can be supported. The integration of the API to the end-user applications remains a challenge, especially in the Digital TV environment.

7 Application in Soccer Games

We describe in this section an application in the soccer domain, which we developed in order to evaluate our framework. In the context of the application we developed an ontology for the description of soccer games and used it for indexing and retrieval of soccer games that have been stored in the system database. The definition of the ontology is based on the rules of FIFA (the international organization responsible for the definition of football rules) [8] and other sources.

We first set up an ontology configuration according to the XML Schema of Schema 1. We have specified in the ontology configuration, shown in Example 3, that the Application Specific Metadata in the context of a soccer game are time periods, places and actor roles (persons), while the Instance Description Metadata are events.

```
<?xml version="1.0" encoding="UTF-8"?>
<OntologyConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation=" ontology_config.xsd" name="soccer" language='"MPEG7"'>
   <ASMFiles>
      <File filename="places.xml" name="Places" type="static"/>
      <File filename="persons.xml" name="Persons" type="dynamic"/>
      <File filename="times.xml" name="Time" type="static"/>
   </ASMFiles>
   <IDFiles>
      <File filename="events.xml" name="Events" type="dynamic"/>
   </IDFiles>
</OntologyConfiguration>
```

Example 3: Soccer Ontology Configuration

The files containing the description of the entities and structure of both the Application Specific Metadata and the Instance Description Metadata use the MPEG-7 syntax and constructs described in subsection 3.1.

The Application Specific Metadata entities defined in the soccer ontology are described in subsection 7.1, while the Instance Description Metadata entities are discussed in subsection 7.2. Finally, we describe the applications supported by the soccer ontology in subsection 7.3.

7.1 Application Specific Metadata for Soccer Games

In this subsection we refer to the Application Specific Metadata entities defined in the soccer ontology. As mentioned above, the Application Specific Metadata in the context of a soccer game belong to the following categories:

Time: Abstract time entities, referring to the time periods in a soccer game fall in this category. The part-of hierarchy of the ontology terms for the time periods of a soccer game is shown in Figure 10.

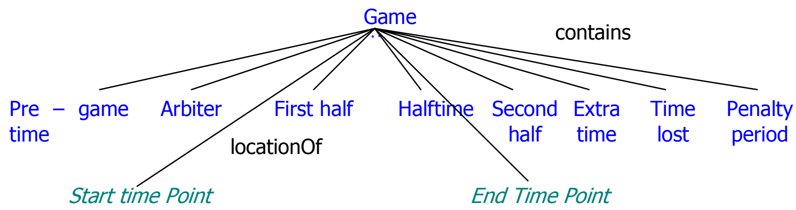

Figure 10: The ontology terms for time periods in a soccer game

As shown in Figure 10, there exist the *Pre–game Time*, the *Arbiter*, the *First half*, the *Halftime*, the *Second half*, the *Extra time*, the *Time lost* and the *Penalty period*. The *Start Time Point* is the beginning of the First half, while the *End Time Point* is the end of the Second half.

Places: Abstract place entities, referring to the places a soccer game takes place fall in this category. Soccer games usually take place in a stadium. In order to make possible the support of more accurate queries (e.g. events near the halfway line), we have defined a part-of hierarchy of the ontology terms for the stadium regions, as shown in Figure 11. According to Figure 11, the ontology terms for the stadium regions are:

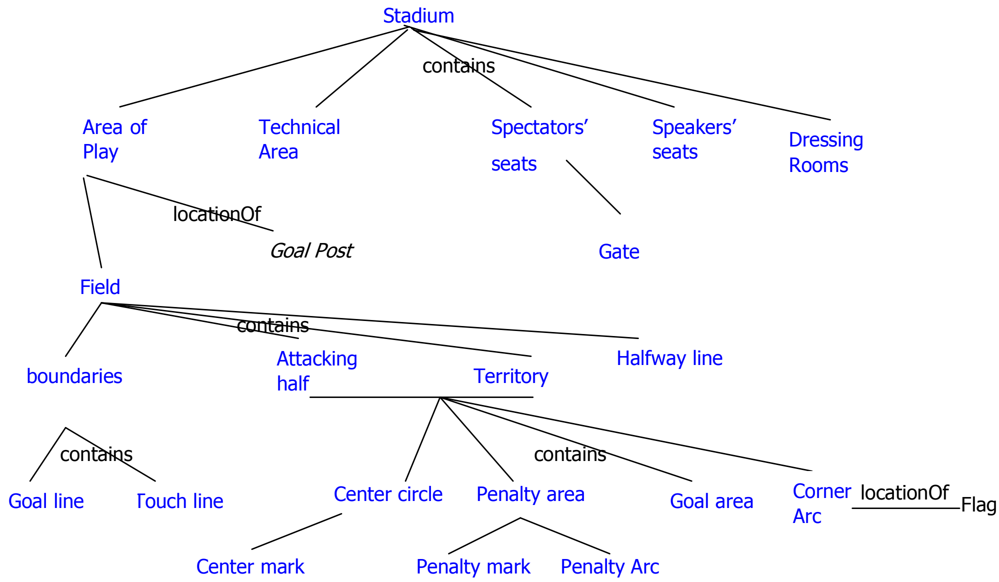

Figure 11: The part-of hierarchy of the ontology terms for regions of the stadium

1. The *Area of Play*. It contains the *Field* and the *Goal Posts*. The field's *Boundaries* are the *Goal Line* and the *Touch Line*. There exist the team *Territories* and their *Attacking Halves* (the territory of a team is the opponent's attacking half). In the field exist the *Center Circle* (that contains the *Center Mark*), the *Penalty Areas* (attached to the *Penalty Arcs*) and the *Corner Arcs* (where the *Corner Flags* are located).
2. The *Technical Area*.
3. The *Spectator's Seats*, including the fans' *Gates*.
4. The *Speaker's Seats*.

5. The *Dressing Rooms*.

Actor Roles: Abstract entities, referring to the roles of the soccer game actors (e.g. teams, players etc.) fall in this category. The ontology entities for the representation of person roles in a soccer game are shown in Figure 12, while the entities and relationships for the description of teams are depicted in Figure 13.

According to Figure 12, the roles in a soccer game are *Referee*, *Coach*, *Player*, *Spectator* and *Chairman*. A referee instance may be the *Principal referee*, an *Assistant referee* or a *Fourth official*. A Player instance may be a *Goalkeeper*, a *Substitute* or may be in a specific place (e.g. *Left Forward*), which refines his role. The roles are represented by abstract MPEG-7 Person entities (AbstractionLevel=2).

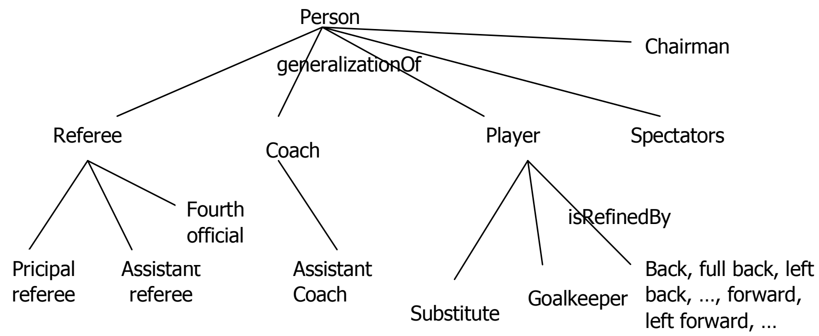

Figure 12: The ontology terms for person roles in a soccer game

As shown in Figure 13, a team, represented as an MPEG-7 Organization instance, may be *Visiting Team* or *Home Team*. Team players comprise PersonGroups.

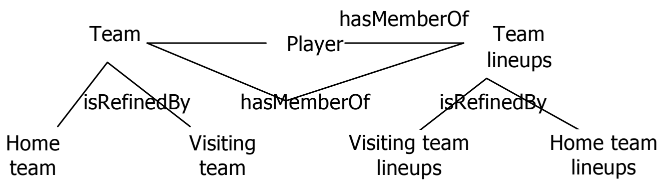

Figure 13: The ontology terms for the teams participating in a soccer game

7.2 Instance Description Metadata for Soccer Games

In this subsection, the Instance Description Metadata in the context of soccer games are described. The Instance Description Metadata are events, as the soccer game descriptions are usually event-driven. The events that take place in a soccer game are specified in the FIFA rules [8]. An event may fall in one of 3 categories: *Referee Actions*, *Game Actions* and *Miscellaneous*. We implement them in our model as MPEG-7 Event instances. As the events are too many (123) to fit in a diagram, their names and the category that each one falls are listed in Table 3.

Referee Action	Principal Referee Action, Whistle, Red Card, Book a player, Yellow Card, Advantage Rule, Warning, Assistant Referee Action, Raise flag, Fourth Official Action
Game action	Interact with a player, Hit the ball, Pass, Shoot, Trap, Player action, Ball action, Clear, Illegal Action, Foul, Off-side, Dangerous play, Half volley, Heel, Toed Ball, Juggling, Deadening the ball, Trap, chest, Trap, foot, Kick, Trap, thigh, Diving, First time, Flick, Bicycle kick, Scissor kick, Backswing, Chip shot, Banana kick, In swinger, Out Swinger, Missile, Pass, back, Pass, front, Power kick, Outlet, Assist, Cross, Through Ball, Pass, drop, Early Service, Pass, forward, Pass, hospital, 50/50 ball, Pass, lead, Nutmeg, Pass, push, Send, Hacking, Hand-to-ball, Obstruction, Carrying the ball, Header, Instep drive, Volley, Pass, square, Pass, hopped, Goal, Golden goal, Sudden death, Clinical goal, Substitution, Switch, Fake, Feint, Close down, Re-start, kick-off, Penalty kick, Free kick, Dropped Ball, Indirect free kick, Direct free kick, Corner kick, Goal kick, Throw-in, Retake, Injury, Jump, Fall, Ball to Hand, Reflection
Miscellaneous	Reaction of the spectators, Applause, Boo, Encourage, Sing, Wave flags, Throw items, Give instructions, Medical treatment, Talk, Fight, Hit, Shake hands, Cheer

Table 3: The ontology terms for the events taking place in a soccer game

Soccer game events are related with the entities that are dominant for their features. For example, a goal is related, as shown in Figure 14, to:

- The scorer (who is a Player) through the "hasCauserOf" relationship.
- The Goalkeeper through the "hasExperiencerOf" relationship.
- The Field through the "hasLocationOf" relationship.
- The Game Time, where the exact time (relative to the game start) the goal was scored is denoted, through the "hasTimeOf" relationship.

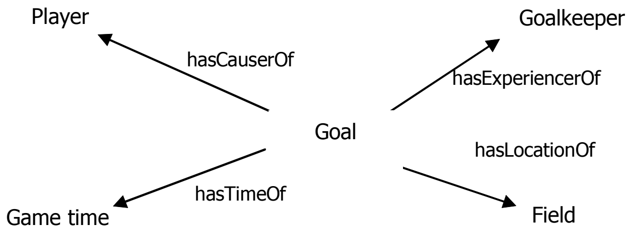

Figure 14: The goal event and its relationships with other entities

7.3 Applications

We discuss in this subsection the applications of the soccer ontology we developed. We have used the soccer ontology to segment and index soccer games. The ontology was imported in the system database with the Ontology Import Component of the Semantic Indexing Component shown in Figure 1. The application specific metadata (players, referees, coaches etc.) were inserted in the database using the Application Specific Metadata Import Component of the Semantic Indexing Component (also shown in Figure 1). The soccer videos were segmented and indexed using the Segmentation and Semantic Indexing Tool (due to the lack of an appropriate voice recogniser the transcriptions were imported by hand). Such a recognizer is described in [6] [27]. The segmentation tool was proven to be very easy to use, and the help of the transcriptions in the indexing process significant. The indexing was done using the ontologies and the application specific metadata, and guarantees correctness in the indexing process, speed through browsing and selection and effective retrieval. The retrieval interfaces took advantage of the ontologies and the application specific metadata for correctness checking in the query specification. Selection through a list of options was the user interface metaphor used for query specification. Although the retrieval interface was implemented on PCs, this metaphor is also appropriate for interfaces on TV sets using a remote control mechanism. The retrieval effectiveness with the use of the soccer ontologies on top of MPEG-7 and TV-Anytime is very high in terms of precision and recall in comparison with keyword based filters.

We should mention here that, given another domain-specific ontology (e.g. Formula 1) the appropriate interfaces for the definition of application specific metadata and instance description metadata would be generated automatically, without any modifications in our system.

8 Conclusions – Future Work

We have described a framework for the support of ontology-based semantic indexing and retrieval of audiovisual content. The development of the framework has been based on the MPEG-7 standard specifications for metadata descriptions, but is also applicable for content-description metadata that have been structured according to the TV-Anytime standard. Our work provides a methodology to enhance the retrieval effectiveness of audiovisual content, while maintaining compatibility interoperability of the tools using those standards.

In our framework, domain-specific ontologies guide the definition of both the application-specific metadata and the instance-description metadata that describe the contents of audiovisual programs and/or segments. We opted for an integration methodology, which uses only types and structures of the standards, so that the resulting metadata descriptions are usable by any existing retrieval or dissemination software that is based on the standards. The ontologies are also used in order to provide compatible descriptions in both standards (MPEG-7 and TV-Anytime) for the same content. This approach allows indexing compatibility and interoperability of TV-Anytime and digital library applications.

The components of the system supporting this framework include a segmentation tool for segmenting audiovisual information, which also provides ontology-based semantic indexing capabilities, and an appropriate API for semantic query and user profile definition support. Queries in the one standard can be translated in queries that use the other standard (possibly resulting in less effective retrieval). A user interface for semantic queries based on the MPEG-7 and TV-Anytime standards has also been implemented. The segmentation tool uses the ontologies for accelerating the segmentation process, as well as for doing more effective semantic indexing of programs and segments. In order to test our framework ideas and implementation, an application testbed for the domain of football games has been developed on top of this system. An ontology for football games has been defined and used for indexing and retrieval of segments of soccer games that have been stored in the system database.

The methodology we developed opens up a wide opportunity for the creation of MPEG-7 and TV Anytime services offering structured domain-specific ontologies that can be integrated to these standards for enhancing multimedia retrieval performance. Using this methodology, applications and tools that use these standards increase their effectiveness while remaining fully compatible to the standards.

Our current research in the audiovisual service provision domain, in addition to the on-going integration of OWL ontologies with our system [28], includes:

- Investigation of system performance and automatic adaptation oriented aspects in our framework.
- Definition of ontologies for other application domains (e.g. news) and the adaptation of existing ones in order to enhance the capabilities of our system. Complexity and cost for these tasks will be investigated.
- The deployment of the framework in order to provide automatic (or semi-automatic) abstractions of the audiovisual content, based on the domain-specific ontologies and the corresponding semantic information stored in the semantic base.
- Investigation of domain specific fuzzy query and user interface aspects.

9 References

1. Q. Al-Khatib, F. Day, A. Ghafoor, B. Berra, "Semantic Modelling and Knowledge Representation in Multimedia Databases", IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 1, pp. 64-80, 1999
2. A. Analyti and S. Christodoulakis, "Multimedia Object Modelling and Content-Based Querying", in Proc. Advanced Course Multimedia Databases in Perspective, Twente, Netherlands, 1995, pp. 213-238
3. Carnegie Mellon University, "Informedia project", http://www.informedia.cs.cmu.edu
4. S. Christodoulakis, N. Pappas, G. Kyriakaki, I. Maragoudakis, Y. Mavraganis, C. Tsinaraki, "The KYDONIA Multimedia Information Server", in Proc. ECMAST, Milan, Italy, pp. 491-506, 1997
5. DELOS Network of Excellence in Digital Libraries, http://delos-noe.iei.pi.cnr.it/
6. S. Eickeler, K. Biatov, M. Larson, J. Koehler, "Two novel applications of speech recognition methods for robust spoken document retrieval", Chania, Greece, 2003, MDL workshop proceedings, http://www.music.tuc.gr/MDL/
7. D. Fallside, "XML Schema Part 0: Primer", W3C Recommendation, 2001, http://www.w3.org/TR/xmlschema-0/
8. Federation Internationale de Football (FIFA), "Football Game Rules", 2003, http://www.fifa.org
9. J.Gemmell, J.Han, R. Beaton, S. Christodoulakis, "Delay-Sensitive Multimedia on Disks", IEEE Multimedia 1(3), pp. 56-67, 1994.
10. W. Grosky, "Managing Multimedia Information in Database Systems", Communications of the ACM, Volume 40, Number 12, pp. 73-80, 1997
11. N. Guarino, "Formal Ontology and Information Systems", in Proc. of FOIS '98, Trento, Italy, 1998, pp. 3-15
12. D. Mc Guiness, F. van Harmelen, "OWL Web Ontology Language Overview", W3C Candidate Recommendation, 2003, http://www.w3.org/TR/owl-features/
13. M-S. Hacid, C. Decleir, J. Kouloumdjian, "A Database Approach for Modeling and Querying Video Data", IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 5, 2000
14. IBM Japan, "Video Enrichment project", http://www.tri.ibm.com/projects/video/index_e.htm
15. ISO/IEC JTC 1/SC 29/WG 11/N3966, "Text of 15938-5 FCD Information Technology – Multimedia Content Description Interface – Part 5 Multimedia Description Schemes", Singapore, 2001
16. A. Jaimes, T. Echigo, M. Teraguchi, F. Satoh, "Learning Personalized Video Highlights from Detailed MPEG-7 Metadata", in Proc. of ICIP, Rochester, NY, USA, 2002
17. H. Jiang, A. Elmagarmid, "WVTDB – A Semantic Content-Based Database System on the World Wide Web", IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 6, pp. 947-966, 1998
18. F. de Jong, T. Westerveld, "MUMIS: Multimedia Indexing and Searching", in Proc. of CBMI Workshop, Brescia, Italy, 2001, pp. 423-425
19. F. Kazasis, N. Moumoutzis, N. Pappas, "Informative Annex: Generic Architecture for handling TVA Metadata using relational database technologies", contribution TV-Anytime Forum/Metadata WG, 2nd Implementer's workshop, Geneva, 2002

20. F. Kazasis, N. Moumoutzis, N. Pappas, A. Karanastasi, S. Christodoulakis, "Designing Ubiquitous Personalized TV-Anytime Services", in Proc. of CAiSE workshops (UMICS), Velden, Austria, 2003, pp 136-149
21. Kyriakaki G., "MPEG Information Management Services for Audiovisual Applications", Master Thesis, Technical University of Crete, 2000
22. C. Brandt, G. Kyriakaki, W. Lamotte, R. Lüling, I. Maragoudakis, Y. Mavraganis, K. Meyer, N. Pappas, "The SICMA multimedia server and the virtual museum application", Proceedings of the European Conference on Multimedia Applications Services and Techniques – ECMAST 98, Berlin, 1998, pp. 85-97
23. MPEG Group, "MPEG-7 (Multimedia Content Description Interface)", http://www.chiariglione.org/mpeg/index.htm
24. The OntoWeb project, http://www.ontoweb.org/
25. N. Pappas, F. Kazasis, N. Moumoutzis, C. Tsinaraki, S. Christodoulakis, "Personalized and Ubiquitous Information Services for TV Programs", MDL workshop proceedings, Chania, Greece, 2003, http://www.music.tuc.gr/MDL/
26. N. Pappas, S. Christodoulakis, "Design & Development of a Stream Service in a Heterogeneous Client Environment", in Proc. VLDB, Cairo, Egypt, 2000, pp. 578-579
27. R. Singh, M. Seltzer, B. Raj, R. Stern, "Speech in Noisy Environments: Robust Automatic Segmentation, Feature Extraction and Hypothesis Combination", ICASSP, Salt Lake City, UT, 2001
28. C. Tsinaraki, S. Christodoulakis, "Integration of OWL ontologies in MPEG-7 and TV-Anytime compliant Semantic Indexing", submitted for publication
29. C. Tsinaraki, E. Fatourou, S. Christodoulakis, "An Ontology-Driven Framework for the Management of Semantic Metadata describing Audiovisual Information", in Proc. of CAiSE, Velden, Austria, 2003, pp 340-356
30. C. Tsinaraki, S. Papadomanolakis, S. Christodoulakis, "A Video Metadata Model supporting Personalization & Recommendation in Video-based Services", in Proc. of MDDE Workshop, Lyon France, 2001, pp. 104-109
31. C. Tsinaraki, S. Papadomanolakis, S. Christodoulakis, "Towards a two - layered Video Metadata Model", in Proc. of DEXA Workshop - DLib, Munich, Germany, 2001, pp 937-941
32. TV-Anytime Forum, http://www.tv-anytime.org/
33. UP-TV project site, http://www.up-tv.de/