ONTOLOGY BASED SEMANTIC RETRIEVAL OF VIDEO CONTENTS USING METADATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SAMET AKPINAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2007

Approval of the thesis:

ONTOLOGY BASED SEMANTIC RETRIEVAL OF VIDEO CONTENTS USING METADATA

submitted by **SAMET AKPINAR** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ______________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Volkan Atalay ______________________
Head of Department, **Computer Engineering**

Assoc. Prof. Dr. Ferda Nur Alpaslan ______________________
Supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. İlyas Çiçekli ______________________
Computer Engineering Dept., BİLKENT

Assoc. Prof. Dr. Ferda Nur Alpaslan ______________________
Computer Engineering Dept., METU

Assoc. Prof. Dr. Nihan Çiçekli ______________________
Computer Engineering Dept., METU

Dr. Ayşenur Birtürk ______________________
Computer Engineering Dept., METU

Certificated Engineer Semra Doğandağ ______________________
ASELSAN

Date: (Thesis defense date)

07.09.2007

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Samet Akpınar

Signature :

ABSTRACT

ONTOLOGY BASED SEMANTIC RETRIEVAL OF VIDEO CONTENTS USING METADATA

Akpınar, Samet
M.Sc., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Ferda Nur Alpaslan

September 2007, 65 pages

The aim of this thesis is the development of an infrastructure which is used for semantic retrieval of multimedia contents. Motivated by the needs of semantic search and retrieval of multimedia contents, operating directly on the MPEG-7 based annotations can be thought as a reasonable way for meeting these needs as MPEG-7 is a common standard providing a wide multimedia content description schema. However, it is clear that the MPEG-7 formalism is deficient about the semantics and reasoning support. From this perspective, additionally, we need to represent MPEG-7 descriptions in a new formalism in order to fill the gap about semantics and reasoning. Then, the semantic web and multimedia technologies intercept at this point of multimedia semantics.

In this thesis, OWL Web Ontology Language, which is based on description logic, has been utilized to model a connection between the ontology semantics and video metadata. Modeling the domain of the videos using ontologies and the MPEG-7 descriptions, and reasoning on the videos by the help of the logical formalism of these ontologies are the main objectives of the thesis.

Keywords: MPEG-7, Semantic Web, Ontology, OWL, Description Logic

ÖZ

VİDEO İÇERİKLERİNİN ÜST DÜZEY BİLGİLER KULLANILARAK
ONTOLOJİ TABANLI ANLAMBİLİMSEL ÇIKARIMI

Akpınar, Samet
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Ferda Nur Alpaslan

Eylül 2007, 65 sayfa

Bu tezin amacı, video içeriklerinin anlambilimsel çıkarımı için kullanılacak bir altyapının geliştirilmesidir. MPEG-7'nin geniş bir çoklu ortam içerik tanımlama şeması sağlayan genel bir standart olması sebebiyle ve çoklu ortam içerikleri için anlambilimsel arama ve çıkarım ihtiyaçlarını da harekete geçirmesiyle, doğrudan MPEG-7 tabanlı etiketlemelerin üzerinde işlem yapmak bu ihtiyaçları karşılamak için makul bir yol olarak düşünülmektedir. Fakat, şu çok açıktır ki MPEG-7 biçimselliğinin anlambilimsellik ve akıl yürütme desteklerini sağlama konusunda eksiklikleri vardır. Bu perspektiften bakıldığında, anlambilimsellik ve akıl yürütme konularındaki boşluğu doldurmak için MPEG-7 tanımlamalarına ek olarak yeni bir biçimsellikteki gösterime ihtiyacımız vardır. Bu durumda, anlambilimsel ağ ve çoklu ortam teknolojileri çoklu ortam anlambilimselliğinin tam da bu noktasında kesişmektedir.

Bu tezde, ontoloji anlambilimselliği ve bir çoklu ortam örneklemesi olarak video üst düzey bilgileri arasındaki bağlantının modellenmesi için tanımlama mantığına dayalı

OWL Ağ Ontoloji Dili'nden faydalanılmıştır. Videoların, kendi MPEG-7 tanımları kullanılarak ontolojiler olarak modellenmesi ve videolar üzerinde ontolojilerin mantıksal biçimsellikleri kullanılarak akıl yürütme işlemlerinin gerçekleştirilmesi tezin ana hedefleridir.

Anahtar Kelimeler: MPEG-7, Anlambilimsel Ağ, Ontoloji, OWL, Tanımlama Mantığı

To My Parents

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Ferda Nur Alpaslan for her guidance, advice, criticism, encouragements and insight throughout the research.

I would also like to thank Assoc. Prof. Dr. Nihan Çiçekli for her suggestions and comments.

The technical assistance of Mr. Özgür Alan and Mr. Orkunt Sabuncu are gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Overview

As a consequence of recent advances of World Wide Web, dramatic increases in the amount of multimedia data have revealed the needs for the semantic retrieval techniques of multimedia contents as the high amount of multimedia data needs high level of management facilities. Not only the amount of multimedia data but also the user desires about gaining the ability of querying the individual multimedia data semantically plays an important role on the researches in the field. That desire basically results from the personalization needs of the users. In other words, the complexity of computer users' management needs on multimedia contents causes them to direct themselves into a way in which the multimedia contents are managed according to the personal interest areas.

In this thesis, video data is used as a case study of multimedia data. Video contents composed of raw video data and video metadata are the main building block elements of the infrastructure. Video metadata exist for enriching the raw video data and it is generated manually using an annotation tool instead of the automatic metadata extraction techniques such as text mining, video processing, and speech recognition. In consideration of being mainly focused on the semantic retrieval of video contents, development of metadata extraction techniques and building advanced annotation tools using these techniques are out of scope of this thesis.

Before generating the metadata, we firstly need to determine a video data model which will be used to model the video contents. According to different application requirements, we can present different modeling techniques. Models based on physical level segmentation defend the idea of handling the video in small pieces that are created according to different techniques. On the other hand, models based on annotation layering aims to segment the videos semantically and assigning an annotation to each segment describing its semantic content. Another approach defending models based on video objects applies the object-oriented approach into the modeling tasks of the semantics of video contents [21].

According to the projected infrastructure in the thesis, assigning video segments into semantic content descriptions comes into prominence. As the study is focused on the retrieval of video contents with MPEG-7 based metadata, a video data model based on annotation layering is favorable. Video metadata are the annotations describing the contents of video segments in the data model. In order to describe the contents semantically, we need to represent the annotations more structured than the free texts. Therefore, using semantic entities in the annotations helps us as an improvement for the data model. “Objects” and “Events” are the primitives used to structure the annotations for semantics. Objects are the entities describing the concepts while events are used to model the relations between the concepts and describing the actions in the video segments.

It is also important to determine the formalism which is used for the representation of video metadata as well as the data model. In order to be based on a strong and standardized video metadata representation formalism, a common standard providing a wide multimedia content description schema is needed. MPEG-7, formally named "Multimedia Content Description Interface", is such a standard developed by Moving Pictures Expert Group (MPEG) providing a rich set of standardized tools to describe multimedia content. Both human users and automatic systems that process audiovisual information are within the scope of MPEG-7. MPEG-7 offers a comprehensive set of audiovisual Description Tools (the metadata elements and their structure and relationships, that are defined by the standard in the form of Descriptors and Description Schemes) to create descriptions which will form the basis for applications enabling the needed effective and efficient access (search, filtering and browsing) to multimedia content [11].

Although MPEG-7 formalism presents wide description options, it is deficient about the semantics and reasoning supports. First of all, MPEG-7 does not have a logical background enabling the definition of semantic relations and reasoning on the concept descriptions and relations. Secondly, not having reasoning support causes MPEG-7 formalism to be weak in the strong semantic retrieval of the multimedia contents. Thus, another formalism that is based on logic is needed.

At this point, semantic web appears as a solution by providing the mentioned formalism. “Ontology” is the key concept for semantic web. From this point of view, ontologies are used to model the video domain semantically concerning the projected infrastructure. In order to carry out the representation of the ontologies, different semantic web standards (RDF, RDFS, OWL, etc.) are used. Regarding the properties of these standards, OWL, Web Ontology Language, is suitable for our problem. As it is based on description logic, it provides semantics and reasoning in order to fill the gap of MPEG-7 in semantics.

Therefore, the querying actions searching the videos are processed using the reasoning procedures working on the ontology instances of each video represented by SWRL. In other words, the answers for the queries on the video contents are extracted by reasoning on the logical representation scheme created from MPEG-7 annotations of the videos and the ontology of the video domain. This SWRL representation provides a structure enabling the usage of OWL with RULE-ML.

In this thesis, the aim is to develop an infrastructure to implement the semantic retrieval of video contents. The infrastructure is based on annotation layering as the data model. Annotations are created using an annotation tool based on MPEG-7 standards and they are mapped into the video frames according to the data model. These MPEG-7 based annotations use semantic entities, i.e. objects-events, in order to define the concepts. Modeling the domain of the videos in OWL using ontology management tools is an important phase in developing the infrastructure. The domain of "soccer videos" is selected as the case study in the thesis. Because the soccer domain is modeled as an ontology, creating ontology instances from the video frames annotated with MPEG-7 based metadata is another important step in the development process of the infrastructure. After acquiring the ontological descriptions of the frames in OWL, frames can be queried using their ontological descriptions for the purpose of reasoning on the videos. The reasoning process is implemented by using an OWL reasoner.

1.2 Impact and Contributions of the Thesis

The main impact of this thesis is to apply the semantic web technologies to multimedia retrieval tasks. In this respect, semantic web standards will be useful for the purpose of semantic querying and reasoning on video contents. Starting from this point, the contributions of the thesis can be listed as follows:

- The infrastructure proposed in this thesis uses a video data model based on annotation layering. But, an object-based approach is also used in the annotations in order to hybridize this data model with the object-based models for structuring the video annotations. Eventually, an improvement for the video data model introduced in [25] is achieved in the context of annotation layering.

- Concerning the needs about semantics in the multimedia domain, semantic web technologies match with multimedia content retrieval tasks by the help of ontology concept. Domain ontologies are used to increase the strength of the semantics in the retrieval tasks as the ontological systems have more powerful reasoning abilities compared to the common keyword-based systems.

- The infrastructure lets the users get answers of the queries which can not be answered by the keyword-based systems without the domain knowledge.

- The annotation tool which is developed in the scope of this thesis uses domain ontologies in order to suit the hybrid video data model.

- For the purpose of creating an infrastructure instead of only implementing a projected system, the system is designed in such a way that the components are independent of each other and the outputs of the components are based on worldwide standards such as MPEG-7 and OWL. This lets the developers study on the components more independently and makes the system standardized.

1.3 Organization

Chapter 2 introduces the related work on the subject of the thesis. Chapter 3 explains Semantic Web, its standards, and languages. Chapter 4 gives information about MPEG-7 and its standards for semantic content definition. Chapter 5 contains the design and implementation of the infrastructure proposed in this thesis. Chapter 6 makes a conclusion with the observations and future work.

CHAPTER 2

RELATED WORK

2.1 State of the Art

In [8], a knowledge representation infrastructure for semantic multimedia content analysis and reasoning is presented. This is one of the major objectives of the aceMedia Integrated Project where ontologies are being extended and enriched to include low-level audiovisual features, descriptors and behavioral models in order to support automatic content annotation. More specifically, the developed infrastructure consists of the core ontology based on extensions of the DOLCE core ontology and the multimedia specific infrastructure components. These are, the Visual Descriptors Ontology, which is based on an RDFS representation of the MPEG-7 Visual Descriptors and the Multimedia Structure Ontology, based on the MPEG-7 MDS. Furthermore, the developed Visual Descriptor Extraction tool is presented, which supports the initialization of domain ontologies with multimedia features.

The research in [13] is based on the studies about Semantic Web technologies, as advanced within the Knowledge Web project, in order to cope with the problems about multimedia metadata on the web. The requirements concerning enhancing the Semantic Web with multimedia metadata are investigated both from the analysis and the annotation perspective and the multimedia-related solutions achieved within the Knowledge Web consortium. These requirements can be listed as follows:

- A formal representation of the structure of a multimedia document depending on its actual type (image, video, audio, etc.)

- Representation of low-level multimodal descriptions in MPEG-7
- Spatio-temporal relations representation
- Fuzzy representation and reasoning
- A formal alignment framework for multimedia ontologies
- Support for multimedia-related data types, like numeric types (integer, float etc.), dates, vectors, arrays, etc.

The research in [4] presents algorithms and techniques that employ enriched ontologies for video annotation and retrieval. It implements a solution for the videos in the soccer domain. An unsupervised clustering method is proposed in order to create pictorially enriched ontologies by defining visual prototypes that represent specific patterns of highlights and adding them as visual concepts to the ontology. Two algorithms that use pictorially enriched ontologies to perform automatic soccer video annotation are proposed and results for typical highlights are presented. Annotation is performed associating occurrences of events, or entities, to higher level concepts by checking their similarity to visual concepts that are hierarchically linked to higher level semantics, using a dynamic programming approach. Usage of reasoning on the ontology is shown, to perform higher level annotation of the clips using the domain knowledge and to create complex queries that comprise visual prototypes of actions, their temporal evolution and relations.

In [24] the use of knowledge for the automatic extraction of semantic metadata from multimedia content is discussed. General purpose ontologies are enriched to represent the low level features of the multimedia for the representation of knowledge. A tool is implemented in order to map the MPEG-7 descriptors to the domain specific concepts.

It is introduced a methodology for extending the audiovisual content description standards MPEG-7 and Tv-Anytime with domain-specific knowledge descriptions expressed in OWL [5]. It also presents an interoperability mechanism between OWL and the audiovisual content description standards, which allows MPEG-7 and TV-Anytime descriptions and their domain-specific extensions to be described in OWL.

In [3] a methodology that allows the coupling of OWL with MPEG-7 and TV-Anytime is described. First, domain-specific knowledge is transparently integrated into MPEG-7 and TV-Anytime. Secondly, multimedia content and domain-specific information described in OWL is mapped into MPEG-7 and TV-Anytime. This methodology greatly facilitates information integration, retrieval and interoperability in Web application environments.

The research in [19] deals with the problem of finding multimedia data that fulfill the requirements of user queries. It is assumed that both the user query and the multimedia data are expressed by MPEG-7 standards. It is stated that MPEG-7 formalism lacks the semantics and reasoning support in many ways. A framework for querying multimedia data based on a tree embedding approximation algorithm, combining the MPEG-7 standard and an ontology is proposed.

In [14], it is proposed that multimedia and the semantic web are a perfect match. While the Semantic Web is providing a stack of languages and technologies for annotating Web resources, enabling machine processing of metadata describing semantics of web content, multimedia applications require metadata descriptions of their media items to facilitate search and retrieval, intelligent processing and effective presentation of multimedia information. This need for multimedia metadata was recognized by the media industry long ago. Semantic Web technologies, however, still play a very minor role within multimedia applications and most approaches employ non-RDF based techniques. This research describes a number of current approaches to multimedia metadata and provides an inventory of the open issues to achieve a practical integration of multimedia metadata into the Semantic Web.

In [16] a new tool called OntoELAN is generated. It is based on ELAN. ELAN is a linguistic multimedia annotation tool that allows users to create, edit, visualize and search annotations of video and audio data. Since the annotations in ELAN are represented by texts, and the meanings of the text are not Recognized by machines, it is impossible to apply further linguistic searches and comparisons on the Web. As the development of GOLD (General Ontology for Linguistic Description), the first linguistic ontology by the linguistic group in University of Arizona, ELAN is extended to support linguistic ontological annotations. In addition to all the functions of ELAN, OntoELAN supports the following actions:

- Creating and editing the profile that links ontological terms to a linguistic ontology and user defined terms
- Creating the ontological tier, on which user can create ontological annotations
- Annotating an ontological tier using the terms from a profile (user defined terms and corresponding linguistic ontology terms)
- Creating the instances of the related linguistic ontology class; generating an annotation file, which is written by OWL

In [2] an XML-based video database system is developed based on the data model [21] which is focused on semantic content of video streams. This system is aimed to be compliant with the MPEG-7 Multimedia Description Schemes in order to obey a universal standard. The system is implemented using a native XML database management system. Query entrance facilities are enhanced via integrating a natural language interface enabling flexible querying developed in the context of the study in [15]. In this study, WordNet domain independent ontology is used for improving semantic querying.

The study in [21] is based on a video data model that supports the spatio-temporal querying of video contents. Both semantic-based features such as objects, events and content based features such as the spatial properties of the objects are the main issues in this model. In addition to this model design, a prototype of the study as an annotation tool is implemented.

2.2 Tools

2.2.1 KAON2

KAON2 [7] is an infrastructure for managing OWL-DL, SWRL, and F-Logic ontologies. It is developed by Information Process Engineering at the Research Center for Information Technologies, Institute of Applied Informatics and Formal Description Methods at the University of Karlsruhe, and Information Management Group at the University of Manchester.

KAON2 is the progression project of the KAON1 project [7]. However, it is impossible to completely evaluate KAON2 as the improved version of KAON1. KAON2 is a new system and there is an important difference concerning the ontology languages supported by KAON1 and KAON2. While KAON1 is using an extension of RDFS, KAON2 uses OWL-DL and F-Logic. Therefore, KAON2 is not compatible with KAON1.

The main features of KAON2 are as follows [7]:

- Offering an API for the management of OWL-DL, SWRL, and F-Logic ontologies
- Enabling the distributed access to the ontologies by implementing an RMI based server
- Providing an inference engine for answering SPARQL queries
- Providing an interface enabling access from tools like protege
- Having the capability of extracting ontology instances from relational databases

The syntax and semantics accepted by the KAON2 is standardized. The syntax is composed of OWL-XML and OWL-RDF separately. On the other hand, syntactically, it supports OWL-DL, F-Logic and SWRL ontologies as it is mentioned before.

KAON2 accepts the SHIQ(D) subset of OWL-DL (except nominals), function-free subset of F-Logic, and the DL-subset of the Semantic Web Rule Language (SWRL) ontologies for reasoning [7]. The DL-subset is chosen for the SWRL in order to enforce the reasoning decidable.

The queries which KAON2 accepts to answer should be in the form of SPARQL. However, all of the specifications of SPARQL are not supported. The queries which can be mapped to conjunctive forms are the ones accepted by KAON2. But, this specification is not enough as KAON2 can also not support second order predicate logic in order to extract the answers. Therefore, KAON2 does not accept the queries including predicate variables.

Although most of the DL reasoners, such as FaCT and Pallet, use tableaux calculus, KAON2 use algorithms reducing a SHIQ(D) knowledge base to a disjunctive datalog program [7]. As a result of using these algorithms, KAON2 answers the queries faster than these systems according to the performance evaluation of KAON2 [1].

CHAPTER 3

THE SEMANTIC WEB AND ONTOLOGIES

3.1 Fundamentals

Semantic web emerged as the "web of data that can be processed directly and indirectly by machines" which is the definition of Tim Berners Lee who is the inventor of World Wide Web. Actually, semantic web is not considered as a totally isolated web alternative. It can be thought as a supplementary technology to the current web. On one hand, current web logic based on HTML (HyperText Markup Language) semi-structured standards carry out the presentation tasks for human sense. On the other hand, semantic web intends to provide the machine understandability with a fully structured approach.

The current web can be defined as the second generation web. While the first generation web was identified by handwritten HTML pages, the second generation has been improved to machine generated HTML pages. These generations were developed for direct human processing [23]. In the third generation Web, the main approach is to make web machine understandable by creating annotations describing the content of web resources.

As it is mentioned above, HTML presents a semi-structured standard. It does not provide the mechanism that helps us reach the machine understandability. In order to achieve that goal, XML (Extensible Markup Language) is an appropriate option constructing the structure of the information according to the mentioned purpose. But the interoperability problem occurs at this point. As the XML markups of different information sources that points the same entities may be different, XML is not enough for ensuring the interoperability of different information sources. Thus, different agents may comment differently about the same information. In other words, a web page structure that is understandable by an agent may have no meaning for another agent in that way. Moreover, XML is also not capable of relating the objects defined by using its tags. So RDF (Resource Description Framework) presents another standard for the relations and interoperability needs. Despite this important capability, it is restricted to simple relations regarding complicated tasks. In this aspect, we need more complicated standards that help us conduct reasoning and inference mechanisms. Therefore, the markup languages like SHOE, DAML-ONT, OIL, DAML+OIL, OWL etc. are created to satisfy these needs.

3.2 Layers of Semantic Web

The semantic web principles are implemented in the layers of web technologies and standards. (Tim Berners Lee, layer cake) The main reason why this approach is preferred is that agreeing on small portions is much easier than agreeing on the big picture. Moreover, the needs for standardization of common concepts can not be discarded. From this perspective, it is important to fix a settlement to create new standards.

In order to achieve the goals of the layered infrastructure, standards developed in a specified level must have the ability of using and giving meaning to the information produced at lower levels. Likewise, these standards should be able to use some worthwhile outcomes of the higher levels. The base layer of the semantic web is occupied by XML. The other models, such as RDF and RDF schema, using XML syntax, are constructed on top of XML. Concerning these facts, two principles should be followed in building one layer of the semantic web on top of another [9]:

- *Downward compatibility:* The information represented according to the standards of lower levels should be understood by the agents designed according to the standards of higher levels.

- *Upward partial understanding:* On the other hand, the agents designed according to the standards of lower levels should be able to partially understand the information represented according to the standards of higher levels.

In Figure 3.1, Tim Berners Lee's "layer cake" is shown. The layers of the semantic web exist in the "layer cake" [20].

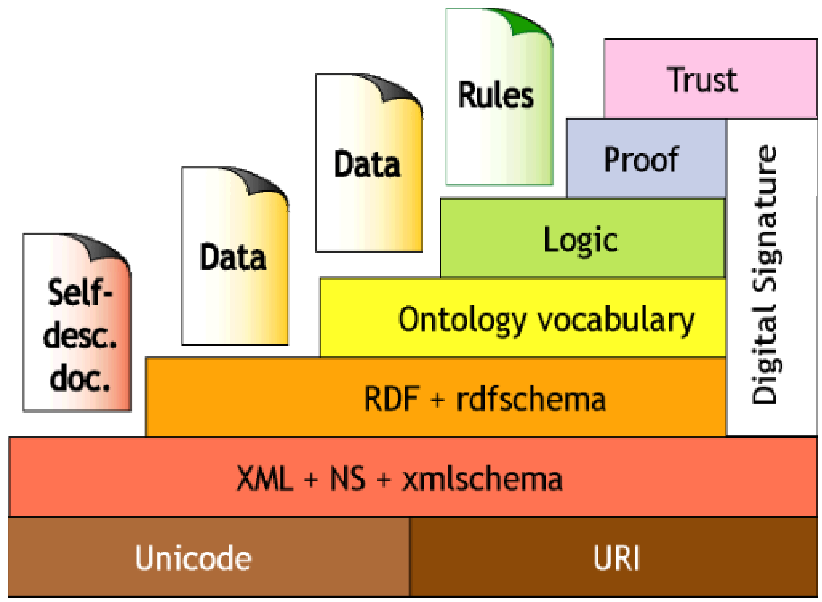

Figure 3.1: Layers of Semantic Web

In the base layer, we can see XML, which allows structuring the data with a predefined vocabulary. XML is an important standard for the communication of documents on the web.

RDF is a standard for the definition of concepts and relations and it is on top of the XML layer. RDF Schema is used for modeling web objects using hierarchies. Classes, properties, subclass and subproperty relations, and domain and range restrictions are some of the key concepts of RDF Schema. RDF Schema is based on RDF.

RDF Schema is a primitive language for creating ontologies. However, it is not enough for representing the complex relations and more powerful ontology languages expanding RDF Schema are needed. Thus, the logic layer is used to enrich the ontology languages, derive the languages having more semantic abilities and allow producing declarative knowledge [9].

3.3 Ontology Concept

In general, ontology is a concept that takes its roots from philosophy. It concerns with the nature of existence as a branch of metaphysics. Nonetheless, the concept is also used in artificial intelligence and web surveys about semantic web in computer science. The classical AI definition (Gruber's definition) of ontology is "an ontology is an explicit and formal specification of conceptualization". In its most common uses in computer science, ontology refers to an engineering structure constructed by a predefined vocabulary to describe an existence. In addition to the vocabulary, a set of logical assumptions are defined in order to give the meaning to the vocabulary for machine understandability and a first order logic theory is used to represent these assumptions. From this aspect, vocabulary can be represented as a set of predicates defining concepts and relations [17].

Occupying some of the abstract layers of semantic web, ontologies are strongly used by semantic web for the purpose of comprehensive and transportable machine understandability [17]. It is mainly related with the definition of metadata. Metadata annotations are very important for the agents to reach the resources effectively. The effect of ontology appears here as a construction that consists of the descriptions of the concepts and their properties in the specified domain. Consequently, ontologies' support occurs with the presentation of these defined concepts to the use of meta-data.

3.4 Basics of Ontology Languages

An ontology language is a means to represent a constructed ontology at an abstract level. More precisely, an ontology language is a formalism which is used for holding the constraints of the domain as well as its conceptualization. That means not only the conceptualization but also the domain specifications of an existence are carried out by the ontology language.

The word “Domain” has an important role at that point. The scope and concept structure of the domain determines the definition of ontology in order to describe the domain and the representation implemented by the ontology language. Let us consider a domain description with an example of health system:

Assume that the scope of the health system domain is restricted with a hospital of faculty of medicine. The first thing we have to do is to specify the distinction of the things we are going to talk about. In order to do that, we should talk about two concepts – classes, objects. Objects are the instances of classes. Classes and objects are known as resources and concepts in an ontology respectively. In our health system, we may talk about particular patients such as Bill Jordan, particular departments such as cardiology and particular doctors such as Adam Price. We can also talk about the professors, specialists, departments etc. This shows the difference of the objects and the classes. More specifically, Bill Jordan is a kind of object, on the contrary professors, departments are the classes.

An important use of classes is to impose restrictions on what can be stated. It can be thought like programming languages. For instance [6], if you want to write an expression such as m + 1 = m in C, the compiler will warn you about the inconsistency of the lvalue of the expression. In the context of ontologies, we may want to block the statements such as “Bill Jordan is treated by cardiology.” and “Pathology is treated by Adam Price.”. In the first statement, range of the property - “is treated by” - is restricted as the target set can only include a doctor. On the other hand, domain of the property - “is treated by” - is restricted in the second statement as the source set can only include the patients.

Class Hierarchies: As we have classes, we should express the relations between them [6]. In the health system example, suppose that the classes are; staff, doctors, professors, associate professors, assistant professors, specialists, assistants, administrative staff, dean, heads of departments, patients, departments

These classes are strongly related with each other. For example, staff – doctors – professors are the classes which have the superclass-subclass relation. While "doctors" class is the subclass of the "staff" class, it is the superclass of the "professors" class. The subclass relationship is also known as subsumption.

Subsumption determines the hierarchy of classes. Let us define a rule for subsumption, suppose I(A) represents the instance set of class A and S(A, B) is a predicate that returns true if "class A is a subclass of class B":

$$\forall A\ \forall B\ [S(A, B) \Leftrightarrow [\forall x\ (x \in I(A) \Rightarrow x \in I(B))]]$$

The logical expression above is the formal representation of subsumption. From an informal point of view, "A is a subclass of B if and only if every instance of A is also an instance of B" describes that relation.

Hierarchies are also very significant for the restrictions of our domain which we have mentioned before. Let us remember the example given to show the impossible statement of "is treated by" property in our domain caused by the domain set restrictions of the property:

S1 : "Pathology is treated by Adam Price."

We have the domain restriction of:

S2 : "Only a patient is treated by doctors."

Suppose that Adam Price is a specialist and we do not have the information that Adam Price is also a doctor. Therefore, we are not able to say that S1 is not possible. Pathology is not a patient but S1 may be true as we do not have the information that a professor is also a doctor in our domain. We can easily see that adding the statement "A specialist is also a doctor." to our description will not be a good idea to overcome the problem. The appropriate solution is inheriting the treating ability to the specialists class from the doctors class [6].

Property Hierarchies: We saw that hierarchical relationships between classes can be defined. The same can be done for properties [6]. For example, “is treated by” is a subproperty of “meets”. If a patient P is treated by a doctor D, then P also meets D. The converse is not necessarily true. For example, P may meet D but this does not mean that P is treated by D. This subproperty relation may be described by the following logical expression:

(Assume that S(A, B) is a predicate that returns true if A is a subproperty of B and

R(x, y, Q) is a predicate that returns true if x and y objects are related by property Q.)

$$\forall A\ \forall B\ [S(A, B) \Leftrightarrow [\forall x\ \forall y\ (R(x, y, A) \Rightarrow R(x, y, B))]]$$

From an informal point of view, “A is a subproperty of B if and only if two objects are related by B whenever they are related by A.” describes that relation.

Consequently, ontology languages consist of [6]:

- Concepts
- Relationships between the concepts
- Relationships between the relationships
- Implicit relationships in the ontology language
- Constraints (domain and range restrictions, etc.)

Ontology languages allow users to write explicit, formal conceptualizations of domains models. The main requirements are syntax, semantics and reasoning.

3.4.1 Syntax

The clear definition of the syntax of any language (natural, programming, ontology, etc.) is very important regarding the human or machine understandability. The importance is focused on machine understandability in the case of ontology languages.

The syntax of ontology languages is generally based on XML [6]. As the XML based syntax is inappropriate for writing ontologies manually, ontology management tools are produced. This reminds us the difference between the first and second generation web discussed in 3.1. The mapping pattern of the web design (HTML generator) tools can be thought as the ontology management tools in this perspective.

3.4.2 Semantics

Formal semantics specifies the meaning of knowledge [6]. The semantics of ontology languages expresses this specification by using domain constraints.

Given a statement in an ontology, the role of the semantics is to devise precisely which are the models of the statement, i.e., all the possible instantiations of the domain that are compatible with the statement. We say that a statement is true in an instantiation of the domain if this instantiation is compatible with the statement; the instantiation of the domain in which a statement is true is of course a model of the statement. So, an ontology will itself derive a set of models, which is the intersection among all the models of each statement in the ontology. The models of an ontology represents the only possible reliable situations.

3.4.3 Reasoning

The fact that the formal semantics associates to an ontology a set of models, allows us to define the notion of entailment. Given an ontology, we say that an additional statement can be entailed by the ontology if it is true in all the models of the ontology. This definition of entailment comes from logic. If a deduction produced by a proof technique produces a statement which is entailed by the ontology, this deduction is called a valid deduction. The process of deriving valid deductions from an ontology is called reasoning. If we consider the typical statements of ontology languages, the following deductions can be introduced [6]:

Class membership: Deduction of being an instance of a class. Example: Adam Price is an instance of class specialists. Specialists class is a subclass of doctors class in the ontology. So, it can be inferred that Adam Price is an instance of doctors class as the resulting statement is true in all the models of the ontology.

Classification: Deduction of the subclass relationships between the existing classes in the ontology. Example: Specialists class is a subclass of doctors class and doctors class is a subclass of staff class in the ontology. Therefore, it can be inferred that specialists class is a subclass of staff class as the model set of specialists class is a subset of the model set of staff class in the same manner with the subclass hierarchy.

Equivalence of classes: Deduction of the equivalence of the classes. Example: Specialists class is equivalent to doctors class and doctors class is equivalent to staff class in the ontology. Therefore, specialists class is equivalent to staff class.

Consistency of the ontology: Deduction of the consistency of the ontology. There is at least a possibility to have an instantiation of the domain compatible with the ontology. Example: Suppose we have described in the ontology:

- David is an instance of both doctors and patients classes.
- Doctors and patients are two disjoint classes.

Then we have an inconsistency because the two constraints can not be satisfied simultaneously. Statement 2 says that the extensions of the two classes can not have any element in common, since they are disjoint, but statement 1 says that David is an instance of both classes. This clearly indicates that there is an error in the ontology, since it does not represent any possible situation.

3.5 Web Ontology Languages

The recognition of the key role that ontologies are likely to play in the future of the web has led to the extension of web markup languages in order to facilitate content description and the development of web based ontologies, e.g., XML Schema, RDF, RDF Schema etc. RDF Schema (RDFS) in particular is recognizable as an ontology representation language: it talks about classes and properties (binary relations), range and domain constraints (on properties), and subclass and subproperty (subsumption) relations. RDFS is, however, a very primitive language (the above is an almost complete description of its functionality), and more expressive power would clearly be necessary in order to describe resources in sufficient detail. Moreover, such descriptions should be responsible for automated reasoning if they are to be used effectively by automated processes, e.g., to determine the semantic relationship between syntactically different terms. Therefore, for a more expressive power, web ontology languages, e.g., SHOE, DAML-ONT, OIL, DAML+OIL, OWL, SWRL are used for ontology construction. The properties of OWL and SWRL are explained below [6, 22]:

3.5.1 OWL

OWL is the name given by the W3CWeb Ontology (WebOnt) Working Group to the ontology language standard they are developing and which is based on DAML+OIL [6]. The main features of the language are very similar to those of DAML+OIL. OWL supports three sublanguages: OWL Lite, OWL DL and OWL Full [6].

3.5.1.1 OWL Lite

This sublanguage includes the features that enable the hierarchy of the classes and the restricted constraint types over the domain. Cardinality constraints and subsumption relation between the classes are the examples of the features which are supported by OWL Lite.

3.5.1.2 OWL DL

This sublanguage is named according to its relation with description logics which is a decidable type of first order logic. It supports description logics semantics and reasoning. OWL DL is complete, decidable and more expressive than OWL Lite. Moreover, it includes all OWL language features and restrictions.

3.5.1.3 OWL Full

This sublanguage is designed for the needs of more expressiveness. As a result of obtaining a high level of expressiveness, the semantics and reasoning features do not meet the logical requirements such as soundness, completeness or decidability. Therefore, it will be difficult to find any reasoning engine that supports all OWL Full semantics.

3.5.2 SWRL

In this section, Semantic Web Rule Language (SWRL) based on a combination of the OWL DL and OWL Lite sublanguages of the OWL Web Ontology Language with the Unary/Binary RuleML sublanguages of the Rule Markup Language is briefly explained. The set of OWL axioms are reorganized in order to possess Horn-like rules and it enables Horn-like rules to be combined with an OWL knowledge base [22].

The rules are in the form of the implications between an antecedent (body) and consequent (head). If the conditions in the antecedent hold, the conditions in the consequent must also hold.

Atoms in these rules can be of the form C(x), P(x,y), sameAs(x,y) or differentFrom(x,y), where C is an OWL description, P is an OWL property, and x,y are either variables or OWL individuals. This explanation is shown in Table 3.1 [22]. It is also easy to see that OWL DL becomes undecidable when extended in this way as rules can be used to simulate role value maps. An example of SWRL ontology is in Figure 3.2.

Table 3.1: SWRL Atom Types

	OWL description	OWL property	Variable	OWL individual
C(x)	X			
P(x, y)		X		
x, y			X	X

```
<swrlx:Ontology swrlx:name="football_ontology_v02"
     xml:base="http://kaon2.semanticweb.org/football_ontology_owl"
     xmlns:owlx="http://www.w3.org/2003/05/owl-xml#"
     xmlns:ruleml="http://www.w3.org/2003/11/ruleml#"
     xmlns:swrlx="http://www.w3.org/2003/11/swrlx#">

<owlx:Class owlx:name="#Goalkeeper" owlx:complete="false">
     <owlx:Class owlx:name="#Player"/>
</owlx:Class>

<owlx:Class owlx:name="#Team" owlx:complete="false">
     <owlx:Class owlx:name="#Group"/>
</owlx:Class>

<owlx:Class owlx:name="#Region" owlx:complete="false">
     <owlx:Class owlx:name="#Object"/>
</owlx:Class>

<owlx:ObjectProperty owlx:name="#miss">
     <owlx:domain owlx:class="#Player"/>
     <owlx:range owlx:class="#Team"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#shoot">
     <owlx:domain owlx:class="#Player"/>
     <owlx:range owlx:class="#Region"/>
</owlx:ObjectProperty>

<owlx:Individual owlx:name="#goal">
     <owlx:type owlx:name="#Region"/>
</owlx:Individual>

<ruleml:imp>
     <ruleml:_head>
          <swrlx:individualPropertyAtom swrlx:property="#miss">
               <ruleml:var>X</ruleml:var>
               <ruleml:var>Z</ruleml:var>
          </swrlx:individualPropertyAtom>
     </ruleml:_head>
     <ruleml:_body>
          <swrlx:individualPropertyAtom swrlx:property="#shoot">
               <ruleml:var>X</ruleml:var>
               <owlx:Individual owlx:name="#goal"/>
          </swrlx:individualPropertyAtom>
</ruleml:imp>
</swrlx:Ontology>
```

Figure 3.2: SWRL Ontology Example

CHAPTER 4

MPEG-7 MULTIMEDIA CONTENT DESCRIPTION STANDARD

4.1 MPEG-7 Overview

MPEG-7 standard aims to provide an infrastructure for interoperable operations on multimedia contents. Indexing, filtering and querying are the operations which needs to be interoperable as the multimedia sources are quite heterogeneous. Predefined properties of multimedia contents are described using MPEG-7 standard. MPEG-7 descriptions take two possible forms [10]:

- a textual XML form suitable for editing, searching, and filtering
- a binary form suitable for storage, transmission, and streaming delivery

Basically, the standard specifies four types of normative elements [10]: Descriptors, Description Schemes (DSs), a Description Definition Language (DDL), and coding schemes.

MPEG-7 Descriptors are designed to be used for describing the low-level multimedia features such as color, location, time, and so on. As the manual annotation process is not expected to be complex, descriptors of low-level features are needed to be extracted automatically using techniques such as video processing and speech recognition in multimedia applications.

MPEG-7 DSs (Description Schemas) are designed to describe high-level multimedia features such as objects, events, segments, etc. The DSs produce more complex descriptions by integrating together multiple Descriptors and DSs, and by declaring relationships among the description components. In MPEG-7, the DSs are categorized as pertaining to the multimedia, audio, or visual domain. Typically, the MDSs describe content consisting of a combination of audio, visual data, and possibly textual data, whereas the audio or visual DSs refer specifically to features unique to the audio or visual domain, respectively. In some cases, automatic tools can be used for instantiating the DSs, but in many cases instantiating DSs requires human assisted extraction or authoring tools. [10]

DDL is a language for specifying the syntax of the Descriptors and DSs [10]. It is based on XML Schema Language. The DDL also allows the DSs and Descriptors to be extended for specific applications. In this sense, MPEG-7 annotations of multimedia contents are obtained by producing DSs or Descriptors as the definitions allowed by the DDL.

After introducing the elements briefly, it is now clear that MPEG-7 descriptions can be created using these elements. An MPEG-7 Description includes a Description Scheme and Descriptor Values which initiate the DS and describe the content.

From another aspect, the MPEG-7 standard consists of several parts. This lets us use the various clusters of technology alone, according to MPEG's toolbox approach to standardization. It also keeps the editing of the standard manageable.

4.2 MPEG-7 Components

After giving a brief overview of the elements and parts of MPEG-7 standard, this part of the chapter provides a brief description of the MPEG-7 components.

4.2.1 MPEG-7 Description Tools

MPEG-7 description tools The Descriptors and Description Schemes together form the set of MPEG-7's predefined description tools. We can group them in different classes according to their functionality [12]; Basic Elements, Schema Tools, Content

Description Tools, Audio Description Tools, Content Management Tools, Content Organization Tools, Navigation and Access Tools and User Interaction Tools.

MPEG-7 description tools are a combination of Descriptors and Description Schemes [12]. In each application, a different subset of these tools is needed. In this aspect, determining the subsets for the applications is still a problem for the developers. Previous MPEG standards provide us experiences in this problem. These experiences show us that it is appropriate to use normative subsets until a real industry implementation is carried out [12]. Despite these facts, different description tools that are not in the standard might be needed in some applications. In such cases, using the DDL is the solution.

4.2.2 MPEG-7 Data Description Language

As it is briefly expressed in the overview, the DDL lets us describe the MPEG-7 description tools (Descriptors and Description Schemes) and extend them with application specific description tools.

During the early phases of MPEG-7, the DDL had been developed as a markup language in the context of MPEG. This standard continued until it is decided to be redeveloped on top of XML Schema. After that occasion, the DDL became more flexible. MPEG-7 concept such as matrices and vectors are supported by adding extensions to XML Schema. As XML Schema is a common representation formalism, the development of the DDL on XML Schema also increased the interoperability of the systems using MPEG-7 as the description standard.

Consequently, MPEG-7 DDL is an important background for MPEG-7 standard and it is the determining factor for the flexibility and interoperability of the MPEG-7 based systems.

4.3 Why MPEG-7

As we have seen, MPEG-7 is an important multimedia standard which is composed of the tools and components in order to generate descriptions explaining the multimedia content. Regarding this feature based definition, MPEG-7 can be thought as a multimedia description schema.

In the infrastructure, MPEG-7 based metadata generation is a middle phase that is fed by raw video data and feeds the reasoning engine which uses OWL and SWRL. But some questions may come into mind at this point: "Why do we use MPEG-7 and why don't we directly generate an OWL based metadata as a result of the annotation process by discarding this phase?" The facts that will be the answers for the questions are listed as follows:

- It is important to keep the video annotations in such a model that this part of the infrastructure is standardized according to a video metadata standard. As MPEG-7 is a common standard for the video content description, it will be appropriate to use MPEG-7 as the video metadata standard in the infrastructure.

- As the projected structure is an infrastructure instead of a simple system, different systems such as MPEG-7 based video database systems should easily be integrated into the infrastructure. Therefore, the connection point should be the video metadata part which uses a worldwide standard in order to make the infrastructure more universal.

- Adding a keyword based multimedia content retriever as a future work for the infrastructure regarding its time efficiency compared to the semantic system will only be possible if a common video metadata standard is used. Thus, generating the video metadata according to the standards and tools of the multimedia content description components instead of generic semantic web standards will make the study more flexible.

CHAPTER 5

THE DESIGN AND IMPLEMENTATION OF THE INFRASTRUCTURE

5.1 Video Data Model

One of the important issues that affects the design of the infrastructure is the video data model used to model the video content. As it is mentioned in part 1.1, the video data model of this infrastructure is based on annotation layering model. However, an object-oriented approach is combined with the annotation-based approach in the lower levels of the data model. Therefore, a hybrid video data model using an object-oriented approach with a model based on annotation layering is obtained. As seen in Figure 5.1, the general structure uses annotation layering while an object-oriented methodology is used in the lower levels of the model.

Before describing the structure of the video data model in detail, it is important to explain the general picture. In this general picture, video contents are modeled as the raw video data and the video metadata separately in the infrastructure. Although the raw video data and the video metadata are physically kept as separate entities, they are mapped semantically. In other words, each physical data set forming the raw video data has at least one corresponding metadata unit that describes its content. In this context, the metadata units are occupying the highest level in the object oriented approach while they are the basic elements of annotation layering.

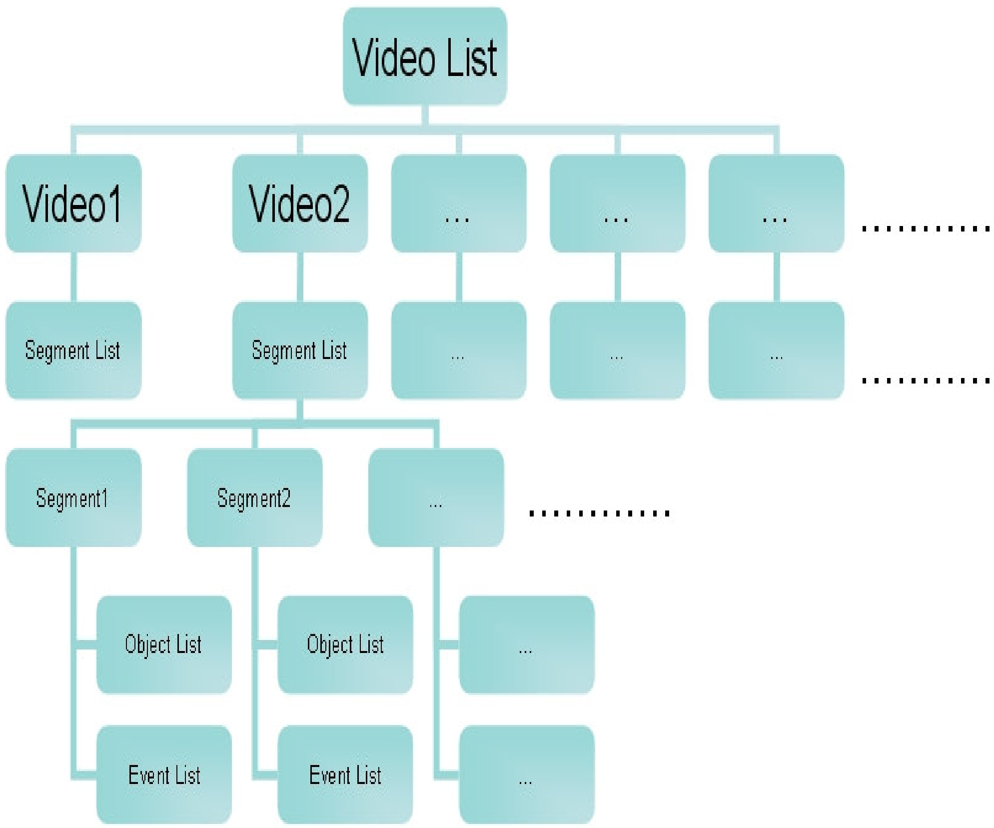

Figure 5.1: Video Data Model

The video data model of this infrastructure has a hierarchical structure as we can see in Figure 5.1. In this hierarchical structure, video contents are described according to different levels of abstractions. In this perspective, the individual models of the hybrid approach should intersect at some level of this hierarchy. As it is pointed out the metadata units are the elements of both annotation layering and the object-oriented models, the level of metadata units is the intersection level of these two models forming the hybrid model. Therefore, the level of video segments is the specified intersection level as the video segments are the metadata units describing the contents of video time intervals.

The levels of hierarchical structure of the video data model and their specifications are described as follows:

Video List: Video List is the highest abstraction level of the video data model. It holds the list of videos that are annotated according to this data model.

Video: Video is an abstraction level which forms the building blocks of the Video List level. Video level is also the highest abstraction level of the annotation layering approach. The list of video segments which appear at the lowest abstraction level of the annotation layering model is held at Video level. The mapping of raw video data with the metadata also takes place at this level.

Segment List: Segment List includes the segments which describe the contents of the video time intervals.

Segment: Segment is an abstraction level in which the contents of the video time intervals are held. This level represents the basic units of the annotation layering model while it is providing the highest abstraction for the object-oriented approach.

Object & Event Lists: Object List and Event List forms the abstraction level in which the smallest semantic units (object, event) are held.

Objects and Events describe the basic semantic-based features of the video segments at the lowest abstraction level of the object-oriented model.

According to the data model described above, the annotation process of the raw video data is implemented. In particular, the video metadata is also modeled using the data model. The annotation process produces a video metadata archive. The video metadata archive is a list of video metadata. The elements of the list are the metadata each of which describes the content of a video. In each of these metadata, the annotations are separated into video segments. In each segment, there exist objects and events. Therefore, the video metadata is in the form of a collection of semantic segment descriptions using objects and events which are the basic elements of object-oriented approach.

5.2 System Architecture and Components

The flow of the infrastructure and the general architecture are shown in Figure 5.2 and Figure 5.3 respectively. As we can see in the architecture, the infrastructure is composed of a domain ontology and four main components: Annotation Tool, MPEG-7 to SWRL/OWL Converter, Reasoner Interface and KAON2 Reasoner. Each component produces data for the next component in the flow. "Annotation Tool", "MPEG-7 to SWRL/OWL Converter" and "Reasoner Interface" are the components that are developed in the scope of this thesis while "KAON2 Reasoner" is provided by an API. Domain ontology is also designed using "KAON2 Reasoner" API.

The "Annotation Tool" produces MPEG-7 based metadata for the "MPEG-7 to SWRL/OWL Converter" which produces an OWL Based ontology instance for the "Reasoner Interface". As the key component for the querying actions is the "Reasoner Interface", it is strongly coupled with the "KAON2 Reasoner" component. The "Reasoner Interface" sends queries to the "KAON2 Reasoner" and gets the answers using OWL Based ontology instances produced by the converter component.

The selected domain for the implementation is the soccer domain which is modeled using soccer domain ontology.

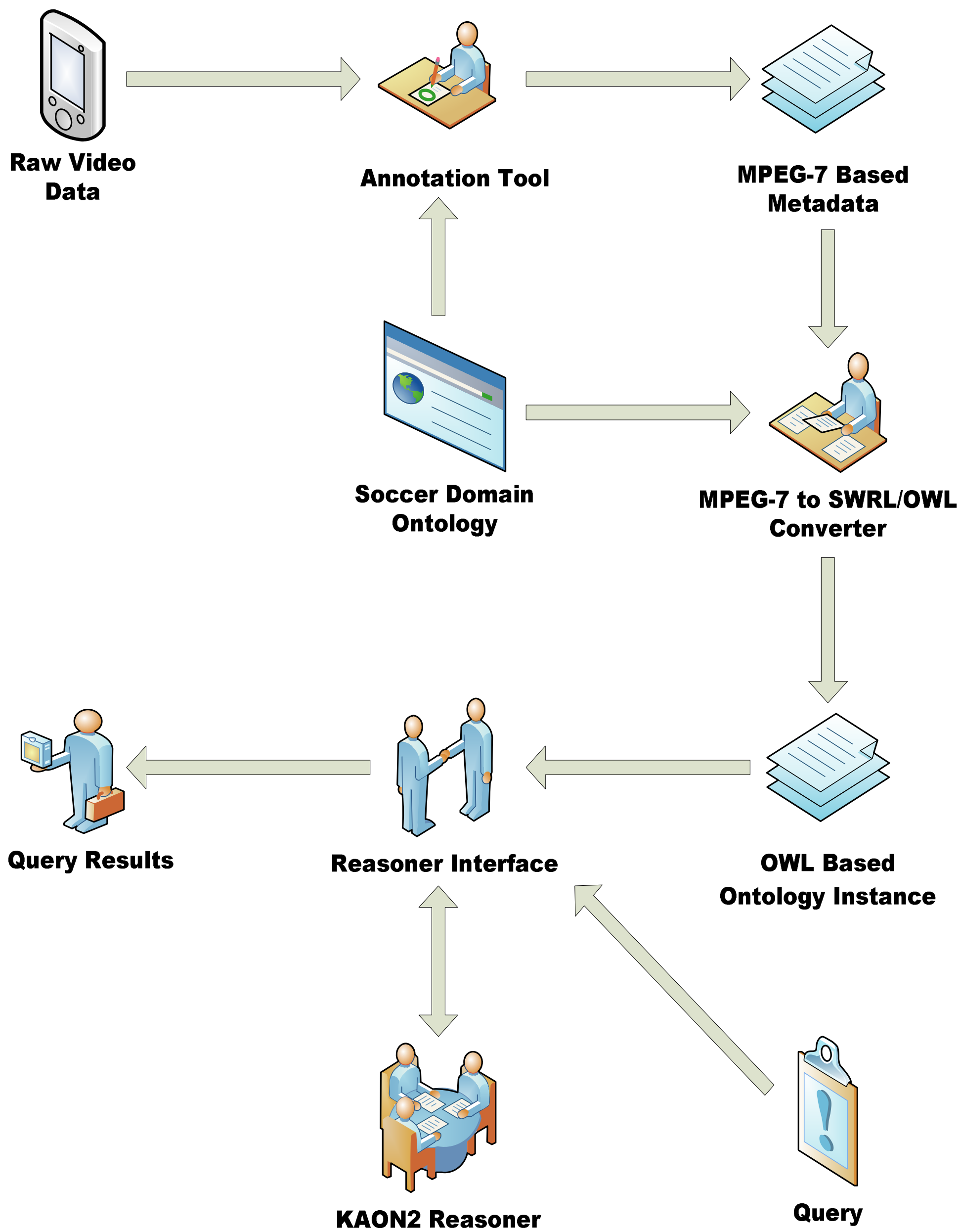

Figure 5.2: Flow of the Infrastructure

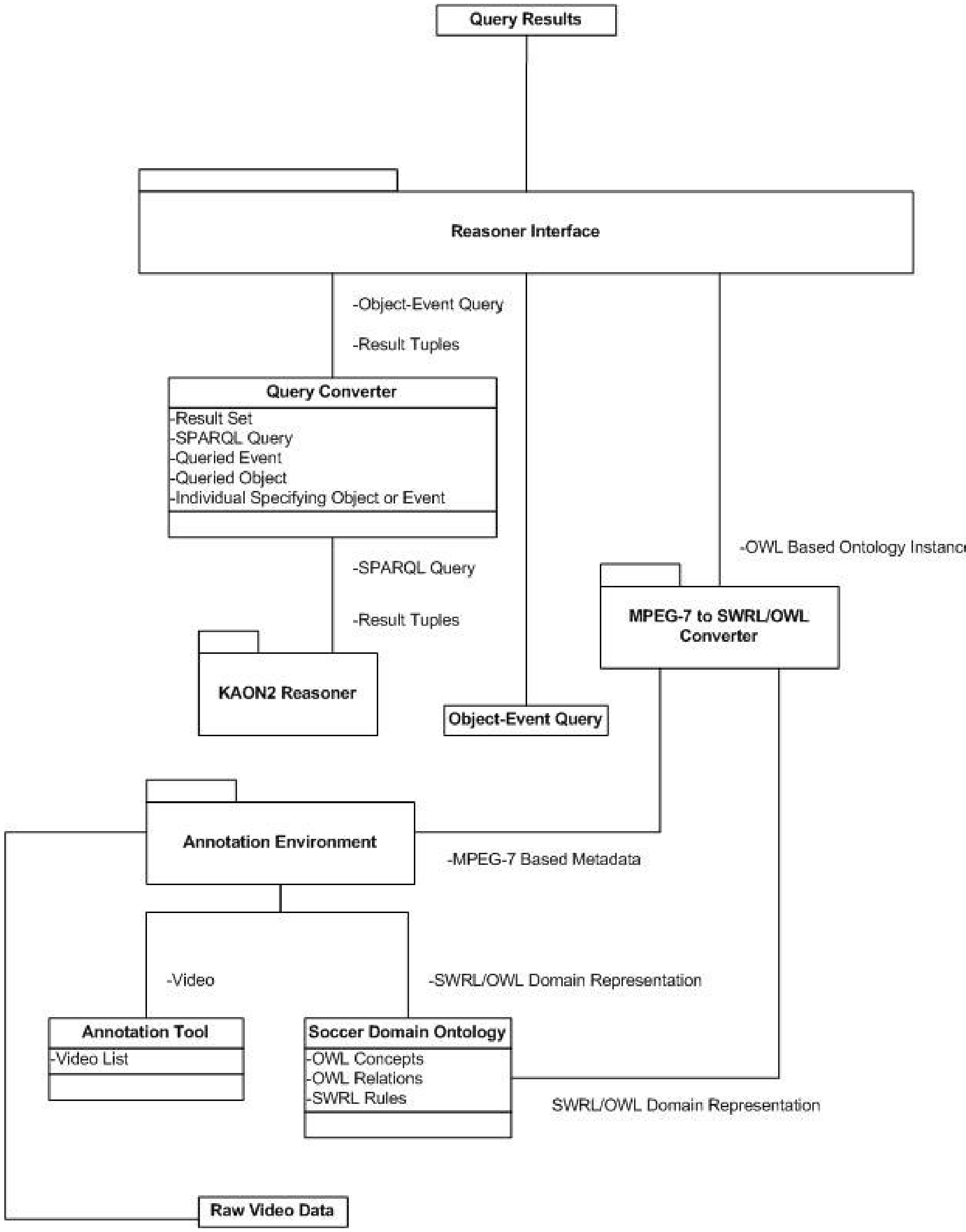

Figure 5.3: System Architecture

5.2.1 Components

In this part, the components constituting the infrastructure are described in detail regarding the ordering in the system flow. Namely, the specifications of the system architecture are elucidated in terms of separate modules.

5.2.1.1 Annotation Tool

Annotation Tool is one of the two basic components of the infrastructure. As the annotation of raw video data is the first step of the retrieval process, it is important to develop a stable tool which enables the users describe the contents of the videos semantically. In order to achieve this goal, an ontology based annotation tool is developed.

As it is mentioned before, MPEG-7 is the standard that will be used for modeling the metadata. That means that the annotation process will produce video metadata in MPEG-7 format. But it is impossible to say that all the videos having MPEG-7 based metadata possess the power of semantics which can be presented for the use of semantic systems. In order to obtain this power, the annotation process should be accompanied by an ontology. This will cause the metadata become meaningful regarding machine understandability. Therefore, the “Annotation Tool” forms an annotation environment together with the domain ontology.

In this infrastructure, domain ontology is used to strengthen the semantics. As the case study for the domain of the video is the soccer domain, the “Annotation Tool” is supported by the developed soccer domain ontology and forms an annotation environment together with the ontology as shown in Figure 5.4. From this perspective, the annotation of a soccer video can be accomplished by using the “Annotation Tool” which utilizes the concepts of the soccer domain and their restrictions described in the soccer domain ontology. As a result of this process, the “Annotation Tool” forms the MPEG-7 based metadata of the soccer video.

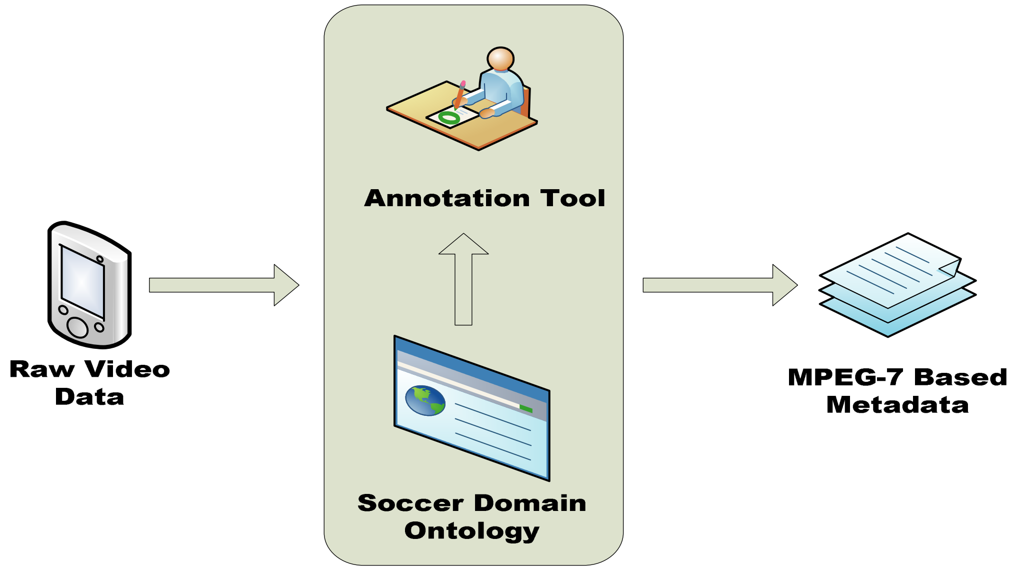

Figure 5.4: Annotation Environment

Some snapshots of the user interface designed for the Annotation Tool are shown in Figure 5.5, Figure 5.6 and Figure 5.7. As shown in these figures, objects and events, frame interval time, and the physical video path with the raw video data are given to the "Annotation Tool" by the user using the raw video data and soccer domain ontology.

The ontology support is also provided using an ontology button which lists the objects and events in the ontology in order to cooperate with the ontology for the purpose of annotating the object type, the event name and the types of the event parameters.

The structure of "Soccer Domain Ontology" and "MPEG-7 Based Metadata" are discussed in part 5.2.2 and 5.2.3.2. There will also be an example for understanding the structure of MPEG-7 Based Metadata in part 5.2.3.2.

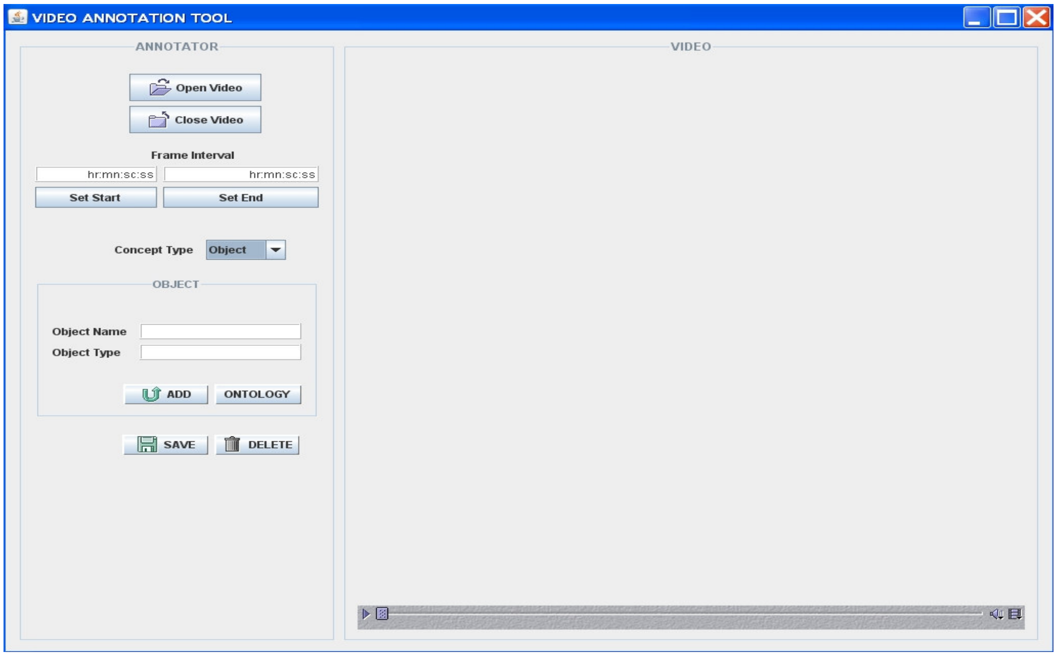

Figure 5.5: Annotation Tool Object Description Interface

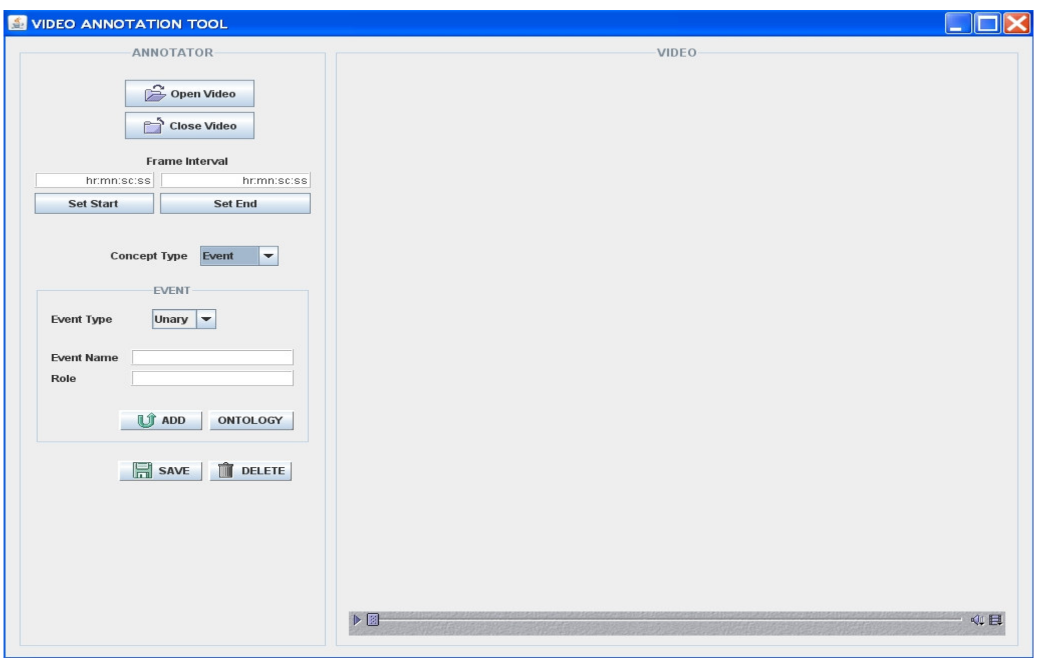

Figure 5.6: Annotation Tool Unary Event Description Interface

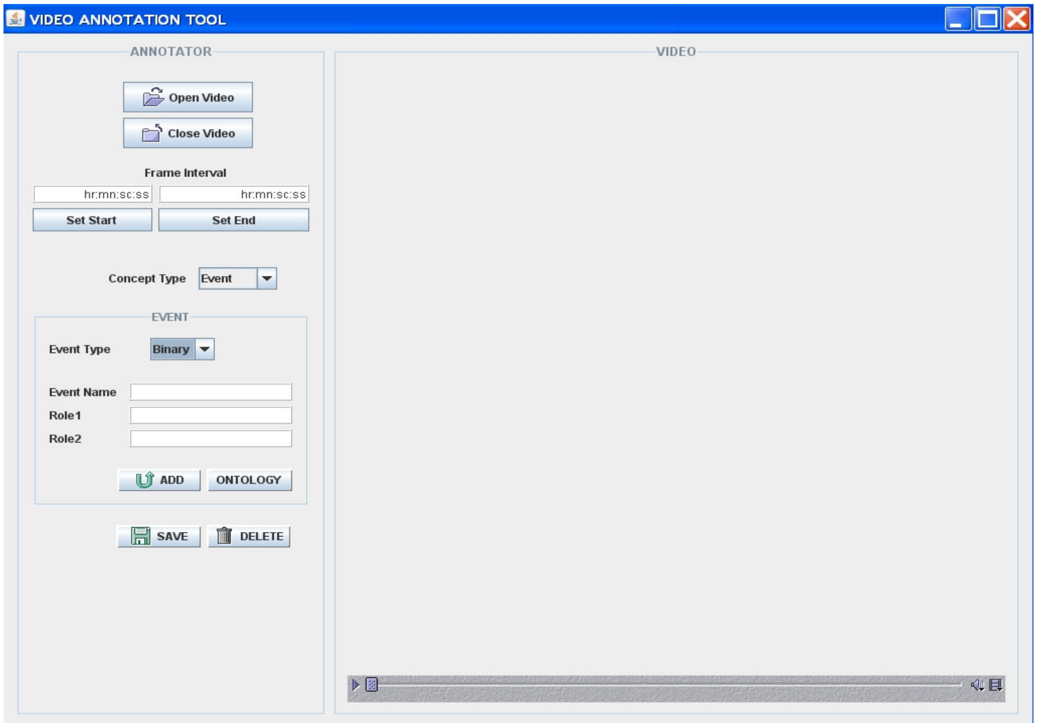

Figure 5.7: Annotation Tool Binary Event Description Interface

5.2.1.2 MPEG-7 to SWRL/OWL Converter

The "MPEG-7 to SWRL/OWL Converter" is the component that works on the "MPEG-7 Based Metadata" to create the OWL based ontological data which can be reasoned. It is developed in such a way that it can work with the annotation tool synchronously.

As the MPEG-7 metadata is produced by the ontological annotation tool developed, it has a semantic structure concerning the tool's features mentioned in 5.2.1.1. In order to utilize the semantics of the metadata, an ontology is needed. The metadata is meaningful along with the ontology used in the annotation process. Therefore, the "MPEG-7 to SWRL/OWL Converter" component uses the soccer domain ontology for producing the semantic content description. Eventually, this description will be an OWL based ontology instance.

While implementing the conversion process, this component first takes the MPEG-7 descriptions from the "MPEG-7 Based Metadata". As the MPEG-7 descriptions are in the form of objects and events defining the time intervals of the videos, the converter component creates OWL concepts and relations from the MPEG-7 object and event descriptions of a video time interval. In particular, the creation of OWL concept and relations is implemented according the one-to-one correspondence between the specified MPEG-7 descriptors and OWL entities as follows:

- (MPEG-7 object) – (OWL concept)
- (MPEG-7 event) – (OWL relation)

The specifications of the OWL concept and relation descriptions and the MPEG-7 object and event descriptions are explained in detail in parts 5.2.3.2 and 5.2.3.3.

After acquiring the OWL entities, the component combines them with the soccer domain ontology. Therefore, an ontology instance describing the selected time interval of the video in SWRL is created. Consequently, the video time interval is ready for reasoning.

The specifications about the ontology instances are particularly explained in part 5.2.3.3.

5.2.1.3 Reasoner Interface

The "Reasoner Interface" component is the part of the infrastructure where the users query the video contents. As it is mentioned in 5.2.1.2, SWRL ontology instances describing the video contents semantically are acquired for the purpose of reasoning. Then, the next step is to provide an interface which lets the communication of KAON2 Reasoner with the ontology instances according to the user queries.

The queries entered using the querying interface are in the form of object-event queries (Figure 5.8). They are firstly converted into SPARQL queries. Secondly, the ontology instances for the videos generated by the converter component are loaded using the API of KAON2. Then, the last phase is to get the answer of the SPARQL query.

In order to get the query results, query is executed using KAON2 Reasoner for each ontology instance. KAON2 Reasoner gives the query results after executing its reasoning mechanism according to the given query. Consequently, the videos whose ontology instances give positive results to the query are listed as the query answer and the answers are filtered according to the value restrictions in the query interface. The detailed specifications of the queries are explained in part 5.2.3.4. The relations between the object-event queries and the SPARQL queries can be understood from that part more clearly.

After developing this reasoner interface and providing integration with "KAON2 Reasoner" component, the most important aim of the thesis, semantic querying, is achieved. Consequently, the queries that can only be answered using the domain knowledge are answered using this component. An informal example is expressed as follows:

- Instantiated Domain Knowledge
    - Henry is an attacker
    - Zidane is a midfielder
    - Zidane assists Henry
    - Other information concerning the teams and other players
- Inferences:
    - Object Query
        - List the players :: Henry, Zidane (Reasoned using subsumption between the attacker, midfielder and player concepts)

- Event Query
  - List the players who score :: Henry (Reasoned using the rules between the assist and score)

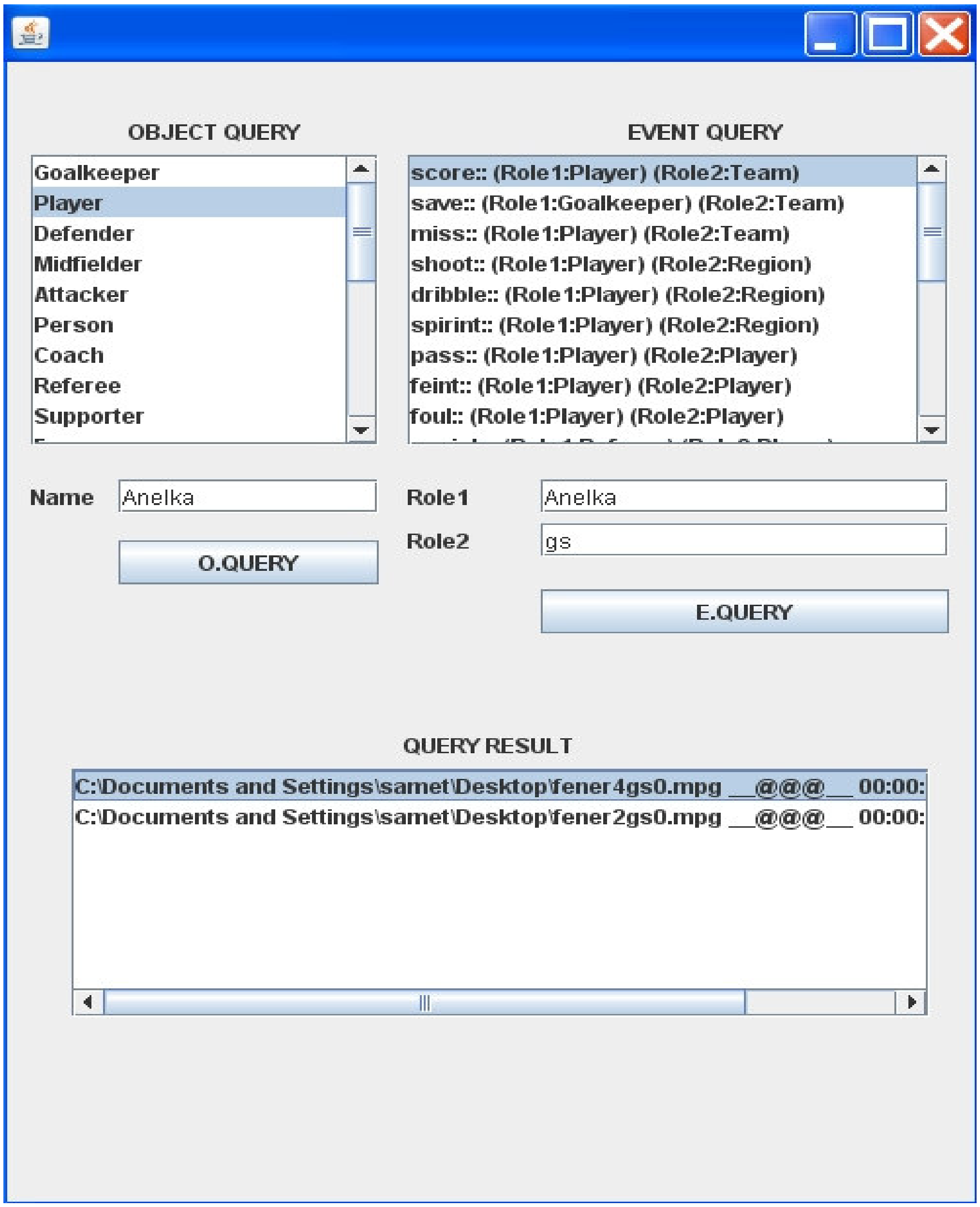

Figure 5.8: Querying Interface

5.2.1.4 KAON2 Reasoner

The KAON2 tool is used as the reasoner of the system. This component implements the reasoning task on the ontology instances. The features of this tool are described in 2.2.1 in detail. In the projected infrastructure, KAON2 communicates with the other parts using the reasoner interface. KAON2 does this by giving its API to the usage of reasoner interface component.

5.2.2 Soccer Domain Ontology

An important part of the infrastructure is the "Soccer Domain Ontology". This ontology describes the football domain formally. The ontology language is SWRL which includes OWL-DL descriptions and RULE-ML rule definitions. Therefore, SWRL is used to adapt these two standards.

The ontology was created by using the KAON2 ontology tool. After designing the ontology, a piece of code utilizing KAON2 API generates the ontology according to this design.

5.2.3 Input-Output Data Standards

After discussing the components and domain representation part of the infrastructure, the sort of data that provides the communication between the components is described in this section.

5.2.3.1 Raw Video Data

Raw video data is the core video data that is encoded according to a standard such as xvid, mpeg2, etc. The raw video data is used by the "Annotation Tool" component to create annotations by creating descriptions. The video is played by the annotation tool and especially time interval start-end points are taken from the raw video data. But, the raw video data is not used physically in the annotation process; the path of video is used instead of its physical volume.

5.2.3.2 MPEG-7 Based Metadata

"MPEG-7 Based Metadata" is one of two key data concepts in the infrastructure. As it is mentioned before, the video contents are needed to be represented semantically in order to implement the retrieval tasks. Concerning this need, MPEG-7 formalism is chosen for the annotation infrastructure.

"MPEG-7 Based Metadata" is produced by the Annotation Tool component. The annotations are generated according to the video data model described in part 5.1. Each metadata element in the metadata archive produced by the "Annotation Tool" is the representation of video abstraction level in the video data model hierarchy while the archive represents the video list level.

As it is mentioned before, the video data model has a hybrid structure and in this hybrid model, the video abstraction level includes the video segment list composed of the lowest abstraction level of the annotation layering model. Video abstraction level also includes the path of corresponding raw video data in order to connect the annotation with the raw video data. This connection is shown in Figure 5.9.

```
<MediaInstance>
    <InstanceIdentifier>Soccer game</InstanceIdentifier>
    <MediaLocator>
        <MediaUri>C:/fener4gs0.mpg</MediaUri>
    </MediaLocator>
</MediaInstance>
```

Figure 5.9: Description of the Video Mapping Raw Video Data

The lower level descriptions have the structure of concept definitions in the lowest abstraction level of the data model. Concepts are categorized into two groups as objects and events. While the objects are defined in one way with their properties (such as "Henry is an object who is an attacker"), the events are defined in two ways as unary and binary events. (such as "assist is an event of two objects" but “shoots is an event of one object). The hierarchy explaining concept types is shown in Figure 5.10.

The labels of these concepts are taken from the “Soccer Domain Ontology”. In the ontology the object and events are shown as the OWL concepts and relations with their hierarchical structure. The examples of OWL concept and relation descriptions are shown in Figure 5.11 and Figure 5.12. Figure 5.11 shows an OWL concept description meaning “Attacker is a concept which has a subclass relation with the Player concept meaning Attacker is a subclass of Player”. On the other hand, Figure 5.12 shows a binary OWL relation description meaning “score is a relation taking Player and Team concepts as the parameters”.

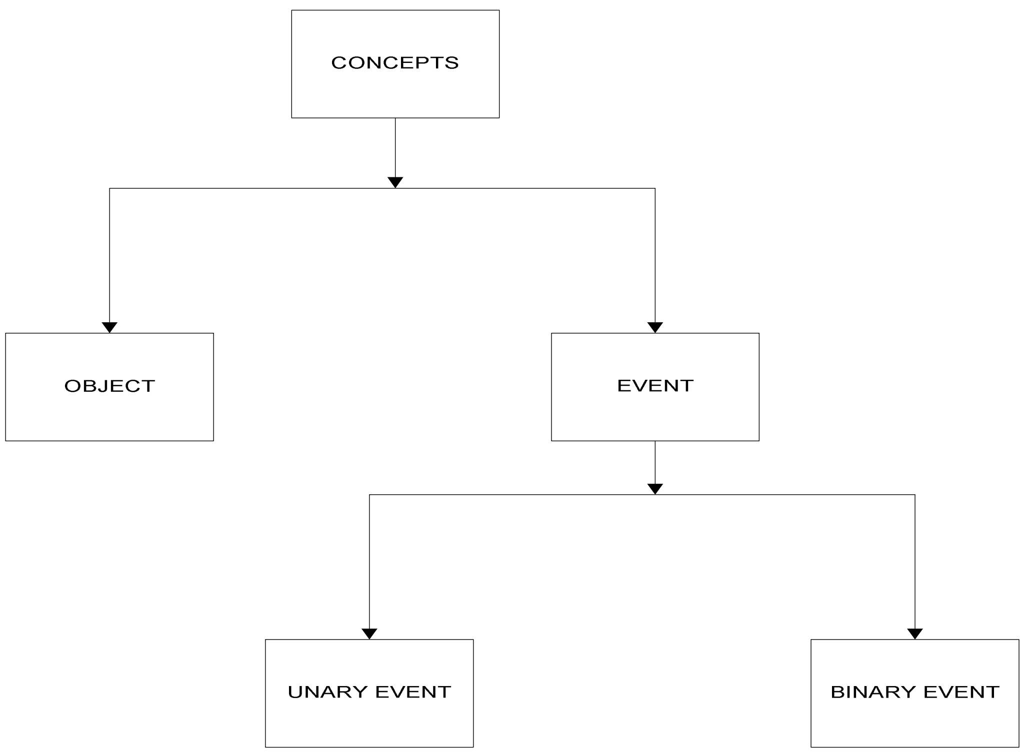

Figure 5.10: Concept Type Hierarchy

```
<owlx:Class owlx:name="#Attacker" owlx:complete="false">
    <owlx:Class owlx:name="#Player"/>
</owlx:Class>
```

Figure 5.11: An Object Description as an OWL Concept

```
<owlx:ObjectProperty owlx:name="#score">
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Team"/>
</owlx:ObjectProperty>
```

Figure 5.12: An Event Description as an OWL Relation

After determining the concept types and getting the ontology support, the most important step is to represent them in MPEG-7. The objects and events which are defined in the scope of time intervals of the video are represented using MPEG-7 descriptors and descriptor schemes. Therefore, the relations of the objects and events with the time intervals should be represented as well as their definitions.

An example of MPEG-7 descriptions of the objects and events are shown in Figure 5.13 and 5.14. Figure 5.13 shows an object description meaning "Henry is an attacker". Figure 5.14 shows an event description meaning "Object with id obj0_0_1 assists the object with id obj0_0_2". The objects with ids obj0_0_1 and obj0_0_2 are also described in the form of the object description in Figure 5.13.

```
<SemanticBase id="obj0_0" xsi:type="ObjectType">
    <Label>
        <Name>Henry</Name>
    </Label>
    <Definition>
        <StructuredAnnotation>
            <WhatObject>
                <Name>Attacker</Name>
            </WhatObject>
        </StructuredAnnotation>
    </Definition>
</SemanticBase>
```

Figure 5.13: MPEG-7 Object Description

```
<SemanticBase id="evt0_0" xsi:type="EventType">
    <Label>
        <Name>assist</Name>
    </Label>
    <Relation type="urn:...:agentOf" source="obj0_0_1"/>
    <Relation type="urn:...:patientOf" source="obj0_0_2"/>
</SemanticBase>
```

Figure 5.14: MPEG-7 Event Description

The video time intervals should also be represented. This is done using the video segment description scheme of the MPEG-7 standard. For each time interval, a video segment is defined with the start point and duration. An example of video segment is shown in Figure 5.15. In this figure, a video segment including an object and an event is described. In particular, the object with id obj0_0 and the event with id evt0_0 are meant to exist in this video segment starting at “second: 10 frame: 18” and ending after 69 frames. obj0_0 and evt0_0 are also described in the form of the object and event descriptions in Figure 5.12 and Figure 5.13.

```
<VideoSegment id="segment0">
    <Relation type="urn:...:hasMediaPerceptionOfObject" target="obj0_0"/>
    <Relation type="urn:...:hasMediaPerceptionOfEvent" target="evt0_0"/>
    <MediaTime>
        <MediaTimePoint>T00:00:10:18F25</MediaTimePoint>
        <MediaIncrDuration mediaTimeUnit="PT1N25F">69</MediaIncrDuration>
    </MediaTime>
</VideoSegment>
```

Figure 5.15: Video Segment

Concerning all these features, an overall MPEG-7 document describing the segments of a video is shown in APPENDIX A.

5.2.3.3 OWL Based Ontology Instance

The other key data concept in the infrastructure is the OWL Based Ontology Instance. This standardized knowledge is acquired after the conversion process of the "MPEG-7 to SWRL/OWL converter". It produces OWL concept and relation instances, and RULEML rule definitions in the context of SWRL from the MPEG-7 objects and events as we have discussed in 5.2.1.2.

An example of an ontology instance is shown in APPENDIX B.

```
<owlx:Individual owlx:name="Anelka">
    <owlx:type owlx:name="Attacker"/>
</owlx:Individual>
```

Figure 5.16: An Object Instance in OWL Based Ontology Instance

```
<owlx:Individual owlx:name="Umit">
    <owlx:type owlx:name="Defender"/>
    <owlx:ObjectPropertyValue owlx:property="assist">
        <owlx:Individual owlx:name="Anelka"/>
    </owlx:ObjectPropertyValue>
</owlx:Individual>
```

Figure 5.17: An Event Instance in OWL Based Ontology Instance

For each video segment, an ontology instance is generated. Combination of the video segments produces the full video content. But, the queries are applied to each segment separately. Therefore, the number of video segments for a video is equal to the number of query applications for that video in one application.

5.2.3.4 Query

A query is a data type that the users create for the reasoner in order to retrieve the video contents. The queries given in the reasoner interface is standardized first. The query standard for the infrastructure is SPARQL. Therefore, the queries given in the form of object-event queries are converted to SPARQL queries. Query Examples are shown in Table 5.1. In the table both two types of queries (object, event) are represented with their SPARQL mappings.

Mainly, the queries are sent to the reasoner interface and used in the communication of KAON2 reasoner. KAON2 answers SPARQL queries and forms the results. However, result set is not enough as the object or event name-type specifications may be entered in order to restrict the result set. So, the specifications on the object name and the event name or roles are taken into account after the general SPARQL query results are executed.

Table 5.1: SPARQL Query Examples

INFORMAL QUERY	SPARQL QUERY
List the players in the videos. (Object Query)	SELECT ?x WHERE { ?x rdf:type <http://kaon2.semanticweb.org/football_ontology_owl#Player>}
List the player-team tuples in which the player scores against the team. (Event Query)	SELECT ?x ?y WHERE { ?x rdf:type <http://kaon2.semanticweb.org/football_ontology_owl#Player> ; <http://kaon2.semanticweb.org/football_ontology_owl#score> ?y}

CHAPTER 6

CONCLUSION

6.1 Observations

The implemented infrastructure for the semantic retrieval of video contents developed in the scope of this thesis brought important issues to the multimedia semantics in many aspects. First of all, the resulting infrastructure holds separate components which can easily be isolated from each other and work with standardized data forms such as MPEG-7 and SWRL/OWL. From this point of view, the system has a flexible structure which provides integrity with the other parts of the infrastructure.

From another point of view, applying semantic web technologies into the multimedia semantics domain and implementing it in a standardized way is an important challenge. This approach will broaden the horizons about the standardized applications of semantic web and make the semantic web closely related with multimedia semantics.

Consequently, enriching the multimedia retrieval tasks with domain knowledge improves the quality of content retrieval. This can be seen by comparing the systems [2, 15] with the infrastructure developed in this thesis concerning this context. A WordNet (Language Ontology on English) based query interface [15] is integrated to the MPEG-7 based video database system [2]. Thus, WordNet semantics (word relations) are useful to improve the query answer qualities. But, the implemented infrastructure will be able to answer more semantic queries compared with the system in [15] as a result of the advantages of the domain knowledge.

6.2 Future Work

There are two main points about the application of the infrastructure in future studies. First, enabling the users determine the video segments may cause chaotic results in some cases because of the fact that the segments may overlap and the system lacks the overlapped parts by not considering the combination of these parts. Therefore, a prevention technique for the segment overlaps is needed. In this context, the solution for the problem can be developing or integrating an automatic key frame detector. By using this detector, video segments can be separated without overlaps automatically.

Secondly, the time complexity of the reasoning mechanism is very high compared to the keyword-base retrieval tools as the reasoning process should be applied in all video segments. In order to overcome this problem Truth Maintenance Systems can be helpful. Each deduction could be added into a local history and firstly the history is searched for a new query. So, the time complexity will be decreased concerning this advantage of query history.

In addition to these two points, integrating WordNet with the annotation environment will be another challenge. In this perspective, the semantics of the querying task will be stronger concerning the additional linguistic inferences as a result of this integration.

Consequently, the query answer accuracy and the time complexity will be the most important factors appearing against the improvement of the system in future work concerning the architecture of the infrastructure.

REFERENCES

[1] B. Motik and U. Sattler, "A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes", *Proc. of the 13th International Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR 2006)*, Phnom Penh, Cambodia, November, 2006.

[2] Çiğdem Çelik, "An MPEG-7 Database System for Content-Based Management and Retrieval", *MSc Thesis, Graduate School of Natural and Applied Sciences of Middle East Technical University*, September 2005.

[3] C. Tsinaraki, P. Polydoros, N. Moumoutzis, and S. Christodoulakis, "Coupling Owl with MPEG-7 and TVAnytime for Domain-specific Multimedia Information Integration and Retrieval", *Proceedings of RIAO* 2004, (Avignon, France), April 2004.

[4] M Bertini, A Del Bimbo, C Torniai, "Enhanced Ontologies for Video Annotation and Retrieval", *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, Hilton, Singapore, 2005.

[5] C. Tsinaraki, P. Polydoros, S. Christodoulakis, "Integration of OWL Ontologies in MPEG-7 and TVAnytime Compliant Semantic Indexing", *Proceedings of CaiSE*, 2004.

[6] G. Antoniou, E. Franconi, and F. van Harmelen, "Introduction to Semantic Web Ontology Languages", 28 April 2005.

[7] B. Motik, "KAON2: Infrastructure for Managing OWL-DL and SWRL Ontologies", *http://kaon2.semanticweb.org/*

[8] S. Bloehdorn, N. Simou, V. Tzouvaras, K. Petridis, S. Handschuh, Y. Avrithis, I. Kompatsiaris, S. Staab and M. G. Strintzis, "Knowledge Representation for Semantic Multimedia Content Analysis and Reasoning", *EWIMT*, 2004.

[9] T. di Dottorato, “Models, Methodologies and Instruments for the Next Generation Web, Applied to the Delivery of Services”, *Thesis, Politecnico di Torino*, February 2006.

[10] P. Salembier and J. R. Smith, “MPEG-7 Multimedia Description Schemes”, *IEEE Trans. Circuits Syst. for Video Technol.,* August 2001.

[11] J. M. Martinez, “MPEG-7 Overview Version 10”, *http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm*

[12] J.M. Martinez, R. Koenen, F. Pereira, “MPEG-7: The Generic Multimedia Content Description Standard”, *IEEE MultiMedia Part 1, v.9 n.2,* pp.78-87, April 2002.

[13] V. Tzouvaras, S. Dasiopoulou, F. Martin-Recuerda, G.Stoilos, Y. Kompatsiaris and G. Stamou, “Multimedia Analysis and Annotation Requirements for the Semantic Web”, *EWIMT*, pp. 443-450, 2005.

[14] J. van Ossenbruggen, G. Stamou, J. Z. Pan, “Multimedia Annotations and the Semantic Web”, *Multimedia, IEEE, 2006*, USA, January-March 2006.

[15] G. Erözel, “Natural Language Interface On a Video Data Model”, *MSc Thesis, Graduate School of Natural and Applied Sciences of Middle East Technical University*, September 2005.

[16] Yu Deng, “OntoELAN: An Ontological Annotation System for Linguistic Multimedia Data”, *MSc Thesis, Graduate School of Wayne State University*, August 2004.

[17] A. Maedche, “Ontology Learning for the Semantic Web”, *Book published by Springer*, 1 January 2002.

[18] F. van Harmelen and D. Fensel, “Practical Knowledge Representation for the Web”, *In Proc. IJCAI’99 Workshop on Intelligent Information Integration*, 1999.

[19] S. Hammiche, S. Benbernou, M.S. Hacid, A. Vakali, “Semantic Retrieval of Multimedia Data”, *Proceedings of the 2nd ACM international workshop on Multimedia Databases,* Washington*, DC, USA, 2004.*

[20] T. Berners-Lee, “Semantic Web Layer Cake”, *Semantic Web - XML2000, http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html*, 2005.

[21] M. Köprülü, N. K. Çiçekli, A. Yazıcı, "Spatio-temporal Querying in Video Databases", *Information Sciences 160, 2004, Elsevier Science*, pp. 131-152, 2004.

[22] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, "SWRL: A Semantic Web Rule Language Combining OWL and RuleML", *WC3 Member Submission*, 2004.

[23] S. Bechhofer, I. Horrocs, "The Wonder Web Ontology Language Layer", *Report and Tutorial*. 7 February 2003.

[24] T. Athanasiadis, V. Tzouvaras, K. Petridis, F. Precioso, Y. Avrithis and Y. Kompatsiaris, "Using a Multimedia Ontology Infrastructure for Semantic Annotation of Multimedia Content", *Proceedings of 5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot '05) at the 4th International Semantic Web Conference (ISWC 2005)*, 2005.

[25] H. Jiang and A.K. Elmagarmid, "WVTDB—A Semantic Content Based Videotext Database System on the World Wide Web", *IEEE Transactions on Knowledge and Data Engineering 10 6 (1998)*, pp. 947–966, 1998.

APPENDIX A

AN EXAMPLE MPEG-7 DEFINITION OF A VIDEO SEGMENT


```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<Mpeg7 xmlns="urn:mpeg:mpeg7:schema:2001"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
       xmlns:xml="http://www.w3.org/XML/1998/namespace"
       xsi:schemaLocation="urn:mpeg:mpeg7:schema:2001 Mpeg7-
2001.xsd">
    <Description xsi:type="ContentEntityType">
        <MultimediaContent xsi:type="VideoType">
            <Video xsi:type="VideoSegmentType">
                <MediaInformation>
                    <Header xsi:type="DescriptionMetadataType">
                        <Comment>
                            <FreeTextAnnotation>
                                Description of soccer game video
                            </FreeTextAnnotation>
                        </Comment>
                    </Header>
                    <MediaProfile master="false">
                        <MediaInstance>
                            <InstanceIdentifier>Soccer
game</InstanceIdentifier>
                            <MediaLocator>

<MediaUri>C:/Documents(empty117)and(empty117)Settings/samet/Belgeler
im/TV_CH28_0619_000106.mpg</MediaUri>
                            </MediaLocator>
                        </MediaInstance>
                    </MediaProfile>
                </MediaInformation>
                <TemporalDecomposition
xsi:type="VideoSegmentTemporalDecompositionType">
                    <VideoSegment id="segment0">
                        <Relation
type="urn:...:hasMediaPerceptionOfObject" target="obj0_0"/>
                        <Relation
type="urn:...:hasMediaPerceptionOfEvent" target="evt0_0"/>
                        <MediaTime>
```

```
<MediaTimePoint>T00:00:10:18F25</MediaTimePoint>
                            <MediaIncrDuration
mediaTimeUnit="PT1N25F">69</MediaIncrDuration>
                        </MediaTime>
                    </VideoSegment>
                </TemporalDecomposition>
            </Video>
        </MultimediaContent>
    </Description>
    <Description xsi:type="SemanticDescriptionType">
        <Semantics id="sem1">
            <Label>
                <Name>soccer</Name>
            </Label>
            <SemanticBase id="obj0_0" xsi:type="ObjectType">
                <Label>
                    <Name>Henry</Name>
                </Label>
                <Definition>
                    <StructuredAnnotation>
                        <WhatObject>
                            <Name>Attacker</Name>
                        </WhatObject>
                    </StructuredAnnotation>
                </Definition>
            </SemanticBase>
            <SemanticBase id="evt0_0" xsi:type="EventType">
                <Label>
                    <Name>assist</Name>
                </Label>
                <Relation type="urn:...:agentOf" source="obj0_0_1"/>
                <Relation type="urn:...:patientOf"
source="obj0_0_2"/>
            </SemanticBase>
            <SemanticBase id="obj0_0_1" xsi:type="AgentObjectType">
                <Label>
                    <Name>Henry</Name>
                </Label>
            </SemanticBase>
            <SemanticBase id="obj0_0_2" xsi:type="ObjectType">
                <Label>
                    <Name>Zidane</Name>
                </Label>
            </SemanticBase>
        </Semantics>
    </Description>
</Mpeg7>
```

APPENDIX B

AN EXAMPLE ONTOLOGY INSTANCE


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE swrlx:Ontology [
    <!ENTITY a
'http://kaon2.semanticweb.org/football_ontology_owl#'>
]>

<swrlx:Ontology
swrlx:name="http://kaon2.semanticweb.org/football_ontology_v02"
    xml:base="http://kaon2.semanticweb.org/football_ontology_v02"
    xmlns:owlx="http://www.w3.org/2003/05/owl-xml#"
    xmlns:ruleml="http://www.w3.org/2003/11/ruleml#"
    xmlns:swrlx="http://www.w3.org/2003/11/swrlx#">

<owlx:Class owlx:name="#Attacker" owlx:complete="false">
    <owlx:Class owlx:name="#Player"/>
</owlx:Class>

<owlx:Class owlx:name="#Coach" owlx:complete="false">
    <owlx:Class owlx:name="#Person"/>
</owlx:Class>

<owlx:Class owlx:name="#Defender" owlx:complete="false">
    <owlx:Class owlx:name="#Player"/>
</owlx:Class>

<owlx:Class owlx:name="#Emotion" owlx:complete="false">
    <owlx:Class owlx:name="#Object"/>
</owlx:Class>

<owlx:Class owlx:name="#Fans" owlx:complete="false">
    <owlx:Class owlx:name="#Group"/>
</owlx:Class>

<owlx:Class owlx:name="#Frame" owlx:complete="false">
    <owlx:Class owlx:name="#Object"/>
</owlx:Class>
```

```
<owlx:Class owlx:name="#Goalkeeper" owlx:complete="false">
    <owlx:Class owlx:name="#Player"/>
</owlx:Class>

<owlx:Class owlx:name="#Midfielder" owlx:complete="false">
    <owlx:Class owlx:name="#Player"/>
</owlx:Class>

<owlx:Class owlx:name="#Occasion" owlx:complete="false">
    <owlx:Class owlx:name="#Frame"/>
</owlx:Class>

<owlx:Class owlx:name="#Player" owlx:complete="false">
    <owlx:Class owlx:name="#Person"/>
</owlx:Class>

<owlx:Class owlx:name="#Referee" owlx:complete="false">
    <owlx:Class owlx:name="#Person"/>
</owlx:Class>

<owlx:Class owlx:name="#Region" owlx:complete="false">
    <owlx:Class owlx:name="#Object"/>
</owlx:Class>

<owlx:Class owlx:name="#Replay" owlx:complete="false">
    <owlx:Class owlx:name="#Frame"/>
</owlx:Class>

<owlx:Class owlx:name="#Supporter" owlx:complete="false">
    <owlx:Class owlx:name="#Person"/>
</owlx:Class>

<owlx:Class owlx:name="#Team" owlx:complete="false">
    <owlx:Class owlx:name="#Group"/>
</owlx:Class>

<owlx:Class owlx:name="#FileName" owlx:complete="false">
    <owlx:Class owlx:name="#MetaInfo"/>
</owlx:Class>

<owlx:Class owlx:name="#StartTime" owlx:complete="false">
    <owlx:Class owlx:name="#MetaInfo"/>
</owlx:Class>

<owlx:Class owlx:name="#EndTime" owlx:complete="false">
    <owlx:Class owlx:name="#MetaInfo"/>
</owlx:Class>

<owlx:ObjectProperty owlx:name="#approve">
    <owlx:domain owlx:class="#Referee"/>
    <owlx:range owlx:class="#Occasion"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#assist">
```

```
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Player"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#belongPerson">
    <owlx:domain owlx:class="#Person"/>
    <owlx:range owlx:class="#Group"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#belongRegion">
    <owlx:domain owlx:class="#Region"/>
    <owlx:range owlx:class="#Group"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#dribble">
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Region"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#feelGroup">
    <owlx:domain owlx:class="#Group"/>
    <owlx:range owlx:class="#Emotion"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#feelPersonal">
    <owlx:domain owlx:class="#Person"/>
    <owlx:range owlx:class="#Emotion"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#feint">
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Player"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#foul">
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Player"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#inside">
    <owlx:domain owlx:class="#Object"/>
    <owlx:range owlx:class="#Region"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#keep">
    <owlx:domain owlx:class="#Goalkeeper"/>
    <owlx:range owlx:class="#Region"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#miss">
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Team"/>
</owlx:ObjectProperty>
```

```xml
<owlx:ObjectProperty owlx:name="#outside">
    <owlx:domain owlx:class="#Object"/>
    <owlx:range owlx:class="#Region"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#pass">
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Player"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#punish">
    <owlx:domain owlx:class="#Referee"/>
    <owlx:range owlx:class="#Player"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#reject">
    <owlx:domain owlx:class="#Referee"/>
    <owlx:range owlx:class="#Occasion"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#save">
    <owlx:domain owlx:class="#Goalkeeper"/>
    <owlx:range owlx:class="#Team"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#score">
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Team"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#shoot">
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Region"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#spirint">
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Region"/>
</owlx:ObjectProperty>

<owlx:ObjectProperty owlx:name="#touch">
    <owlx:domain owlx:class="#Player"/>
    <owlx:range owlx:class="#Object"/>
</owlx:ObjectProperty>

<owlx:Individual owlx:name="#C-\soccerVideo.mpg">
    <owlx:type owlx:name="#FileName"/>
</owlx:Individual>

<owlx:Individual owlx:name="#00-01-08-00">
    <owlx:type owlx:name="#StartTime"/>
</owlx:Individual>
```

```
<owlx:Individual owlx:name="#00-01-27-16">
    <owlx:type owlx:name="#EndTime"/>
</owlx:Individual>

<owlx:Individual owlx:name="#FB">
    <owlx:type owlx:name="#Team"/>
</owlx:Individual>

<owlx:Individual owlx:name="#GS">
    <owlx:type owlx:name="#Team"/>
</owlx:Individual>

<owlx:Individual owlx:name="#Mondragon">
    <owlx:type owlx:name="#Goalkeeper"/>
    <owlx:ObjectPropertyValue owlx:property="#belongPerson">
        <owlx:Individual owlx:name="#GS"/>
    </owlx:ObjectPropertyValue>
    <owlx:ObjectPropertyValue owlx:property="#save">
        <owlx:Individual owlx:name="#FB"/>
    </owlx:ObjectPropertyValue>
</owlx:Individual>

<owlx:Individual owlx:name="#Tuncay">
    <owlx:type owlx:name="#Attacker"/>
    <owlx:ObjectPropertyValue owlx:property="#belongPerson">
        <owlx:Individual owlx:name="#FB"/>
    </owlx:ObjectPropertyValue>
    <owlx:ObjectPropertyValue owlx:property="#shoot">
        <owlx:Individual owlx:name="#goal"/>
    </owlx:ObjectPropertyValue>
</owlx:Individual>

<owlx:Individual owlx:name="#angry">
    <owlx:type owlx:name="#Emotion"/>
</owlx:Individual>

<owlx:Individual owlx:name="#ball">
    <owlx:type owlx:name="#Object"/>
</owlx:Individual>

<owlx:Individual owlx:name="#goal">
    <owlx:type owlx:name="#Region"/>
    <owlx:ObjectPropertyValue owlx:property="#belongRegion">
        <owlx:Individual owlx:name="#GS"/>
    </owlx:ObjectPropertyValue>
</owlx:Individual>

<owlx:Individual owlx:name="#happy">
    <owlx:type owlx:name="#Emotion"/>
</owlx:Individual>

<owlx:Individual owlx:name="#out">
    <owlx:type owlx:name="#Region"/>
</owlx:Individual>
```

```
<owlx:Individual owlx:name="#post">
    <owlx:type owlx:name="#Region"/>
</owlx:Individual>

<owlx:Individual owlx:name="#seat">
    <owlx:type owlx:name="#Region"/>
</owlx:Individual>

<owlx:Individual owlx:name="#sorry">
    <owlx:type owlx:name="#Emotion"/>
</owlx:Individual>

<ruleml:imp>
    <ruleml:_head>
        <swrlx:individualPropertyAtom swrlx:property="#foul">
            <ruleml:var>T</ruleml:var>
            <ruleml:var>X</ruleml:var>
        </swrlx:individualPropertyAtom>
    </ruleml:_head>
    <ruleml:_body>
        <swrlx:individualPropertyAtom swrlx:property="#punish">
            <ruleml:var>Y</ruleml:var>
            <ruleml:var>T</ruleml:var>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom
swrlx:property="#belongPerson">
            <ruleml:var>X</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
    </ruleml:_body>
</ruleml:imp>
<ruleml:imp>
    <ruleml:_head>
        <swrlx:individualPropertyAtom swrlx:property="#miss">
            <ruleml:var>X</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
    </ruleml:_head>
    <ruleml:_body>
        <swrlx:individualPropertyAtom swrlx:property="#shoot">
            <ruleml:var>X</ruleml:var>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom
swrlx:property="#belongRegion">
            <owlx:Individual owlx:name="#goal"/>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#outside">
            <owlx:Individual owlx:name="#ball"/>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
    </ruleml:_body>
```

```
</ruleml:imp>
<ruleml:imp>
    <ruleml:_head>
        <swrlx:individualPropertyAtom swrlx:property="#save">
            <ruleml:var>X</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
    </ruleml:_head>
    <ruleml:_body>
        <swrlx:individualPropertyAtom swrlx:property="#shoot">
            <ruleml:var>Y</ruleml:var>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom
swrlx:property="#belongPerson">
            <ruleml:var>Y</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#outside">
            <owlx:Individual owlx:name="#ball"/>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#touch">
            <ruleml:var>X</ruleml:var>
            <owlx:Individual owlx:name="#ball"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#keep">
            <ruleml:var>X</ruleml:var>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
    </ruleml:_body>
</ruleml:imp>
<ruleml:imp>
    <ruleml:_head>
        <swrlx:individualPropertyAtom swrlx:property="#score">
            <ruleml:var>T</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
    </ruleml:_head>
    <ruleml:_body>
        <swrlx:individualPropertyAtom swrlx:property="#shoot">
            <ruleml:var>T</ruleml:var>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom
swrlx:property="#belongRegion">
            <owlx:Individual owlx:name="#goal"/>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#inside">
            <owlx:Individual owlx:name="#ball"/>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#approve">
```

```
            <ruleml:var>X</ruleml:var>
            <ruleml:var>Y</ruleml:var>
        </swrlx:individualPropertyAtom>
    </ruleml:_body>
</ruleml:imp>
<ruleml:imp>
    <ruleml:_head>
        <swrlx:individualPropertyAtom swrlx:property="#pass">
            <ruleml:var>X</ruleml:var>
            <ruleml:var>Y</ruleml:var>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#score">
            <ruleml:var>Y</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
    </ruleml:_head>
    <ruleml:_body>
        <swrlx:individualPropertyAtom swrlx:property="#assist">
            <ruleml:var>X</ruleml:var>
            <ruleml:var>Y</ruleml:var>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom
swrlx:property="#belongPerson">
            <ruleml:var>T</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#keep">
            <ruleml:var>T</ruleml:var>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
    </ruleml:_body>
</ruleml:imp>
<ruleml:imp>
    <ruleml:_head>
        <swrlx:individualPropertyAtom swrlx:property="#shoot">
            <ruleml:var>X</ruleml:var>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom
swrlx:property="#belongPerson">
            <owlx:Individual owlx:name="#goal"/>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#outside">
            <owlx:Individual owlx:name="#ball"/>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
    </ruleml:_head>
    <ruleml:_body>
        <swrlx:individualPropertyAtom swrlx:property="#miss">
            <ruleml:var>X</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
    </ruleml:_body>
```

```
</ruleml:imp>
<ruleml:imp>
    <ruleml:_head>
        <swrlx:individualPropertyAtom swrlx:property="#shoot">
            <ruleml:var>T</ruleml:var>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom
swrlx:property="#belongRegion">
            <owlx:Individual owlx:name="#goal"/>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#inside">
            <owlx:Individual owlx:name="#ball"/>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
    </ruleml:_head>
    <ruleml:_body>
        <swrlx:individualPropertyAtom swrlx:property="#score">
            <ruleml:var>T</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
    </ruleml:_body>
</ruleml:imp>
<ruleml:imp>
    <ruleml:_head>
        <swrlx:individualPropertyAtom
swrlx:property="#belongPerson">
            <ruleml:var>Y</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#outside">
            <owlx:Individual owlx:name="#ball"/>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#touch">
            <ruleml:var>X</ruleml:var>
            <owlx:Individual owlx:name="#ball"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#keep">
            <ruleml:var>X</ruleml:var>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
    </ruleml:_head>
    <ruleml:_body>
        <swrlx:individualPropertyAtom swrlx:property="#shoot">
            <ruleml:var>Y</ruleml:var>
            <owlx:Individual owlx:name="#goal"/>
        </swrlx:individualPropertyAtom>
        <swrlx:individualPropertyAtom swrlx:property="#save">
            <ruleml:var>X</ruleml:var>
            <ruleml:var>Z</ruleml:var>
        </swrlx:individualPropertyAtom>
    </ruleml:_body>
```

```
</ruleml:imp>
</swrlx:Ontology>
```