AUTOMATIC SEMANTIC CONTENT EXTRACTION IN VIDEOS USING A SPATIO-TEMPORAL ONTOLOGY MODEL

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

BY

YAKUP YILDIRIM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

MARCH 2009

Approval of the thesis:

AUTOMATIC SEMANTIC CONTENT EXTRACTION IN VIDEOS USING A SPATIO-TEMPORAL ONTOLOGY MODEL

submitted by **YAKUP YILDIRIM** in partial fullfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

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

Prof. Dr. Müslim Bozyiğit _____
Head of Department, **Computer Engineering**

Prof. Dr. Adnan Yazıcı _____
Supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. İsmail Hakkı Toroslu _____
Computer Engineering Dept., METU

Prof. Dr. Adnan Yazıcı _____
Computer Engineering Dept., METU

Prof. Dr. Özgür Ulusoy _____
Computer Engineering Dept., Bilkent University

Assoc. Prof. Dr. Ahmet Coşar _____
Computer Engineering Dept., METU

Assist. Prof. Dr. Murat Koyuncu _____
Computer Engineering Dept., Atılım University

Date: _____

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 : Yakup Yıldırım

Signature :

ABSTRACT

AUTOMATIC SEMANTIC CONTENT EXTRACTION IN VIDEOS USING A SPATIO-TEMPORAL ONTOLOGY MODEL

Yıldırım, Yakup

PhD., Department of Computer Engineering

Supervisor: Prof. Dr. Adnan Yazıcı

March 2009, 147 pages

Recent increase in the use of video in many applications has revealed the need for extracting the content in videos. Raw data and low-level features alone are not sufficient to fulfill the user's need; that is, a deeper understanding of the content at the semantic level is required. Currently, manual techniques are being used to bridge the gap between low-level representative features and high-level semantic content, which are inefficient, subjective and costly in time and have limitations on querying capabilities. Therefore, there is an urgent need for automatic semantic content extraction from videos. As a result of this requirement, we propose an automatic semantic content extraction system for videos in terms of object, event and concept extraction. We introduce a general purpose ontology-based video semantic content model that uses object definitions, spatial relations and temporal relations in event and concept definitions. Various relation types are defined to describe fuzzy spatio-temporal relations between ontology classes. Thus, the video semantic content model is utilized to construct domain ontologies. In addition, domain ontologies are enriched with rule definitions to lower spatial relation computation cost and to be able to define some complex situations more effectively. As a case study, we have performed a number experiments for event and concept extraction in videos for basketball and surveillance domains. We have obtained satisfactory precision and recall rates for object,

event and concept extraction. A domain independent application for the proposed framework has been fully implemented and tested.

Keywords: Semantic Content Extraction, Video Content Modeling, Ontology

ÖZ

KONUMSAL VE ZAMANSAL BİR ONTOLOJİ MODELİ KULLANARAK VİDEOLARDAN OTOMATİK ANLAMSAL İÇERİK ÇIKARIMI

Yıldırım, Yakup

Doktora, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Adnan Yazıcı

Mart 2009, 147 sayfa

Bir çok uygulamada video kullanımının son dönemdeki artışı, videolardan içeriğin elde edilme ihtiyacını ortaya çıkarmıştır. Ham video verisi ve alt seviye özellikler tek başına kullanıcı ihtiyaçlarını tam olarak karşılayamadığı için, içeriğin derinlemesine incelenerek anlamsal seviyede ele alınması gerekmektedir. Günümüzde, alt seviye temsili özellikler ile üst seviye anlamsal içerik arasında yer alan boşluğun kapatılması için yetersiz, öznel, zaman kaybına ve sorgu kabiliyetlerinde kısıtlamalara sebep olan manuel teknikler kullanılmaktadır. Bu nedenle, videolardan anlamsal içeriğin otomatik olarak çıkarılma ihtiyacı zorunlu hale gelmiştir. Bu ihtiyacı karşılamak üzere, nesne, olay ve kavram çıkarımını otomatik olarak yapan bir video anlamsal içerik çıkarım sistemi önermekteyiz. Olay ve kavram tanımlarında nesne tanımlarını ve konumsal ve zamansal ilişkileri kullanan, genel amaçlı ontoloji destekli anlamsal bir video modeli ortaya koymaktayız. Ontoloji sınıfları arasında yer alan bulanık konumsal ve zamansal ilişkileri tanımlamak amacı ile çeşitli ilişki tipleri oluşturulmuştur. Bu anlamsal video modeli alan ontolojilerinin oluşturulmasında kullanılmaktadır. Buna ek olarak, alan ontolojileri, konumsal ilişki hesaplama maliyetini düşürmek ve bazı karmaşık durumların daha etkin tanımlanabilmesi için kural tanımlarıyla zenginleştirilmiştir. Örnek olay incelemesi olarak basketbol ve gözetleme alanları için videolardan olay ve kavram çıkarımı üzerine deneyler yapılmıştır. Nesne, olay ve kavram çıkarımı

için tatmin edici geri getirme ve duyarlılık yüzdeleri elde edilmiştir. Önerilen çatı için alan bağımsız bir uygulama geliştirilmiş ve test edilmiştir.

Anahtar Kelimeler: Anlamsal İçerik Çıkarımı, Video İçerik Modelleme, Ontoloji

ACKNOWLEDGMENTS

I would like to express my inmost gratitude to my supervisor Prof. Dr. Adnan Yazıcı for his guidance, advice, insight throughout the research and trust on me. It is an honour for me to share his knowledge and wisdom. He has been a constant source of support, inspiration and common sense throughout the course of my studies. His continuous support made this thesis possible.

I would like to thank to my thesis jury members Prof. Dr. İsmail Hakkı Toroslu, Prof. Dr. Özgür Ulusoy, Assoc. Prof. Ahmet Coşar and Asst. Prof. Murat Koyuncu for their valuable comments and guidance.

Also, I am grateful to Turgay Yılmaz for his support especially during the implementation phase of this thesis.

I am very thankful to my beloved family, my dear wife Zülfiye and our sweet daughter Beyza, for believing in me and supporting me during this study. I am greatly indebted to the sacrifice they made so that I could have the time to complete the work. I am also grateful to my parents who always encouraged me to finish the thesis.

To My Family

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ACL	Agent Communication Language
API	Application Programming Interface
ASCEF	Automatic Semantic Content Extraction Framework
ASR	Automatic Speech Recognition
AVIS	The Advanced Video Information System
BDA	Boundary Detection Accuracy
BRDF	Best Representative and Discriminative Feature
CBVIR	Content Based Video Information Retrieval
DAML	DARPA Agent Markup Language
DARPA	The Defense Advanced Research Projects Agency
DBN	Dynamic Bayesian Networks
DDL	Description Definition Language
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
Ds	Descriptors
DSs	Description Schemes
ERL	Event Recognition Language
GA	Genetic Algorithm
GFO	General Formal Ontology
GUI	Graphical User Interface
HMM	Hidden Markov Models
HTML	Hyper Text Markup Language
IOVA	Integrated Ontology for Video Annotation
IR	Information Retrieval
ISO	International Organization for Standardization

JENA	A Semantic Web Framework for Java
KB	Knowledge Base
MBR	Minumum Bounding Rectangle
MOM	Multimedia Ontology Manager
MPEG	Moving Picture Experts Group
Ncut	Normalized Cut(Image Segmentation)
OIL	Ontology Inference Layer
OWL	Ontology Web Language
PNF	Past Now Feature
QBIC	Query By Image Content
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
SHOE	Simple HTML Ontology Extensions
SPARQL	SPARQL Protocol and RDF Query Language
SUMO	Suggested upper merged ontology.
SWRL	Semantic Web Rule Language
TIME	Time Interval Multimedia Event
VERL	Video Event Recognition Language
VISCOM	Video Semantic Content Model
XM	eXperimentation Model(MPEG-7 Reference Software)
XML	Extensible Markup Language
XOL	XML based ontology-exchange language
W3C	World Wide Web Consortium

CHAPTER 1

INTRODUCTION

The rapid increase in the available amount of video data has revealed an urgent need to develop intelligent methods to model and extract the video content. Typical applications in which modeling and extracting video content is crucial include surveillance, video-on-demand systems, intrusion detection, border monitoring, sports, criminal investigation systems and many others. The ultimate goal is to enable users to retrieve the desired content from massive amounts of video data in an efficient and semantically meaningful manner.

Free browsing, text-based retrieval and content-based retrieval are basic ways of retrieving the video data [76]. Free browsing is an inefficient and time-consuming process and it becomes completely impractical for large video data. The success of text-based retrieval systems is limited to the quality of the metadata produced during the cataloguing process, which can often be incomplete, inaccurate and ambiguous [76]. In order to overcome the inefficiencies, limitations and scalability problems of free browsing and text-based retrieval, many research groups [27, 29, 40, 44, 45, 68, 76, 79, 88, 97, 100] have investigated possible ways of retrieving video based solely on video content.

There are basically three levels of video content as raw video data, low-level feature and semantic content. First, raw video data consists of elementary physical video units together with some general video attributes such as format, length and frame rate. Second, low-level features are characterized by audio, text and visual features such as texture and color distribution. Third, semantic content contains high-level concepts such as objects and events. The first two levels on which content modeling and extraction approaches are based use automatically extracted data, which represent the low-level content of a video, but they hardly provide semantics which is much more appropriate for users. Users are mostly interested in querying and retrieving the video in terms of what the video is about. Therefore, raw video data and low-level features alone are not sufficient to fulfill the user's

need; that is, a deeper understanding of the information at the semantic level is required.

However, it is very difficult to extract semantic content directly from raw video data. This is because video is a temporal sequence of pixel regions without a direct relation to its semantic content [94]. Therefore, many different presentations using different sets of data such as audio, visual features, objects, events, temporality, motion and spatial relations are partially or fully used to model and extract the semantic content. No matter which type of data set is used, the process of extracting the semantic content is complex because it usually requires domain knowledge or user interaction.

While there has been a significant amount of research in this area, most of the previous semantic content extraction studies propose manual methods to extract the semantic content. The major limitations of manually extraction approaches are that they are tedious, subjective and time consuming [96]. Furthermore, they are inefficient and have limitations on querying capabilities. Therefore, the need for automatic semantic content extraction arises.

In order to address this need, in this dissertation, a new framework as an automatic semantic content extraction system for videos, which provides a reasonable approach in bridging the gap between low-level representative features and high-level semantic content in terms of object, event, concept, spatial and temporal relation extraction is proposed. The starting point for the extraction process is object extraction, which is very important and challenging to support content-based video retrieval. Specifically, a genetic algorithm based method for object extraction which supports fuzziness by both making multiple categorization and fuzzy decisions on the objects is used. For each representative frame, objects and spatial relations between objects are extracted. Consecutive representative frames are processed to extract temporal relations, which are the other important facets in the semantic content extraction process. In this context, spatial and temporal relations among semantic contents are extracted automatically considering the uncertainty in relation definitions. Objects, spatial relations between objects and temporal relations between events are utilized in event extraction process. Similarly, objects and events are utilized in concept extraction process.

In this study, in order to address the need for object, event and concept modeling, a domain independent ontology based **VI**deo **S**emantic **CO**ntent **M**odel (VISCOM) that uses objects and spatial/temporal relations in event and concept definitions is developed. Video models for semantic representation should:

- be able to capture and represent various types of information about objects,

- be able to define relationships between objects as well as events and concepts,
- be useful for content extraction and
- allow causal and other inferences to be made from the extracted content.

To address the requirements of semantic video representation listed above, various relation types are defined to describe fuzzy spatio-temporal relations between ontology classes.

VISCOM is utilized to construct domain ontologies which are enriched with rule definitions to lower spatial relation computation costs and to be able to define some complex situations more effectively. Objects, events, domain ontologies and rule definitions are utilized in the automatic event and concept extraction process. The semantic content representation and extraction approach is illustrated in Figure 1.1.

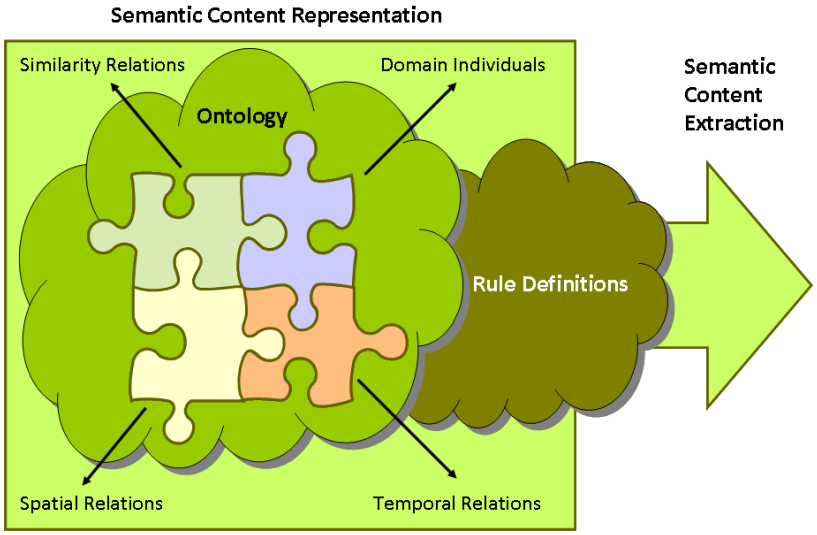


Figure 1.1: Semantic Content Representation and Extraction

Many researchers utilize spatial and/or temporal relations for semantic content representation. Studies such as BilVideo [40], extended-AVIS [70], multiView [43] and classView [44] propose methods using spatial/temporal relations but do not use ontology based models for semantic content representation. [19] presents a video semantic content analysis framework based on a domain ontology that is used to define semantic events with a temporal description logic where event extraction is done manually and event descriptions are conducted by using only temporality. [100] proposes an ontology model using spatio-temporal relations to extract complex events where the extraction process is manual. In [18], each linguistic

concept in the domain ontology is associated with a corresponding visual concept with only temporal relations for soccer videos. In this dissertation, spatial relations between objects and temporal relations between events are utilized together in an ontology-based model to support automatic semantic content extraction.

Automatic event extraction has also been studied by many researchers using different methodologies such as object detection and tracking, multimodality and spatio-temporal derivatives. Most of them propose techniques for specific event type extraction. To the best of our knowledge, up to now there has been no study that proposes a method for automatic concept extraction. In [37], simple periodic events are recognized where event extraction success is highly dependent on the robustness of the tracking. The event recognition methods described in [78] are based on a heuristic method that could not handle multiple-actor events. Event definitions are made through predefined object motions and their temporal behaviour. The bottleneck of this study is its dependence on motion detection. In [57], scenario events are modeled from shape and trajectory features using a hierarchical activity representation extended from [78]. [52] proposes a method to detect events in terms of temporally related chain of directly measurable and highly correlated low-level actions (sub-events) by using only temporal relations.

A domain independent application for the proposed system was fully implemented and tested. As a case study, some experiments were conducted for event and concept extraction in videos for basketball and office surveillance domains. First of all, object, event and concept individuals were determined. Secondly, class individuals were defined for each class type of VISCOM. Thirdly, spatial relation individuals between objects and temporal relation individuals between events were defined. Finally, similarity and role definitions were included in the ontologies. Additionally, a number of domain specific rules were defined. No problem or restriction was encountered during the construction of the ontologies. Satisfactory precision and recall rates in terms of object, event and concept extraction were obtained by using the framework given in this dissertation.

The model and semantic content extraction solution provided in this dissertation can be utilized in various areas such as surveillance, sports and news video applications.

1.1 Contributions of The Dissertation

The aim of this work is to contribute to the state-of-the-art in semantic content extraction from videos by proposing an automatic semantic content extraction methodology enriched

with rules. This is accomplished through the development of an ontology-based semantic content model and a number of semantic content extraction algorithms. The approach proposed in this dissertation differs from other semantic content extraction and representation studies in many directions and contributes in a number of ways to semantic video modeling and semantic content extraction research area. Scientific contributions achieved by this thesis are as follows:

- A reasonable approach to bridging the gap between low-level representative features and high-level semantic contents from a human point of view is provided. It offers automatic mapping from low-level features to high-level contents.
- An automatic semantic content extraction framework for videos is introduced. This approach is different from other semantic content extraction studies because it proposes an automatic framework that is domain and semantic content independent.
- A domain independent ontology-based semantic meta model for videos is proposed to generate domain ontologies where temporal/spatial relations are used for event and concept representation.
- Domain ontologies are enriched with rule definitions to lower spatial relation computation costs and to be able to define some complex situations more effectively.
- Spatial relations between objects, temporal relations between events and domain specific rule definitions are utilized together to make automatic semantic content extraction.
- The success of semantic content extraction is improved by adding fuzziness in class, relation and rule definitions.
- An automatic genetic algorithms based object extraction study is integrated to the proposed system.
- Ontology-based modeling and extraction capabilities are utilized during all phases of the dissertation.

The developed system is a full-fledged framework, capable of extracting the semantic content from videos using the semantic content and rule definitions given by domain experts. We think that, the system developed as a result of this study can be used in practical applications. The only requirement to obtain successful semantic content extraction results

is to use a well and correctly-defined domain ontology and correct object instances as input. The primitive versions of the framework proposed with this dissertation can be found in [126, 127].

1.2 Thesis Outline

In Chapter 2, an introduction to the basic concepts used in this dissertation is given. In this chapter, the ontology concept is introduced and its definition, types, usage areas, standards for ontology representation and tools developed for ontology management are provided. Then, an overview of video content modeling terminology is given. As being one of the major semantic contents, event is described and classified. Fuzzy logic and its terminology are described at the end of this chapter.

In Chapter 3, a literature survey on the related topics is presented. With the emerged need for semantic modeling, ontology-based semantic content analysis is described and examples for it are given. Next, event representation is presented. The last issue surveyed in terms of related work is event detection and recognition. How researchers approach this problem is described with example studies.

In Chapter 4, the proposed video semantic content model (VISCOM) and the enrichment of the model with rule usage is described in detail. After giving basic definitions, the utilization of the model to construct domain ontologies is described with an example ontology.

Chapter 5 explains the semantic content extraction system in details. Starting with the main architecture of the system, this chapter contains details about the object extraction process, spatial/temporal relation calculations, and ontology-based semantic content extraction process.

In Chapter 6, brief information about the standards, tools and libraries which are utilized during the implementation, the implementation details of the system, the experiments performed and the evaluations are given.

Finally, in Chapter 7, a short summary of the work is provided and the dissertation is concluded with future directions for research.

CHAPTER 2

BACKGROUND

In this chapter, a detailed overview of different material used extensively throughout the thesis is given. Ontology concept is presented in Section 2.1. An overview of video content modeling terminology is given in Section 2.2. General concepts on event and event representation are given in Section 2.3. Fuzzy logic and its terminology are described in Section 2.4.

2.1 Ontology

Ontology is a representation vocabulary specialized to some domain or subject matter. It is used to refer to a body of knowledge describing a commonsense knowledge domain using a representation vocabulary. In computer and information domain, ontology is defined as a formal representation of a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of the domain, and may be used to define the domain [123]. In [90], more ontology definitions are given followed by some briefly described ontology applications. The following items can be identified as essential aspects of ontology from ontology definitions:

- Ontology is used to describe a specific domain.
- Users of ontology agree on the meanings of the terms.
- There is a mechanism to organize the terms (relations).
- The terms and relations are clearly defined in that domain.

The aim of ontologies is to define primitives with their associated semantics for knowledge representation in a given context. Ontologies are typically formulated in languages that

allow abstraction away from data structures and implementation strategies. This allows the ontology designer to be able to state semantic constraints. For this reason, ontologies are said to be at the "semantic" level.

The terms knowledge base (KB) and ontology are somewhat interchangeable. The KB refers to a more concrete entity, a data structure which is supposed to serve as the actual instantiation of a given ontology. An ontology may be seen as an abstraction of a KB, as a scheme for carving up the world into concepts, relationships, and possibly rules about those concepts. To draw an analogy to traditional databases, an ontology is like a database schema. It defines what the data is and how it is related to everything else. The KB is like the database itself. And just as a database holds all the table meta-data and schema information, a KB contains the ontology as well as instance data. Ontologies are aimed at answering the question "What kind of objects exist in one or another domain of the real world and how are they interrelated?". Thus, an ontology describes the logical structure of a domain, its concepts and the relations between them [50].

2.1.1 Ontology Usage

Ontology is one of the most important concepts in knowledge representation. Moreover, it can be used to support a great variety of tasks in diverse research areas such as natural language processing, information retrieval, databases, semantic web, multimedia modeling, knowledge management, on line database integration, digital libraries, geographic information systems, visual information retrieval and multi agent systems. At present, there are many applications of ontology with commercial, industrial, academic or research focuses.

Ontology provides meta information which describes data semantics. Semantical relationships in ontologies are machine readable, in such a way that they enable making statements and asking queries about a subject domain. Ontologies enable knowledge level interoperation and support shared understanding, interoperability between tools, reusability and declarative specification. On the other hand, they are also used to build knowledge bases.

Ontologies can be used as a tool for knowledge acquisition or to classify the knowledge of an organization. They allow users to reuse knowledge in new systems. They can form a base to construct knowledge representation languages. Some applications use a domain ontology to integrate information resources and others allow each resource to use its own ontology. In information retrieval applications, ontologies not only serve to disambiguate user queries, but also elaborate taxonomies of terms in order to enhance the quality of retrieved results.

2.1.2 Ontology Types

Researchers categorize ontologies by taking several criteria into account such as the formality of the language and the level of dependence on a particular task. [49] considers the second one and identifies basic kinds of ontologies as generic (top-level) and domain dependent ontologies.

CYC [71], WordNet [81] and Sensus [111] are examples of generic ontologies. Their purpose is to make a general framework for all (or most) categories encountered by human existence. Generic ontologies are generally very large. Nevertheless, they are not very detailed and it is difficult to build them. They describe general concepts like space, time, matter, object, event or action, which do not depend on a particular problem or domain.

Different from generic ontologies, domain dependent ontologies are much smaller. This is because a domain dependent ontology provides concepts in a fine grain, while generic ontologies provide concepts in coarser grain. A domain ontology (or domain-specific ontology) models a specific domain, or a part of the world. It represents the particular meanings of terms as they apply to that domain. Some example domain ontologies are GFO [5], OpenCyc [9], SUMO [14] and DOLCE [4].

Ontologies are usually constructed by a domain expert, someone who has mastery over the specific content of a domain. During the construction of ontologies, the following points should be kept in mind [62]. Ontologies should be:

- open and dynamic: Ontologies should be readily capable of growth and modification.
- scalable and interoperable: An ontology should be easily scaled to a wider domain and adapt itself to new requirements.
- easily maintained: It should be easy to keep ontologies up-to-date. Ontologies should have a simple and clear structure.

Ontologies are categorized under three groups in terms of their way of generation:

Manual ontologies: Ontology engineering is done by domain experts. All concepts, properties and relations are defined by domain experts.

Semi-automatic ontologies: Initial definitions of the ontology are defined by domain experts. Ontology is upgraded by using the information/relevance feedback mechanism generated by the domain. The changes are controlled by the domain experts.

Automatic ontologies: Domain experts do not make any intervention at the beginning. Ontologies are generated by using defined facts and rules automatically.

Ontologies are categorized under two groups in terms of the uncertainty of concepts and relations defined in the ontology:

Crisp Ontologies: Concepts and relations are defined in a crisp manner (especially when exist/not exist or 0/1 is enough for representation).

Fuzzy Ontologies: Ontology contains fuzzy concepts or fuzzy relation definitions. In fact, a fuzzy ontology is a kind of crisp ontology that is expanded with fuzzy relations.

2.1.3 Ontology Components

Ontologies share many structural similarities, regardless of the language in which they are expressed. Typically, an ontology describes *concepts/classes*, concept *properties*, *relationships* between concepts and *individuals*. In this section, each of these components is discussed in turn.

Concept/Class

Concept is a class of items that together share essential properties which define the class. A concept represents a group of objects or beings sharing characteristics that enable them to be recognized as forming and belonging to this group. Concepts are typically represented with linguistic terms. In general, each concept in the ontology contains a label name which is unique to the ontology, and a list of synonyms.

Individual

Individuals are the basic, "ground level" components of an ontology. They may be instances or objects. An ontology does not have to include any individuals, but one of the general purposes of an ontology is to provide a means of classifying individuals.

Property/Attribute

Properties are used to describe concepts by assigning attributes, which have at least a name and a value.

Relationship/Object Property

Concepts are interconnected by means of relationships. *Is-A*, *Instance-Of* and *Part-Of* are three basic relationship types. *Is-A* is used to represent concept inclusion. A concept is said to be a specialization of another concept if it is a *kind of* or an *example of* it. *Instance-Of* is used to show membership. It denotes a single named existing entity but not a class. If a concept is a member of another concept then the interrelationship between them corresponds to an instance of. Lastly, *Part-Of* is used when a concept can be part of another concept.

The graphical representation of a simple document ontology is shown as an example in Figure 2.1. *Document*, *Book*, *Periodical*, *Edited Book*, *Journal* and *Magazine* are concepts of this ontology. *ISBN* within the *Book* concept and *Volume* and *Number* within the *Periodical* concept are properties of the related concepts. *Is-A* is the only relation type used in the document ontology given in Figure 2.1.

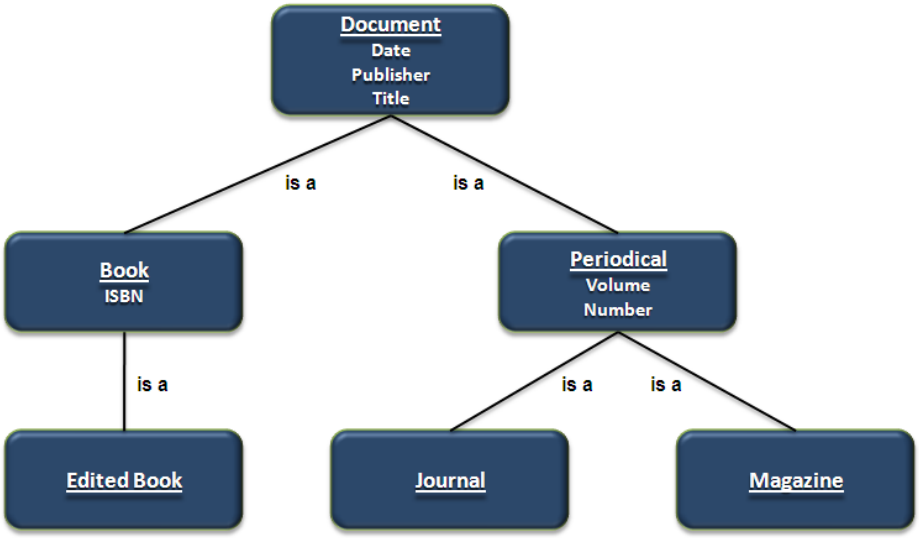


Figure 2.1: Ontology for Documents

2.1.4 Ontology Creation

The first step of ontology creation is finding the answer to the question, "What level of granularity of knowledge needs to be taken into account in the ontology?". The steps listed below are followed in order to create an ontology:

Step 1. Determine the domain and scope of the ontology.

Step 2. Consider reusing existing ontologies.

Step 3. Define the classes.

Step 4. Define the class hierarchy.

Step 5. Define the properties of classes-slots.

Step 6. Define the facets of the properties; cardinality, domain and range of a property.

Step 7. Create individuals.

2.1.5 Ontology Representation Languages and Tools

Ontologies are represented with languages that are classified as informal, semi-formal and formal languages according to their formality. Informal and semi-formal languages are developed by using natural languages which have none or limited structured form of natural languages. Both of them are readable but have limitations on automatic processing. Therefore, ontologies need formal languages for their specification in order to enable automatic processing.

Formal ontology representation languages can be classified as *structural* and *semantic web* languages. Structural languages which are mainly used in artificial intelligence domain have limitations on readability and usability. Some of the structural languages used in ontology representation are: ACL, LOOM, CyCL, F-Logic, RIF and conceptual graphs. On the other hand, the need to distribute, share and exchange information became crucial with semantic web that decreased the usage of structural languages. Several semantic web description languages have been defined to address this need. Some of the most popular ones are: XOL (XML based ontology-exchange language) [93], SHOE (Simple HTML Ontology Extensions) [56], Resource Description Framework (RDF) [11], Resource Description Framework Schema (RDFS) [30], Ontology Interchange and Inference (OIL) [58], Darpa Agent Markup Language (DAML) [3], DAML+OIL [34] and Web Ontology Language (OWL) [107].

Integrated tool suites provide a core set of ontology-related services for ontology representation, development and management. They have an extensible architecture and they are usually independent of ontology languages. Some of them are:

- **KAON1** is an open-source infrastructure for ontology creation and management, and provides a framework for building ontology-based applications [7].
- **Apollo CH** is a user-friendly knowledge modeling application [1].

- **OntoEdit** is an engineering environment for development and maintenance of ontologies using graphical means [110].
- **Ontolingua** is a distributed collaborative environment to browse, create, edit, modify and use ontologies [8].
- **Protege** is a free, open source ontology editor [10].
- **SymOntoX** is a web-based ontology management system conceived for the business domain [82].
- **WebODE** is an advanced ontological engineering workbench that provides various ontology related services, and covers and gives support to most of the activities involved in the ontology development process [117].
- **WebOnto** is a Java applet coupled with a customized web server allowing to browse and edit knowledge models over the web [15].
- **Chimaera** is a system for creating and maintaining distributed ontologies [2].

Libraries and reasoners are used to process through ontologies represented with formal ontology languages. Libraries are used to save, update, and query ontologies. Some of the libraries and tools in this context are: *SESAME* [13], *RDFSTORE* [12] and *JENA* [6]. Reasoners, on the other hand, are used to derive implicit knowledge through inferences. *FaCT*, *RACER*, *PELLET* are examples of ontology reasoners based on formal semantics.

2.1.6 OWL Related

In this dissertation, OWL, which is recommended by the W3C as the ontology language for semantic content applications, is chosen as the ontology representation language because of the availability of libraries and tools developed for OWL. OWL and OWL related concepts are presented briefly in this section.

OWL is a semantic markup language for publishing and sharing ontologies on the World Wide Web. It is developed as a vocabulary extension of RDF. It has more facilities for expressing meaning and semantics than XML, RDF and RDFS, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web. As seen in Figure 2.2 (appears in [39]), OWL uses all lower layers (XML, RDF, RDFS) as a data provider to represent semantics. These concepts are introduced below.

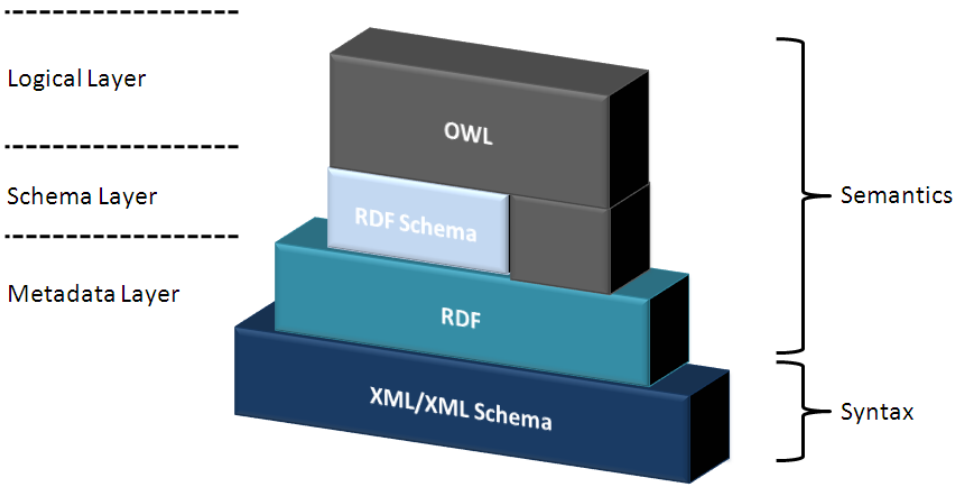


Figure 2.2: OWL in the Semantic Web Architecture

RDF

RDF is a framework for representing information in the Web. It defines a simple model for describing relationships among resources in terms of properties and values. RDF properties may be thought of as attributes of resources and in this sense correspond to traditional attribute-value pairs. RDF properties also represent relationships between resources. The underlying structure of any expression in RDF can be viewed as a directed labeled graph, which consists of nodes and labeled directed arcs that link pairs of nodes. The RDF graph is a set of triples as seen in Figure 2.3:

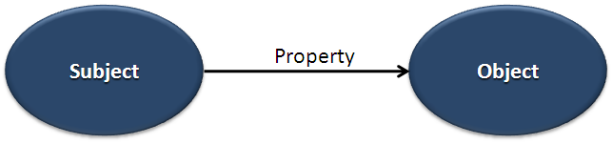


Figure 2.3: RDF Graph

Each property arc represents a statement of a relationship between the nodes that it links, having three parts:

1. a *property* that describes some relationship (also called a *predicate*),
2. a value that is the *subject* of the statement, and
3. a value that is the *object* of the statement.

For example, the notion, "The sky has the color blue", is represented in RDF with the triple: a *subject* denoting "the sky", a *predicate* denoting "has the color", and an *object* denoting "blue".

RDF Schema

The RDF data model itself, however, provides no mechanism for describing the properties, nor does it provide any mechanism for describing the relationships between the properties and other resources. That is the role of RDF Schema. The RDF vocabulary description language, RDF Schema [30], defines classes and properties that can be used to describe other classes and properties. It allows vocabulary designers to represent descriptions of classes and properties by describing ways in which combinations of classes, properties and values can be used together meaningfully.

Table 2.1 and Table 2.2 presents an overview of the basic vocabulary of RDF, drawing together vocabulary originally defined in the RDF model and syntax specification with classes and properties that originate with RDF Schema.

Table 2.1: RDF Classes

Class Name	Comment
rdfs:Resource	The class resource, everything.
rdfs:Literal	This represents the set of atomic values, e.g. textual strings.
rdfs:XMLLiteral	The class of XML literals.
rdfs:Class	The concept of Class.
rdf:Property	The concept of a property.
rdfs:Datatype	The class of datatypes.
rdf:Statement	The class of RDF statements.
rdf:Bag	An unordered collection.
rdf:Seq	An ordered collection.
rdf:Alt	A collection of alternatives.
rdfs:Container	This represents the set Containers.
rdfs:ContainerMembershipProperty	The container membership properties, rdf:1, rdf:2, ..., all of which are sub-properties of 'member'.
rdf:List	The class of RDF Lists.

Table 2.2: RDF Properties

Property name	Comment	Domain	Range
rdf:type	Indicates membership of a class.	rdfs:Resource	rdfs:Class
rdfs:subClassOf	Indicates membership of a class.	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	Indicates specialization of properties.	rdf:Property	rdf:Property
rdfs:domain	A domain class for a property type.	rdf:Property	rdfs:Class
rdfs:range	A range class for a property type.	rdf:Property	rdfs:Class
rdfs:label	Provides a human-readable version of a resource name.	rdfs:Resource	rdfs:Literal
rdfs:comment	Use this for descriptions.	rdfs:Resource	rdfs:Literal
rdfs:member	A member of a container.	rdfs:Container	not specified
rdf:first	The first item in an RDF list. Also often called the head.	rdf:List	not specified
rdf:rest	The rest of an RDF list after the first item. Also often called the tail.	rdf:List	rdf:List
rdfs:seeAlso	A resource that provides information about the subject resource.	rdfs:Resource	rdfs:Resource
rdfs:isDefinedBy	Indicates the namespace of a resource.	rdfs:Resource	rdfs:Resource
rdf:value	Identifies the principal value (usually a string) of a property when the property value is a structured resource.	rdfs:Resource	not specified
rdf:subject	The subject of an RDF statement.	rdf:Statement	rdfs:Resource
rdf:predicate	the predicate of an RDF statement.	rdf:Statement	rdf:Property
rdf:object	The object of an RDF statement.	rdf:Statement	not specified

DAML+OIL

DAML+OIL [34] is a semantic markup language for web resources. It builds on earlier W3C standards such as RDF and RDF Schema, and extends these languages with richer modeling primitives. DAML+OIL was built from the original DAML ontology language in an effort to combine many of the language components of OIL. DAML is used for ontology representation and OIL is used for inferencing. The language has a clean and well defined semantics. DAML+OIL triples can be represented in many different syntactic forms with any set of RDF triples.

OWL

The Semantic Web is build on XML's ability to define customized tagging schemes and RDF's flexible approach to representing data. The first requirement for the Semantic Web is an ontology language that can formally describe the meaning of the terminology used in web documents. If machines are expected to perform useful reasoning tasks on these documents, the language must go beyond the basic semantics of RDF Schema. OWL [107] is developed for this purpose. It has all features of RDFS and can use all classes as declared by RDF. The relation between OWL and RDF/RDFSchema is illustrated in Figure 2.4 (appears in [17]).

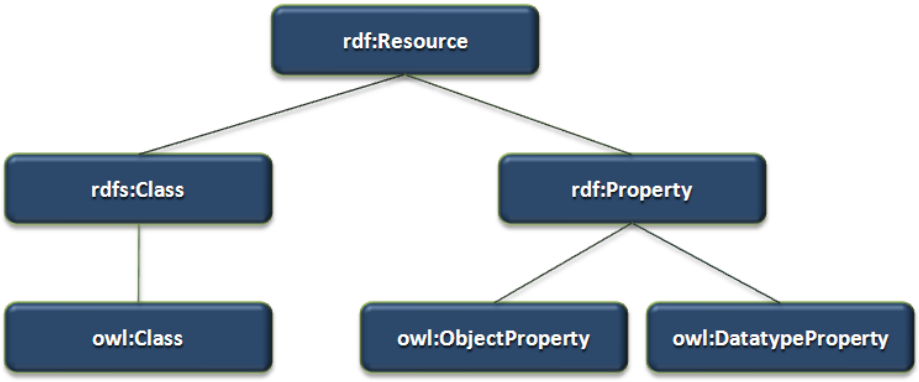


Figure 2.4: RDF-OWL Relation

OWL provides open world assumption and allows importing and mixing various ontologies. In order to provide such capabilities and, at the same time, to support calculations and reasoning, OWL introduces three expressive sublanguages for various purposes; OWL Lite, OWL DL and OWL Full. OWL Lite is intended mostly to support classification hierarchy and simple constraint features. It is decidable with desirable computational properties and it supports the classification hierarchy inference and simple constraint features. OWL DL includes all OWL language constructs that can be used only under certain restrictions. OWL DL is so named due to its correspondence with description logic, a field of research that has studied the logics that form the formal foundation of OWL. OWL Full contains all the OWL language constructs and provides free, unconstrained use of RDF constructs. OWL Full is typically useful for people who want to combine the expressivity of OWL with the flexibility and meta modeling features of RDF. OWL Full is an extension of OWL DL, which is an extension of OWL Lite, thus every OWL Lite ontology is OWL DL and OWL Full ontology and every OWL DL ontology is OWL Full ontology.

2.2 Video Content Analysis and Modeling

The need for analyzing and modeling the video content efficiently has become significantly important because of the growing consumer demand for visual information. The fundamental approach in all of the video content analysis and modeling studies is to index video data and make it a structured media.

MPEG-7 is issued as a standard for multimedia content description which captures video content description as a multimedia type. MPEG-7, formally named *"Multimedia Content Description Interface"*, is an ISO/IEC 1 standard that aims at describing the multimedia data content by attaching metadata to them. It specifies a standard set of description tools that consist of Descriptors (Ds) which represent features or attributes of multimedia data such as color, texture, textual annotation and media format, Description Schemes (DSs) which specify the structure and semantic of the relationships between their components, a Description Definition Language (DDL) which defines and extends the Ds and DSs.

However, the extraction and the usage of useful information in practical systems such as multimedia search engines are not considered in MPEG-7. The MPEG-7 formalism lacks the semantics and reasoning support in many ways. It does not convey a formal semantic since its DDL is XML schema based. XML is mainly used to provide a structure for documents and does not impose any common interpretation of the data contained in the document. Thus, XML schema helps to add structure to the MPEG-7 standard but it does not express the meaning of the structure. Additionally, inference mechanisms are not supported by MPEG-7.

Because MPEG-7 does not fulfill end-users' content modeling needs, content-based modeling of video data has received growing attention in the research community over the past decade. Early approaches proposed manual content description methodologies. Generating video content description manually is time consuming and more costly to the point that it is almost impossible. Moreover, when available, it is subjective, inaccurate, and incomplete. These drawbacks directed researchers to propose semi automatic or automatic models for video content modeling.

Video content is approached at raw data, low-level feature and semantic content levels. Raw video data consists of elementary video units together with some general video attributes such as format and frame rate. This kind of content does not mean anything for most of the users. Users are mostly interested in querying and retrieving the video in terms of what the video is about. Models based on low-level feature use automatically extracted features,

which represent the content of a video but they hardly provide semantics that describe high-level video concepts. Therefore, low-level features alone are not sufficient to fulfill the user's needs alone. Semantic content contains high-level concepts such as objects and events which are much more valuable for users.

"Video is a structured medium in which objects and events in time and space convey stories, so, a video must be viewed as a document, not as a non-structured sequence of frames" [109]. Therefore, the core research in content-based video retrieval is developing technologies to automatically parse video to identify meaningful composition structure and to represent and extract the semantic content from any video source.

2.2.1 Video Content Analysis

Content based video indexing and retrieval systems bear on modeling and extracting effective features describing visual media being indexed [22]. The underlying features can be low-level (primitive) or high-level (semantic), but the extraction and matching process are predominantly automatic.

Video content analysis involves four primary processes [38]: Feature extraction, structure analysis, abstraction, and indexing. The typical scheme of video-content analysis process is illustrated in Figure 2.5.

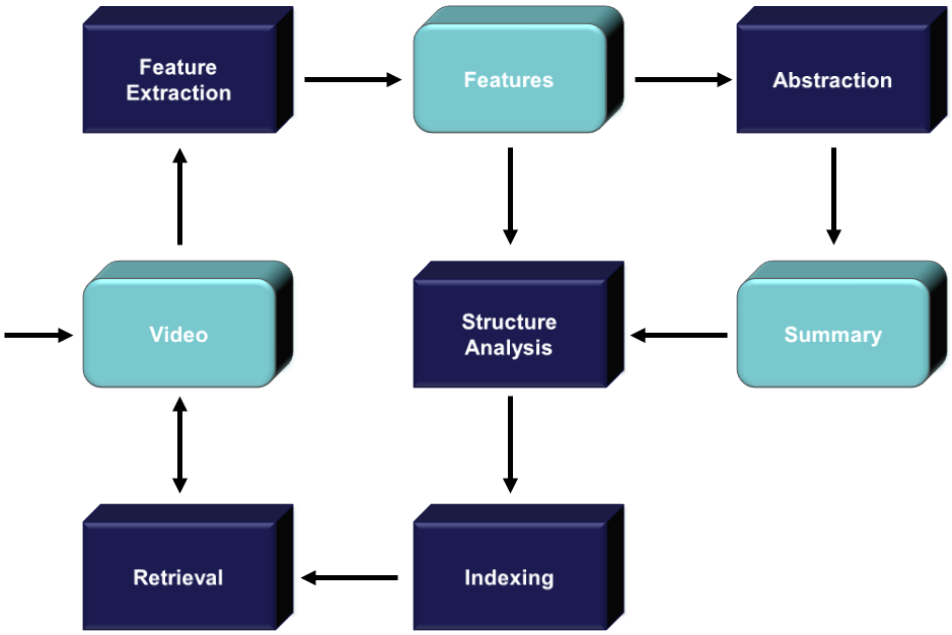


Figure 2.5: Video Content Analysis Processes

Feature Extraction

The effectiveness of an indexing scheme depends on the effectiveness of attributes in content representation. However, extractable video features (such as color, texture, shape and motion) can not be easily mapped into semantic concepts (objects and events).

Visual content is the major source of information in a video. In addition to visual content, there is some other valuable information carried in other media components, such as text, audio, and speech.

Structure Analysis

Video structure analysis is the next step in overall video-content analysis. This is the process of extracting temporal structural information of video sequences. This process allows video data to be organized according to its temporal structure and relations. Many effective and robust algorithms such as [74, 130, 133] for video parsing have been developed to segment a video into its temporal composition units. The top level of the units consists of sequences, which are composed of sets of scenes. Scenes are further partitioned into shots. Each shot contains a sequence of frames recorded contiguously and representing a continuous action in time or space.

Video Abstraction

Video abstraction is the process of creating a presentation of visual information about the structure of video, which should be much shorter than the original video. The abstraction process is similar to extraction of keywords or summaries in text document processing. Keyframe is the smallest abstract of shots. Keyframes are still images, extracted from original video data that best represent the content of shots in an abstract manner. They play an important role in the video abstraction process.

Indexing for Retrieval and Browsing

Based on the extracted structural and content attributes, video indices are built.

2.2.2 Video Content Modeling

Video data is continuous and unstructured. In order to analyze and understand its contents, video needs to be parsed into smaller chunks suitable for feature or conceptual abstraction analysis. Most of the existing video database systems start off with temporal segmentation of

video into a hierarchical model of frames, shots and scenes. The next logical step is compact representation and modeling of contents inside each shot using keyframes and objects.

Even if videos consist of sequences of images and, thus share all the attributes of image data, they have additional temporal and relational attributes. As a consequence, a video model should provide facilities to efficiently capture these additional attributes. Therefore, a temporal management of video information is required [51]. Most of the existing techniques detect shot boundary by extracting some form of features for each frame in the video sequence, then evaluating a similarity measure on features extracted from successive pairs of frames in the video sequence, and finally declaring shot boundary if the difference exceeds a fixed global threshold. For a review of major conventional shot boundary detection techniques, refer to [28].

Most of the existing systems represent the content by using one representative frame from each shot, called keyframe. One approach is to use the first frame of each shot as a keyframe. Although the approach is simple, each shot gets only one frame for its representation no matter how complex the shot contents are. Since video shots encapsulate spatial, temporal and high-level semantic information, more sources of information taken into account are likely to yield more accurate results.

Early approaches in video retrieval only added the functionality for segmentation and key-frame extraction to the existing image retrieval systems [95]. After key-frame extraction, similarity measurements based on features are applied. This is not satisfactory because video is a temporal media, so sequencing of individual frames creates new semantics that may not be present in any of the individual frames.

The naive user is interested in querying at the semantic level rather than having to use features to describe his concepts. At the physical level, video is a temporal sequence of pixel regions without direct relation to its semantics. Thus, modeling the semantic content is far more difficult than modeling the low-level visual content of a video. Because it is difficult to explore semantic content from the raw video data, semantic models at first used free text, attribute or keywords annotation to represent high-level concepts of the video data which results in many drawbacks. The major limitations of these approaches are that they are tedious, subjective and time consuming. Semantic models attempt to represent the meaning of video sequences, taking into account a number of aspects such as objects, spatial relationships between objects, events, and temporal relationships between events.

One important consideration is the importance of multi-modal nature of video data comprising of sequence of images along with associated audio and in many cases, textual

captions. Video bitstream that contains audio stream and possibly closed caption text along with sequence of images contains a wealth of rich information about objects and events being depicted. In [108], a multimodal multimedia event-based video indexing model, time interval multimedia event (TIME) framework, is mentioned as an approach for classification of semantic events in multimodal video documents. In [135], the proposed data model supports visual, auditory and textual modalities.

The next step towards future Content Based Video Information Retrieval (CBVIR) systems is the full induction of intelligence into systems as they need to be capable of communicating with the user, understanding the audio-visual content at a higher semantic level.

2.2.3 Sum Up

Content-based analysis of video requires methods which automatically segment video sequences and keyframes into image areas corresponding to salient objects, track these objects in time, and provides a flexible framework for object and event recognition, indexing, retrieval. Although multimedia standards, such as MPEG-4 and MPEG-7, provide the basic functionalities in order to manipulate and transmit objects and metadata, at the semantic level most video content is out of the scope of the standards.

Feature extraction, shot detection and object recognition are important phases in developing general purpose video content analysis studies. Significant results have been reported in the literature for the last two decades, with several successful prototypes such as [121, 60, 86]. However, the lack of precise models and formats for video semantic content representation makes the development of fully automatic video semantic content analysis and management a challenging task [19].

The main challenge, often referred as the semantic gap, is mapping high-level semantic concepts into low-level and spatio-temporal features that can be automatically extracted from video data. Many semantic content analysis systems have been presented recently such as [42, 72, 129]. These studies use MPEG motion vectors, Hidden Markov Models, occurrences of one or several slow motion shots, Finite State Machines or object trajectory information for semantic content analysis. In all of these systems low-level content analysis is not associated with any formal representation of the domain. At this point the use of domain knowledge becomes very important to enable higher level semantics to be integrated into the techniques that capture the semantics.

2.3 Event as a Semantic Content

Actually, events can be defined as long-term temporal objects, which usually extend over tens or hundreds of frames [132]. They occur within the video at defined content segments and represent the context for objects that are present within the video.

Most of the studies categorize events in terms of their complexity as; simple events, compound events (Multiple simple events taking place in time and space to achieve complex activities), and domain specific high level events (Interpretation of events in a particular context). In [99], Polana and Nelson separate the class of events into three groups; temporal textures which are of indefinite spatial and temporal extent (e.g., flowing water), activities which are temporally periodic but spatially restricted (e.g., a person walking), and motion events which are isolated events that do not repeat either in space or in time (e.g., smiling).

No matter what type the event is, event extraction is an essential task in semantic retrieval applications. In order to make successful event extraction, a good definition of what constitutes an event itself must be provided clearly. The goal of video event representation is to formalize the knowledge for the system to be able to detect video events. Therefore, an event representation language should be able to represent wide variety of events. It should to be formal and flexible to be able to add new event classes incrementally but still be natural for users. In addition, the representation should be useful to annotate instances in video and to recognize events automatically from video data [88].

Basically, there are two sources for content representation as *implicit* and *explicit* sources. The implicit sources are the low-level features and context. This kind of sources are much valuable for object representation and detection and can be used for image/video segmentation, interest point detection or finding similarities between whole images.

There is another possibility for semantic content representation which concerns an arrangement of a certain type (i.e. spatial, temporal) among semantic objects. Here, explicit knowledge is needed to form this arrangement. In fact, although linguistic terms are appropriate to distinguish event and object categories, they are inadequate when they describe specific patterns of events. The use of domain knowledge is probably the only way by which higher level semantics can be incorporated into techniques that capture the semantic concepts.

Several methodologies are proposed in order to describe specific patterns for event representation. Spatial relations between objects and object trajectories can give information about an event. Spatial relationships have a duration due to object motion and thus may

differ over time. Spatial relations include distance (far, near), geometrical, topological (left, right, top, bottom) relations. Events can have pre and post conditions that can be objects (information or relation between objects) or events. Temporal relations between events can be defined by using these conditions. Temporal relations (before, after, during, covers, overlaps, contains) between objects are treated through the events associated to them. The spatio-temporal relations (moves left, moves toward) characterize the evolution of spatial relations in time.

In [98], Allen's temporal relationships [16] are used to express parallelism and mutual exclusion between different subevents. Past-Now-Future networks (PNF-networks) are utilized to allow fast detection of actions and sub-actions.

In [104], declarative models are used to describe activities (states of the scene, events and scenarios). Activities are described by the conditions between the objects of the scene.

To increase the efficiency of processing temporal constraints, Vu et al. [120] suggest that, in a preprocessing step, scenario models can be decomposed into simpler scenario models containing at most two sub-scenarios. Then, the recognition of these simpler scenarios just tries to link two scenario instances instead of trying to link together a whole set of combinations of scenario models.

Petri nets have been suggested by Castel et al. [31] as an inference mechanism to represent the evolution of specific event types. A symbolic language is defined to capture the logical and algebraic conditions that are handled in a set of prototypes. [46] shows how to use Petri nets for event representation and recognition. The user defines objects and primitive events, and then expresses composite events using logical, temporal and spatial relations. Then the Petri net representations of these queries are automatically generated. Petri nets are provided with manually declared primitive events detected from video streams and are used as complex filters to recognize composite events.

Stochastic inference methods are also applied successfully to event representation and recognition from video data. Examples include Hidden Markov models [92], stochastic context free grammars [65] and Bayesian networks [83, 103].

2.4 Fuzzy Logic

The idea of fuzzy logic was first advanced by Dr. Lotfi Zadeh as a consequence of the development of the theory of fuzzy sets in [131]. Fuzzy logic is a superset of boolean logic that has been extended to deal with reasoning that is approximate rather than precise. It

aims to handle the concept of partial truth - truth values between "completely true" and "completely false". It includes 0 and 1 as extreme cases of truth but also includes the various states of truth in between so that, for example, the result of a comparison between two things could be not "tall" or "short" but ".68 of tallness."

"Temperature is high" and "person is tall"' are examples of fuzzy concepts. When is a person tall, at 180 cm , 190 cm or 200 cm? If we define the threshold of tallness at 190 cm, then the implication is that a person of 187 cm is not tall. When humans reason with terms such as "tall" they do not normally have a fixed threshold in mind, but a smooth fuzzy definition. Humans can reason very effectively with such fuzzy definitions, therefore, in order to capture human fuzzy reasoning fuzzy logic is needed.

Fuzzy logic differs from multi-valued logic by introducing concepts such as linguistic variables and hedges to capture human linguistic reasoning. A linguistic variable such as age may have a value such as young or its antonym old. However, the great utility of linguistic variables is that they can be modified via linguistic hedges applied to primary terms. This is achieved by associating linguistic hedges with certain functions.

Uncertain reasoning and fuzzy reasoning are confused most of the times. Probabilistic reasoning is concerned with the uncertain reasoning about well defined events or concepts. On the other hand, fuzzy logic is concerned with the reasoning about fuzzy events or concepts.

The behavior of a fuzzy system is completely deterministic. This makes it suitable to be utilized in many areas. Possible application areas for the use of fuzzy logic include fuzzy control, fuzzy pattern recognition, fuzzy arithmetic, fuzzy probability theory, fuzzy decision analysis, fuzzy databases, fuzzy expert systems and fuzzy computer software and hardware.

2.4.1 Fuzzy Set

In binary sets with binary logic, named also crisp logic, characteristic functions of sets only take values 1 (members) or 0 (non-members). In fuzzy set theory, characteristic functions are generalized to take value in the real unit interval [0, 1], or more generally, in some algebra or structure. Such generalized characteristic functions are more usually called *membership functions*, and the corresponding sets are called *fuzzy sets*.

Fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval [0, 1]. An element mapping to the value 0 means that the member is not included in the fuzzy set, 1 describes a fully included member. Values strictly between 0 and 1 characterize the fuzzy members.

2.4.2 Fuzzy Membership

The membership function of a fuzzy set is a generalization of the indicator function in classical sets. Membership functions on X represent fuzzy subsets of X. The membership function which represents a fuzzy set A is usually denoted by μ_A, where $\mu_A : X \rightarrow [0, 1]$. For an element x of X, the value $\mu_A(x)$ is called the membership degree/value or confidence factor of x in the fuzzy set A, where $A = \{(x, \mu_A(x))|x \in X\}$. The membership degree $\mu_A(x)$ quantifies the grade of membership of the element x to the fuzzy set A. The value 0 means that x is not a member of the fuzzy set; the value 1 means that x is fully a member of the fuzzy set. The values between 0 and 1 characterize fuzzy members, which belong to the fuzzy set only partially. Membership function graph of a fuzzy set is given in Figure 2.6.

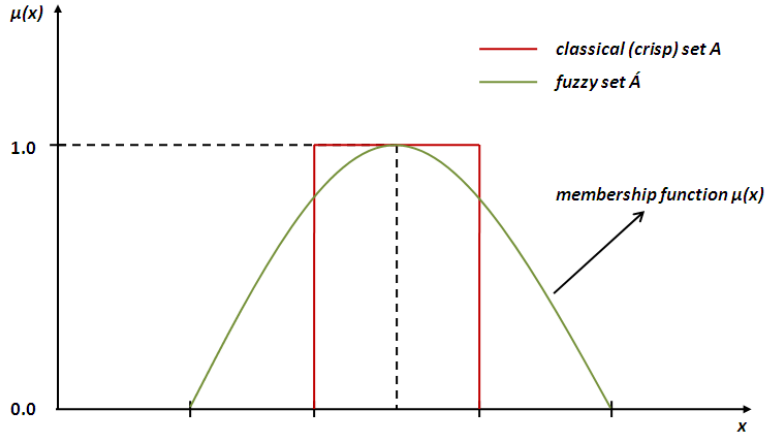


Figure 2.6: Membership Function Graph of a Fuzzy Set

As an example, the membership function graph of the term "tall" is represented in Figure 2.7. It shows the degree of membership with which a person belongs to the set "tall". Full membership of the class 'tall' is represented by a value of 1, while no membership is represented by a value of 0. At 150 cm and below, a person does not belong to the class "tall". At 210cm and above, a person fully belongs to the class "tall". Between 150cm and 210cm the membership increases linearly between 0 and 1. The degree of belonging to the set "tall" is called the confidence factor or the membership value.

Normally fuzzy concepts have a number of values to describe the various ranges of values of the objective term which they describe. For example, the fuzzy concept "tallness" may have the values "tall", "medium height" and "'short". Typically, the membership function graphs of these values are as shown in Figure 2.8:

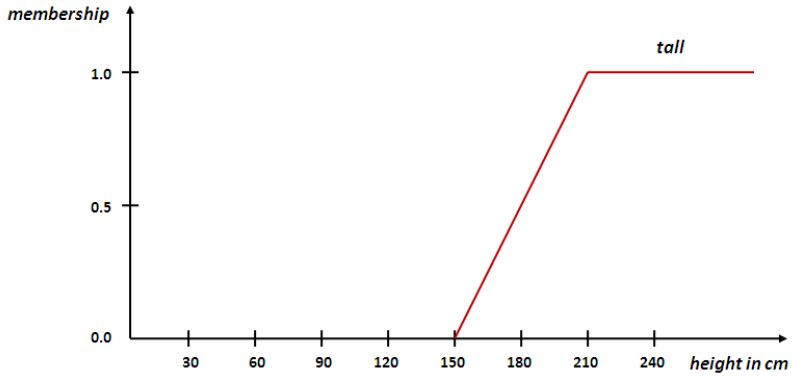


Figure 2.7: Membership Function Graph of "tall"

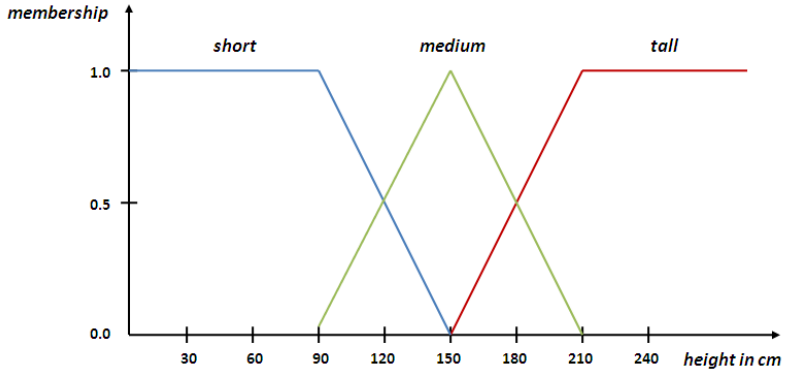


Figure 2.8: Membership Function Graphs of "tall", "medium" and "short"

The shape of the membership function curve can be non-linear. The most commonly used membership functions in practice are triangles, trapezoids, bell curves, Gaussian, and Sigmoid functions. Membership function graphs of triangle, trapezoid and gaussian functions are illustrated in Figure 2.9.

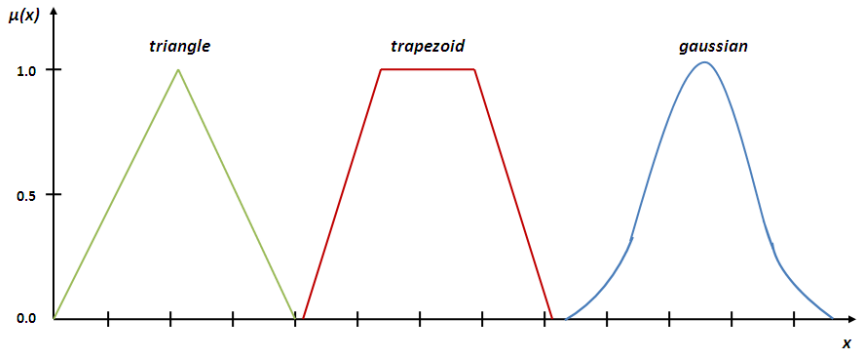


Figure 2.9: Membership Function Types

Basic operations on fuzzy sets include set union, set intersection and set complement. The following equations show membership function value calculation alternatives for the elements of the new set of a set union operation:

$$A \cup B \Leftrightarrow \{x, \mu_{A \cup B}(x) | (x \in A \vee x \in B) \wedge \mu_{A \cup B}(x) = max(\mu_A(x), \mu_B(x))\} \tag{2.1}$$

$$A \cup B \Leftrightarrow \{x, \mu_{A \cup B}(x) | (x \in A \vee x \in B) \wedge \mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x).\mu_B(x)\} \tag{2.2}$$

$$A \cup B \Leftrightarrow \{x, \mu_{A \cup B}(x) | (x \in A \vee x \in B) \wedge \mu_{A \cup B}(x) = min(1, \mu_A(x) + \mu_B(x))\} \tag{2.3}$$

The following equations show membership function value calculation alternatives for the elements of the new set of a set intersection operation:

$$A \cap B \Leftrightarrow \{x, \mu_{A \cap B}(x) | (x \in A \wedge x \in B) \wedge \mu_{A \cap B}(x) = min(\mu_A(x), \mu_B(x))\} \tag{2.4}$$

$$A \cap B \Leftrightarrow \{x, \mu_{A \cap B}(x) | (x \in A \wedge x \in B) \wedge \mu_{A \cap B}(x) = \mu_A(x).\mu_B(x)\} \tag{2.5}$$

$$A \cap B \Leftrightarrow \{x, \mu_{A \cap B}(x) | (x \in A \wedge x \in B) \wedge \mu_{A \cap B}(x) = max(0, \mu_A(x) + \mu_B(x) - 1)\} \tag{2.6}$$

The following equation show the membership function value for the elements of the new set of a set complement operation:

$$\neg A = \{x, \mu_{\neg A}(x) | (\mu_{\neg A}(x) = (1 - \mu_A(x))\} \tag{2.7}$$

CHAPTER 3

RELATED WORK

In this chapter, recent studies that are most relevant to this dissertation are reviewed under spatial and temporal relation usage in event representation, ontology-based semantic modeling and event detection/recognition categories. Based on this survey, it is deduced that semantic content representation in an ontology enriched with qualitative attributes of semantic objects, spatial relations, temporal relations, and rule definitions can be used for automatic semantic content extraction.

3.1 Spatial/Temporal Relation Usage in Event Representation

Spatial relations between objects and temporal relations between semantically meaningful intervals are utilized by many researchers to define events. In this section, brief information about spatial and/or temporal relation usage in event representation is given.

In [109], a domain knowledge ontology for video event description is given. Semantic concepts in the context of the video events are described and enriched with attributes of the semantic objects and low level features (pixel color and motion vectors). A set of spatial (approach, touch and disjoint) and temporal (before, meet, after, starts and completes) relation types are used in event representation.

[29] proposes a model to represent events for automatic video interpretation. An ontology structure is built to design concepts relative to video events. Non-temporal constraints (logical and spatial) to specify physical objects involved in a concept and temporal constraints including Allen's interval algebra operators to describe relations (e.g. temporal order, duration) between sub-concepts are used.

[119] represents a scenario model by specifying the objects involved in the scenario, the sub-scenarios and the constraints between the sub-scenarios. Spatio-temporal and logical

constraints are used. In this study, the authors propose a recognition algorithm for processing temporal constraints and combining several actors defined within the scenario.

In [100], a top level ontology which provides a framework for describing the semantic features in video is presented. First, key components of semantic descriptions like objects and events and how domain specific ontologies can be developed from these key components are identified. Second, a set of predicates for composing events and describing various spatio-temporal relationships between events are presented. Third, a scheme for reasoning with the developed ontologies to infer complex events from simple events is developed.

[27, 88, 89] define an event ontology that allows natural representation of complex spatio-temporal events. At the lowest level, primitive events are defined directly from object properties. An Event Recognition Language (ERL) that allows users to define the events without interacting with the low level processing is defined. The proposed video version, VERL, is intended to be a language for representing events for the purpose of designing an ontology of the domain and for manually annotating data with the categories in that ontology. As mentioned by the authors, the framework needs to be exercised on much more complex events from different domains.

[114] presents an approach for automatic scene interpretation of airport aprons based on a multi-camera video surveillance system. The video event model of this study is composed of a set of object variables, a set of temporal variables, a set of forbidden variables corresponding to the components that are not allowed to occur during the detection of events, a set of constraints (symbolic, logical, spatial and temporal constraints) and a set of decisions corresponding to the tasks predefined by experts. It categorizes video events into four types: primitive states, composite states, primitive events and composite events. A state describes a situation characterizing one or several physical objects defined at a time or a stable situation defined over a time interval. While a primitive state corresponds to a visual property directly computed, a composite state corresponds to a combination of primitive states. An event, on the other hand, is an activity containing at least a change of state values between two consecutive times. A primitive event is a change of primitive state values and a composite event is a combination of states and/or events. Events are represented only with spatial changes of objects.

3.2 Ontology-Based Semantic Video Modeling

"The research goals in semantic modeling are not unique but they are mostly a function of the granularity of the semantics in question" [97]. The goal could be the extraction of a single or multiple semantics of the entire video. In the latter case, the semantics could be generic or specific. Depending on the goal, the task of semantics extraction can be considered as a classification, recognition or understanding task that all share in common the effort for solving the semantic gap.

Ontologies are effectively used to perform semantic content representation and extraction of video. As described before, ontologies are formal, explicit specifications of a domain knowledge: they consist of concepts, concept properties, and relationships between concepts and are typically represented using linguistic terms. Semantic concepts within the context of the examined domain can be defined in an ontology, and enriched with attributes of the semantic objects, events, concepts, numerical data and low-level features.

Ontology makes the video systems user-centered and enables the experts to fully understand the terms. Moreover, ontology is useful to evaluate the video systems and to understand exactly what types of events a particular video system can recognize. Ontology is also useful for video application developers to share and reuse models dedicated to the recognition of specific events [29]. Furthermore, ontology usage for multimedia information processing offers several advantages [48]:

- Ontology provides a source of precisely defined terms that is used
  - to index the metadata describing the semantic content,
  - to express the queries,
  - to describe the content of each source.
- An ontology-based approach allows more precise queries on metadata.
- The inferences that are drawn from the ontology help to derive information that was not explicitly stated in the metadata.

Ontology-based semantic content representation approaches differ from each other in terms of what they use to represent the semantic content. The first group uses multimedia content and/or descriptor ontologies with a domain ontology. The second group uses spatial and/or temporal relations for semantic content description. Third group builds ontologies

using low-level features (visual, audio, speech or textual). Some example studies from each group are given in the following sections.

3.2.1 Domain and Multimedia Content Ontologies

In [97], a multimedia semantic model links domain specific ontologies, in which concepts are represented by domain-specific terms, with multimedia content ontologies that represent the content structure in multimedia documents. Additionally, it describes characteristics of multimedia objects in terms of low-level features and structural descriptions. The aim of this project, named as **BOEMIE**, is to provide information about mid-level concept instances in video data. BOEMIE approach view is given in Figure 3.1 (appears in [97]).

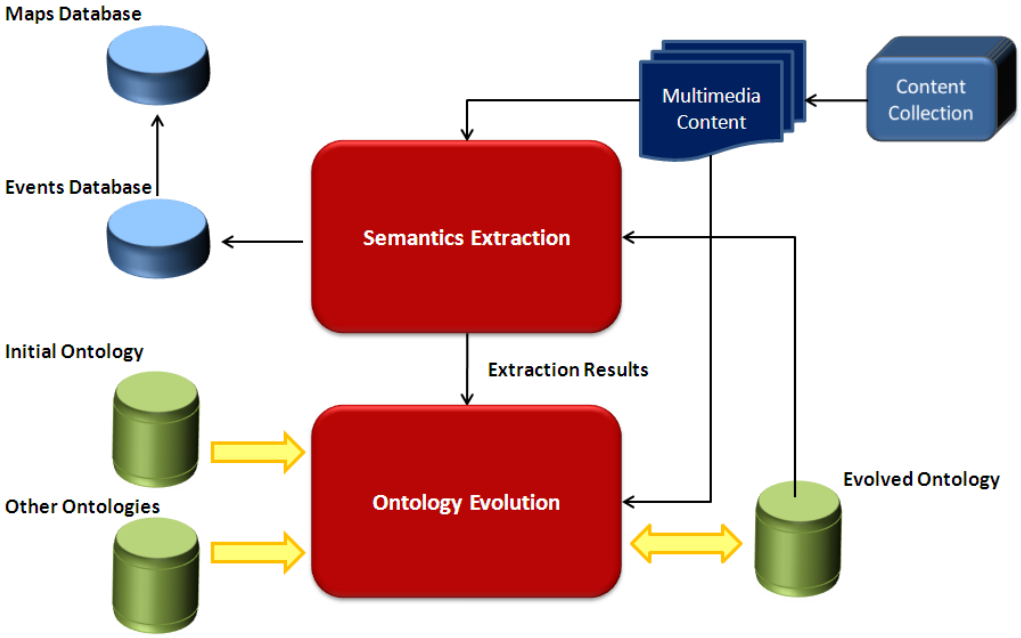


Figure 3.1: BOEMIE Approach View

[21] presents an annotation approach that uses a domain specific ontology together with a domain independent video ontology that encodes the structure and attributes of video data. The two ontologies are integrated with a domain specific semantic linkage. The integrated ontology for video annotation, named as IOVA, is represented in OWL with a description logic based ontology language. In Figure 3.2 (appears in [21]), *VideoClip* class of IOVA is given.

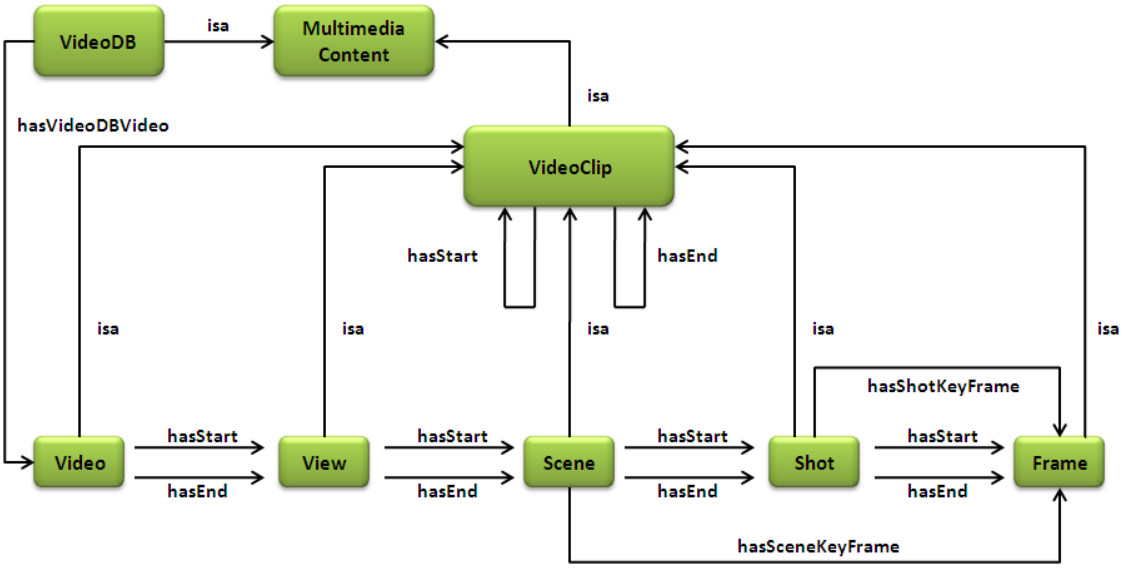


Figure 3.2: VideoClip Ontology

In [53], MPEG-7 standard is extended by an ontology for domain knowledge representation allowing some reasoning mechanism over MPEG-7 description. In the proposed framework, both the user queries and the database descriptions are specified as ordered labeled trees. A tree embedding algorithm is used as a multimedia data retrieval tool.

In [118], an approach that uses multimedia ontologies based on the MPEG-7 standard and domain-specific vocabularies is presented. MPEG-7 is used to model structural and low-level aspects of multimedia documents. High-level semantics are modeled using a domain-specific ontology designed for soccer games.

A domain specific linguistic ontology with multimedia lexicons is presented in [101]. Domain ontologies and reasoning algorithms are utilized to automatically create a semantic annotation of soccer video sources.

In [26], a visual descriptor ontology based on MPEG-7 visual descriptors and a multimedia structure ontology based on MPEG-7, are used together with a domain ontology in order to support content annotation.

[115] proposes a framework that supports ontology-based semantic indexing and retrieval of audiovisual content for metadata descriptions. This work provides a methodology to enhance the retrieval effectiveness of audiovisual content. In this framework, domain-specific ontologies guide the definition of both application specific metadata and instance description metadata that describe the contents of audiovisual segments.

3.2.2 Ontologies using Spatial/Temporal Relations

[19] presents a video semantic content analysis framework based on ontology. Domain ontology is used to define high level semantic concepts and their relations. Low-level features and video content analysis algorithms are integrated into the ontology to enrich video semantic analysis. OWL is used as the ontology description language. Rules are defined to describe how features and algorithms for video analysis should be applied. Temporal Description Logic is used to describe the semantic events, and a reasoning algorithm is proposed for event detection. Event extraction is done manually and only temporal relations are used for event description. The proposed framework is given in Figure 3.3 (appears in [19]).

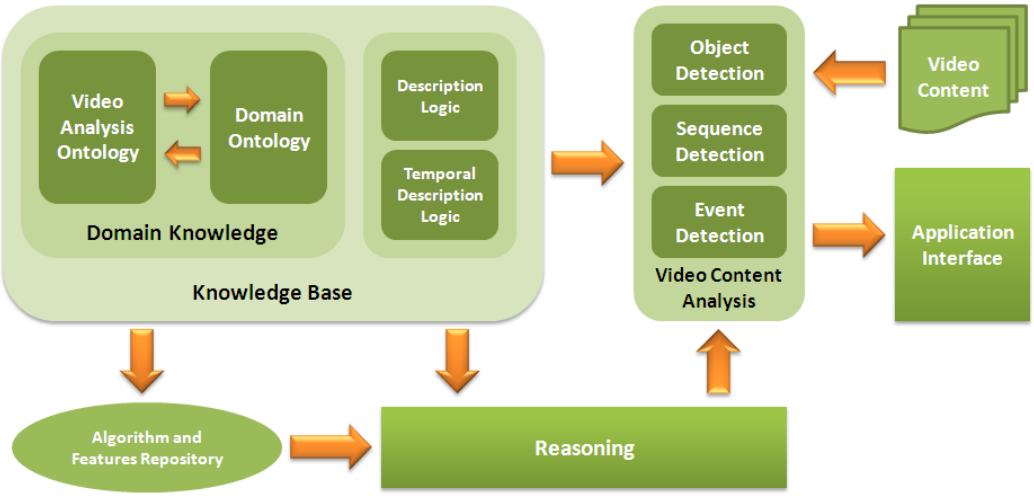


Figure 3.3: Ontology-based Semantic Content Analysis Framework

[100] proposes an ontology model using spatio-temporal relations in order to make complex event extraction. A top level ontology is presented that provides a framework to describe the semantic features in video. A set of predicates is presented for composing events and describing various spatio-temporal relationships between events. However, the semantic content annotation process is manual.

[47], [54] and [113] present a fuzzy spatio-temporal OWL extension approach. Ontology is used for recognizing concepts relevant to a video scene by making inferences from other ontological concept definitions and relations. Bayesian networks are used as the reasoning mechanism.

[79] and [80] present an ontology-based study that is enriched with relevance feedback mechanism. MPEG-7 compliant low-level descriptors describing the color, shape, position,

and motion of the resulting spatio-temporal objects are extracted and automatically mapped to appropriate intermediate-level descriptors forming a simple object ontology. Spatial and temporal relations between objects in multiple-keyword queries can also be expressed with the shot ontology. Concepts are expressed as keywords and are mapped in an object ontology, a shot ontology and a semantic ontology. The shot ontology proposed by this study is given in Figure 3.4 (appears in [79]).

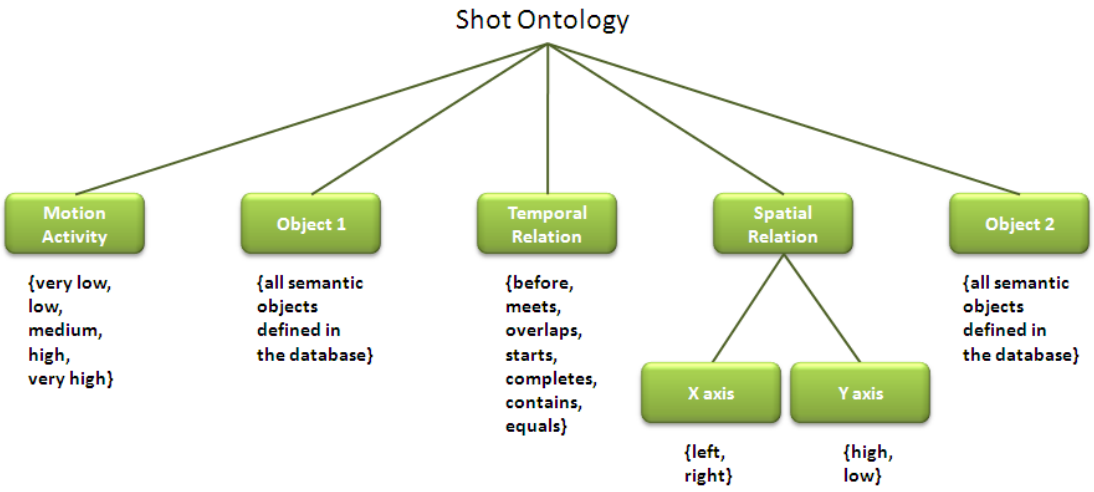


Figure 3.4: Shot Ontology

3.2.3 Ontologies using Low Level Features

[18] presents a solution for the definition and implementation of multimedia ontologies for the soccer video domain. Each linguistic concept in the domain ontology is associated with a corresponding visual concept. Visual concepts are clustered according to the similarity of their spatio-temporal patterns. Additionally, the video structure ontology describes the component elements of a video such as clips, shots, frames. Video structure, visual and linguistic ontology is utilized for semantic content extraction. Object relations are not used during the extraction process.

In [35], an ontology infrastructure to annotate video content is presented. Visual prototype instances are manually linked to the domain ontology to detect semantic video objects in the improved version [36] of [35]. Semantic concepts are defined in a RDF(S) ontology together with attributes (e.g. color homogeneity), low-level features, object spatial relations

and multimedia processing methods (e.g. color clustering). Rules in F-logic are used to make detection on video objects.

The approach proposed in [55] utilizes a set of intermediate textual descriptors in the form of a large taxonomic classification scheme that are applied to visual scenes. Thus, general-purpose semantic content annotation and retrieval is enabled through these descriptors. The semantic concepts are then used for both manual and automatic indexing of video footage.

Marco Bertini et al., in [23, 24], propose a system named as Multimedia Ontology Manager (MOM), that supports dynamic creation and update of multimedia ontologies and provides facilities to automatically perform annotations. The multimedia ontology is created by linking video sequences as instances of concepts in the linguistic ontology, and performing an unsupervised Fuzzy C-Means clustering of instance clips. Annotation of clips is performed by checking their similarity to the visual concepts in the ontology.

In [63, 68], an object ontology, coupled with a relevance feedback mechanism, is introduced to facilitate the mapping of low-level to high-level features. The study is done as a part of *AceMedia* project. In aceMedia, ontologies are extended and enriched to include low level audio visual features, descriptors and behavioral models in order to support automatic content annotation.

An extended linguistic ontology with a multimedia ontology is presented in [66] to support video understanding. First, multimedia ontologies are constructed manually. Second, each video is pre-processed by performing scene cut detection, automatic speech recognition (ASR), and metadata extraction. In addition, videos are automatically indexed based on visual content by extracting syntactic (e.g., color, texture, etc.) and semantic features (e.g., face, landscape, etc.).

3.3 Event Detection and Recognition

Automatic event detection and recognition from videos is gaining attention in the computer vision research community. The analysis of events is important in a variety of applications including surveillance, vision-based human-computer interaction and content-based retrieval. The type of events to be recognized can vary from a small-scale action to a large-scale activity. Addressing all the issues in event detection is thus enormously challenging.

The task of event recognition is to bridge the gap between numerical pixel level data and a high-level abstract activity description. There are several challenges that need to be addressed in the event detection process. Some of them are as follows:

- Motion detection and object tracking from real video data.
- The interpretation of low-level features.
- There is a spatio-temporal variation in the execution style of the same activity by different actors, leading to a variety of temporal durations.
- Similar motion patterns may be caused by different activities.

First, a good definition of what constitutes an event is lacking. Second, detection and recognition of objects, actions and their evolving interrelationships are required to understand events. Moreover, events are often multimodal, requiring information available in multiple media sources.

In [52], event detection approaches are arranged into three categories. First, approaches which utilize pre-defined event models either manually encode the event models or provide a grammar or rules to detect events in videos. Force dynamics, stochastic context free grammars, state machines, and PNF Networks are used by the approaches in this category. Second, approaches that learn the event models are generally used to make activity recognition. They either model single object activities or require prior knowledge about the number of objects involved in the events. There is no straight-forward method of expanding the domain to other events after training. Hidden Markov Models (HMMs), Coupled HMMs and Dynamic Bayesian Networks (DBNs) are used by the approaches in this category. Third, approaches do not model the events, but utilize clustering methods for event detection. These methods assume maximum length of an event is restricted to single object non-interactive event detection.

Most of the current event recognition approaches are composed of defining models for specific event types that suit the goal in a particular domain and develop procedural recognition methods.

In [37], simple periodic events are recognized by constructing the dynamic models of human movements. Unfortunately, the proposed model is highly dependent on the robustness of the tracking.

Bayesian networks have been used to recognize simple events from the visual evidence gathered during one video frame. [25, 64] are examples of this type. The use of Bayesian networks differs in the way how they are applied (e.g., what data is used as input, how this data is computed and the structure of the networks, etc.). One of the limitations of using Bayesian networks is that they are not suitable for encoding the complex events.

Hidden Markov Model formalism, as an alternative to Bayesian networks, has been extensively applied to event recognition. [92, 102, 124] are examples of this type. Even though HMMs are robust against the variation of the temporal segmentations of events, the structures and probability distributions need to be learned accurately using an iterative method. Because of this, for complex events, the parameter space may become prohibitively large.

Because it is difficult to track multiple objects in a scene and to maintain the parameters of the temporal granularity of the event models, there is only a limited amount of research on multi-agent events. [64, 84] are examples of this type.

In literature, a variety of approaches have been proposed for the detection and recognition of events in video sequences. Below, there are some examples that use different methodologies such as object detection and tracking, multimodality and spatio-temporal derivatives.

3.3.1 Detection and Recognition of Regions/Objects

Object detection and recognition algorithms can be classified into four broad categories as feature-based, model-based, motion-based, and data association algorithms [129]. In feature-based algorithms, features of objects are used to discriminate target objects from other objects within a frame. Model-based algorithms use not only features but also high-level semantic representation and domain knowledge to discriminate target objects from other objects. Motion-based algorithms, on the other hand, rely on the methods for extracting and interpreting the motion consistency over time to segment the moving object. And finally, data association algorithms are designed to solve the data association problem, which is a problem of finding the correct correspondence between the measurements for the objects and the known tracks.

In most natural scenes, there is a significant number of moving objects and it is the analysis of their trajectories and interaction with the features of the scene which helps in classification and recognition of interesting events. Event definitions are made only through predefined object motions and their temporal behavior.

[78] presents a system which takes a video stream obtained from an airborne moving platform and produces an analysis of the behavior of the moving objects in the scene. The system relies on two modular blocks. The first one detects and tracks moving regions in the sequence. The second module takes these trajectories as input, together with user-provided information to instantiate likely scenarios.

In [57], events are modeled from shape and trajectory features using a hierarchical activity representation extended from [78], where events are organized into several layers of

abstraction. The event recognition methods described in [78] are based on a heuristic method and could not handle multiple-actor events. In this study, an event is considered to be composed of action threads, each thread being executed by a single actor. The bottleneck of this study is its dependence on motion detection.

In [72], motion information associated to an MPEG-2 bitstream is considered. The problem is addressed by identifying a correlation between semantic events and the low-level motion indices associated to video sequences.

3.3.2 Fusion of Multimodal Information

There has been a significant amount of work toward the fusion of multimodal information (e.g., color, motion, acoustic, speech, and text) for event recognition in recent years. Many approaches such as [87, 91, 112] rely on contextual knowledge and are limited to specific domains (e.g., offices, classrooms, and TV programs).

[32, 122] propose a neural network based framework for semantic event detection in soccer videos. The framework provides a solution for soccer goal event detection. A learning-based event detection framework is proposed in this study, which incorporates both the strength of multimodal analysis and the ability of neural network ensembles. In addition, a bootstrapped sampling approach is adopted for rare event detection.

In [134], a multi-modal framework for semantic event extraction from basketball games based on webcasting text and broadcast video is presented. Text analysis for event detection and semantics extraction, video analysis for event structure modeling and event moment detection, and text/video alignment for event boundary detection in the video are main areas focused by this framework. An unsupervised cluster based method is proposed instead of pre-defined keywords to automatically detect events from web-casting text. In addition, a statistical approach is proposed instead of a finite state machine to detect event boundary in the video.

In [105], an audio-visual feature based framework for event detection in broadcast sport videos is proposed. Features indicating significant events are selected and robust detectors are built.

In [20], a semantic event detection approach based on Finite State Machines to automatically detect significant events within soccer videos is proposed.

In [42], a framework for automatic, real-time soccer video analysis and summarization by using cinematic and object features is proposed. A flowchart of the proposed framework is given in Figure 3.5.

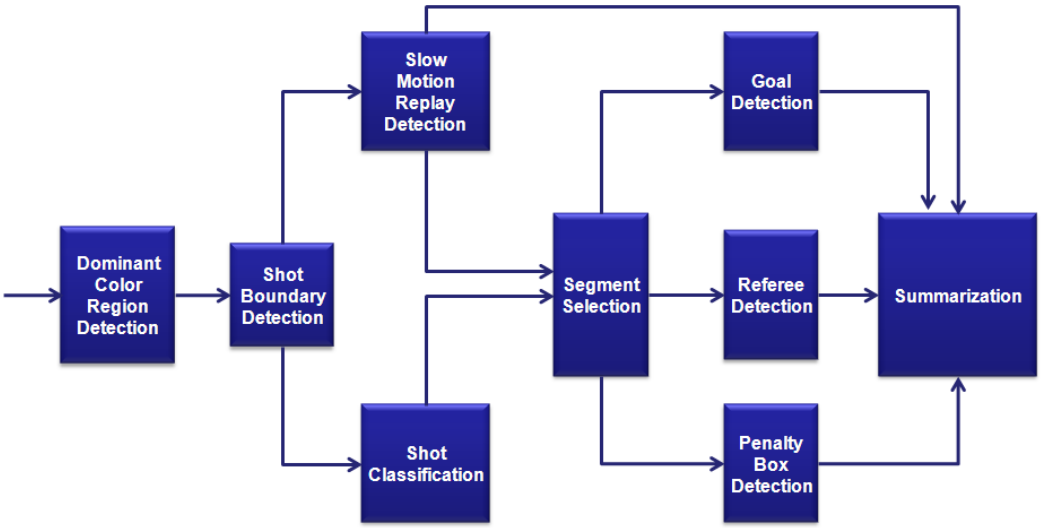


Figure 3.5: Real-time Soccer Video Analysis and Summarization

3.3.3 Spatio-Temporal Relation Usage

[52] proposes a method to detect events involving multiple objects and to learn event structure in terms of temporally related chain of sub-events. The proposed method has two significant contributions over existing frameworks. First, a video event graph is proposed to learn the event structure from training videos. The video event graph is composed of temporally correlated sub-events, which is used to automatically encode the event correlation graph. Second, the problem of event detection in videos is posed as clustering the maximally correlated sub-events where normalized cuts are used to determine these clusters. The principal assumption made in this work is that the events are composed of highly correlated chain of sub-events.

The video model given in [95] integrates feature-based and annotation-based approaches, in such a way that annotations are extracted automatically from visual features. The underlying video data model provides a framework for automatic mapping from features to semantic concepts by integrating audio and video primitives. This study addresses content-based video retrieval with an emphasis on spatio-temporal modeling and querying of events.

CHAPTER 4

ONTOLOGY-BASED VIDEO SEMANTIC CONTENT MODEL

In order to address the semantic content modeling need of the dissertation, a domain independent ontology-based semantic content model which uses objects and spatial/temporal relations in event and concept definitions is developed. In this chapter, the proposed video semantic content model and the enrichment of the model with rule usage is described in detail. Organization of the chapter is as follows: First, an overview, main components and relation types between the components of the model are introduced. After giving basic definitions, the utilization of the model to construct domain ontologies is described with an example ontology. At the end of the chapter, we introduce how we extend the representational and reasoning capabilities of the model with rule definitions.

4.1 Overview of the Model

As described in Section 2.1 and Section 3.2, ontology has many advantages and capabilities for content modeling which attracted many researchers' attention to ontology usage for semantic content representation in videos. However, a great majority of the ontology-based video content modeling studies propose domain specific ontology models which contain a limited set of semantic components specific to a domain. On the other side, generic ontology models generally propose solutions for multimedia structure representation. Thus, in this dissertation, a domain independent video content model which is utilized to model the semantic content in videos is proposed.

Objects, events, concepts, spatial and temporal relations are components of this generic ontology-based model. Similar generic models such as [79, 100, 109] which use objects and spatial and temporal relations for semantic content modeling neither use ontology in content

representation nor support automatic content extraction. To the best of our knowledge, there is no domain independent video semantic content model which uses both spatial and temporal relations between objects and which supports automatic semantic content extraction as this model does.

The domain independent **VI**deo **S**emantic **CO**ntent **M**odel of this dissertation is named **VISCOM**. Domain experts define domain specific components such as objects, events and concepts as individuals of VISCOM classes to generate domain ontologies where the granularity of the semantic contents can be determined at any level by the domain expert. Domain ontologies are utilized in the automatic semantic event and concept extraction process.

The starting point is identifying what video contains and which components can be used to model the video content. Because raw video consists of elementary video units together with some general video attributes, these components should be investigated through to find how they can be utilized to extract semantically meaningful units. Keyframes are the elementary video units which are still images, extracted from original video data that best represent the content of shots in an abstract manner. Name, domain, frame rate, length, format are examples of general video attributes which form the metadata of video. An instance of video, V_i, is represented as: $V_i = \left\langle V_{i_{metadata}}, V_{i_{keyframe}} \right\rangle$, where $V_{i_{keyframe}}$ is the set of keyframes of V_i and $V_{i_{metadata}} = \left\langle V_{i_{name}}, V_{i_{domain}}, V_{i_{framerate}}, V_{i_{length}}, V_{i_{format}} \right\rangle$. $V_{i_{domain}}$ is an attribute of video metadata that represents the domain of the video instance, where $V_{i_{domain}} \in D$. $D = \{D_0, \ldots, D_n\}$ is the set of all possible domains.

Each $D_x \in D$ contains semantically meaningful content common for D_x, which can be represented with an ontology ONT_x, where $ONT_x \in ONT$. $ONT = \{ONT_0, \ldots, ONT_n\}$ is the set of all possible domain ontologies.

ONT_x is a domain ontology and represented as $ONT_x = \langle MetaModel, CI_x \rangle$, where $MetaModel$ is the model having domain independent content definitions in terms of types and relations. In our case, these definitions are semantic contents. CI_x is the set of domain specific $MetaModel$ individuals for domain D_x.

The model in this study is a $MetaModel$ and represented with $VISCOM = \langle VC, DII \rangle$. VC is the set of $VISCOM$ classes and DII is the set of domain independent $VISCOM$ class individuals. Each VC_x in VC is represented as $VC_x = \langle VC_{x_{name}}, VC_{x_{prop}} \rangle$, where $VC_{x_{name}}$ is the name of the class and $VC_{x_{prop}}$ is the set of relations and properties of class VC_x. $VISCOM$ has a number of classes representing semantically meaningful components of video, where $VC_{x_{name}} = \{Component, Object, Event, Concept, Similarity, ...\}$.

Domain independent $VISCOM$ class individuals (DIIs) are grouped under four relation

types. $DII = MRI \cup TRI \cup OCTI \cup SRI$, where $MRI = \{down, up, right, left\}$ is the set of movement relation types, $TRI = \{before, meets, starts, finishes, overlaps, equal, during\}$ is the set of temporal relation types, $OCTI = \{composedOf, isA, partOf, substanceOf\}$ is the set of relation types used to define concept inclusion, membership and structural object relations, and $SRI = DSRI \cup PSRI \cup TSRI$ is the set of spatial relation types, where $TSRI = \{inside, partiallyInside, disjoint, touch\}$ is the set of topological spatial relation types, $PSRI = \{right, left, above, below\}$ is the set of positional spatial relation types, and $DSRI = \{far, near\}$ is the set of distance spatial relation types.

Each domain ontology is also enriched with rule definitions to be able to define some complex situations more effectively. $R_i \in R$ represents rule definitions for domain $D_i \in D$, where $R = \{R_0, \ldots, R_n\}$ represents all possible rule sets for all domains. Each rule is composed of two parts as $R_{i_x} = \langle body, head \rangle$, where *body* part contains any number of domain class or property individuals and *head* part contains only one individual with a value, μ, representing the certainty of the definition given in the *body* part to represent the definition in the *head* part, where $0 \leq \mu \leq 1$.

The automatic semantic content extraction framework takes V_i, ONT_j and R_j, where V_i is a video instance, ONT_j is the domain ontology for domain D_j which V_i belongs to and R_j is the set of rules for domain D_j. The output of the extraction process is a set of semantic contents, named VSC_i, and represented as $VSC_i = \langle V_i, OI_i, EI_i, KI_i \rangle$. $OI_i = \{OI_{i_0}, \ldots, OI_{i_n}\}$ is the set of object instances occurring in V_i, where an object instance is represented as $OI_{i_j} = \langle frameno, MBR, \mu, type \rangle$. MBR is the minimum bounding rectangle surrounding the object instance. μ represents the certainty of the extraction, where $0 \leq \mu \leq 1$. *type* is an individual of a class CI_j in ontology ONT_j. $EI_i = \{EI_{i_0}, \ldots, EI_{i_n}\}$ is the set of event instances occurring in V_i, where an event instance is represented as $EI_{i_j} = \langle startframeno, endframeno, \mu, type \rangle$. μ represents the certainty of the extraction, where $0 \leq \mu \leq 1$. *type* is an individual of a class CI_j in ontology ONT_j. $KI_i = \{KI_{i_0}, \ldots, KI_{i_n}\}$ is the set of concept instances occurring in V_i, where a concept instance is represented as $KI_{i_j} = \langle startframeno, endframeno, \mu, type \rangle$. μ represents the certainty of the extraction, where $0 \leq \mu \leq 1$. *type* is an individual of a class CI_j in ontology ONT_j.

After giving formal representation and usage of VISCOM, the following sections introduce the main components and relation types of VISCOM.

4.2 VISCOM Class Definitions

The linguistic part of VISCOM contains classes and relations between these classes. Some of the classes represent semantic content types such as *Object* and *Event* while others are used in the automatic semantic content extraction process. Because VISCOM is domain independent, classes and relations are generic and functional for any domain. Relations defined in VISCOM give ability to model events and concepts related with other objects and events.

C-Logic [33] is used for formal representation of classes of the model and operations of the automatic semantic content extraction framework. It is a logical framework for natural representation and manipulation of structured entities in the real world. Entities are classified into various classes according to their attributes. C-Logic allows direct transformation of the specification into first order formulas. It also allows class and subclass specification independently, where this specification can be easily implemented in a programming language. In addition, it supports many useful aspects of sets where this facility is essential in some of the class definitions of VISCOM.

There are some other alternative logical formalisms like O-Logic [75], a primitive version of C-Logic, and F-Logic [67]. F-Logic adds new features on C-Logic and O-Logic to support deduction and formal semantics for object-oriented approaches. Logical languages such as event calculus [69] which are proposed for representing and reasoning events/actions and their effects, are other alternative formalisms that can be used to represent temporal class definitions formally. We did not prefer using O-Logic because it does not support sets as the specification formalism. To be consistent throughout the dissertation, we did not focus on formalisms directly related with temporality. For formal representation, we only need to define classes which have various properties and relations with other classes and rules which contain these classes and relations in their definitions. Therefore, we chose C-Logic because its semantics is first-order, it can be understood easily and it satisfies the dissertation's formal representation needs.

C-Logic proposes a representation framework for entities, their attributes, and classes using identities, labels and types. In the semantics of C-Logic, a class can be defined with only one label, and various pieces of descriptions. In C-logic, descriptions take the following form:

$$ClassName : ObjectIdentifier\,[Attribute_1 \Rightarrow Value_1, \ldots, Attribute_i \Rightarrow Value_i] \quad (4.1)$$

The basic syntax of C-Logic has parentheses, logical connectives $(\wedge, \vee, \neg, \forall, \exists, \supset)$ and

an object language that may contain accountably infinite set of variables, a set of function symbols, a set of predicate symbols, a set of labels and a countable and partially ordered set of type symbols.

The value of an attribute may be simple, an enumerated type or another object description. The range of the atomic properties can be a group or a single element. Formally, the semantics in C-logic is defined directly in terms of object, attribute and value structures.

After giving basic definitions and syntax of C-Logic, it is time to define principal elements of video domain starting with *Video Instance*. A *Video Instance* is composed of Iframes and contains some general video attributes such as name, domain, length and format. With C-Logic it is represented as:

$$Video : \begin{cases} \left[\; metadata \Rightarrow \{M_i\}, \;\; framelist \Rightarrow \{Iframelist\} \; \right] \\ where \\ individual\,(M_i, Metadata)\,. \end{cases} \tag{4.2}$$

where the predicate *individual(Entitiy,Class)* is used to mean "an entity is defined as an individual of a class" in formal representation of classes. General attributes of video mostly named as metadata are out of this dissertation's scope and are only used in the definition of a video instance and represented as:

$$Metadata : \begin{cases} \left[\begin{array}{l} name \Rightarrow [string]\,, \;\; format \Rightarrow [string]\,, \\ length \Rightarrow [float]\,, \;\; domain \Rightarrow \{D_i\} \end{array} \right] \\ where \\ individual\,(D_i, Domain)\,. \end{cases} \tag{4.3}$$

VISCOM is developed on an ontology-based structure where semantic content types and relations between these types are collected under ontology classes and properties of these classes. In addition, there are some domain independent class individuals. VISCOM, VISCOM Classes, VISCOM Data Properties which associate classes with constants and VISCOM Object Properties which are used to define relations between classes are represented with the following formulation:

$$VISCOMClass : \begin{cases} \left[\begin{array}{l} name \Rightarrow [string]\,, \\ dataProp \Rightarrow \{DP_i\}, \\ objectProp \Rightarrow \{OP_j\} \end{array} \right] \\ where \\ individual\,(DP_i, VISCOMDataProperty)\,, \\ individual\,(OP_j, VISCOMObjectProperty)\,. \end{cases} \tag{4.4}$$

$$VISCOMDataProperty : \left\{ \left[\begin{array}{l} name \Rightarrow [string]\,, \\ range \Rightarrow [string, integer, float, ...]\,. \end{array} \right] \right. \tag{4.5}$$

$$VISCOMObjectProperty : \left\{ \begin{array}{l} \left[\begin{array}{l} name \Rightarrow [string]\,, \\ rangeClass \Rightarrow \{RC_i\} \end{array} \right] \\ where \\ individual\,(RC_i, VISCOMClass)\,. \end{array} \right. \tag{4.6}$$

$$VISCOM : \left\{ \begin{array}{l} \left[\begin{array}{l} class \Rightarrow \{C_i\}, \\ temporalRelInd \Rightarrow \left\{ \begin{array}{l} before, meets, equal, overlap, \\ during, finishes, starts \end{array} \right\}, \\ objectCompInd \Rightarrow \left\{ \begin{array}{l} composedOf, isA, memberOf, \\ partOf, substanceOf \end{array} \right\}, \\ movementInd \Rightarrow \{down, up, right, left, stationary\}, \\ spatialRelInd \Rightarrow \left\{ \begin{array}{l} far, near, disjoint, inside, parInside, \\ touch, above, below, left, right \end{array} \right\} \end{array} \right] \\ where \\ individual\,(C_i, VISCOMClass)\,. \end{array} \right. \tag{4.7}$$

Classes have object properties and object properties have range classes in their definition. The following situation is considered in order to guarantee not to have a reference from a class to an object property that uses the same class as the range class. We use the predicate *attribute(X, Y)* to mean "Y is an attribute of class X".

$$\begin{array}{l} individual\,(X, VISCOMClass) \ \wedge \ individual\,(B, VISCOMObjectProperty) \ \wedge \\ individual\,(Y, VISCOMClass) \ \wedge \ individual\,(C, VISCOMObjectProperty) \ \wedge \\ individual\,(B, attribute\,(X, objectProp)) \ \wedge \\ individual\,(Y, attribute\,(B, rangeClass)) \ \wedge \\ individual\,(C, attribute\,(Y, objectProp)) \ \wedge \\ individual\,(X, attribute\,(C, rangeClass)) \ \Rightarrow \ B \neq \ C \end{array} \tag{4.8}$$

In the following sections, each class is introduced with its description, formal representation and relations (relation is named as property in ontology domain and this term is used in class definitions). Properties are given with their names and related classes, if they exist. All of the properties in VISCOM are defined to satisfy one of the following:

1. specialization between a class and its sub-classes,

2. composition of an object or an event made of sub-parts,

3. low-level object features and spatio-temporal relations between objects,

4. temporal relations between events and event related classes.

4.2.1 Component

VISCOM collects all of the semantic content under the class of *Component*. A component can have synonym names and similarity relations with other components. *Component* class has three subclasses as *Objects*, *Events* and *Concepts*. *Component* class is represented as:

$$Component: \begin{cases} \left[\ type \Rightarrow \{O_i, E_j, C_k\},\ sim \Rightarrow \{S_m\},\ synname \Rightarrow [string]\ \right] \\ where \\ individual\,(S_m, Similarity)\,,\ individual\,(O_i, Object)\,, \\ individual\,(E_j, Event)\,,\ individual\,(C_k, Concept)\,, \\ at\ most\ one\ of\ i,\ j,\ k > 0. \end{cases} \tag{4.9}$$

where *hasSynonymName* and *hasSimilarContext* are properties of the *Component* class. *hasSynonymName* is utilized to define synonym names for components. It is used for multilingual extensions and to detect synonym names of a component. *hasSimilarContext* is used to associate similar components in a fuzzy manner when there is a similar component in the ontology with a component that is supposed to be extracted. For example, in the basketball ontology, when a *free throw made* event instance is extracted, a *score* event instance can be extracted by using a *hasSimilarContext* relation individual between *free throw made* and *score* event individuals.

4.2.2 Object

Objects have the narrowest meaning in the domain and they correspond to existential entities. Object is the starting point of the composition. An object has a name, low level features and composed-of relations. *Basketball player*, *referee*, *ball* and *hoop* are examples of objects for the basketball domain.

$$Object: \begin{cases} \left[\begin{array}{l} name \Rightarrow [string]\,,\ lowLevelFeature \Rightarrow \{L_i\}, \\ composedOf \Rightarrow \{COR_j\} \end{array}\right] \\ where \\ individual\,(L_i, LowLevelFeature)\,, \\ individual\,(COR_j, ComposedOfRelation)\,. \end{cases} \tag{4.10}$$

where *hasObjectLowLevelFeature* and *hasComposedOfObjectRelation* are properties of this class. *hasObjectLowLevelFeature* is used to define low level features of an object. Low level feature values are used in rule definitions. For example, in a hospital ontology *patient* and *nurse* are individuals of *person* object. A detected object as *person* can be classified as a *nurse* or *patient* by using the rule definition *"nurses wear white"* in terms of dominant color low-level feature. *hasComposedOfObjectRelation* is used to define concept inclusion, membership and structural object relations such as *part of*, *member of*, *substance of*, *is a* and *composed of*. It has a relevance degree and a reference to an "object composed-of group" individual in its definition.

4.2.3 Event

Events occur within the video and represent the context for objects and object relations. Actually, events are long-term temporal objects and object relation changes. They are described by using objects and spatial/temporal relations between objects. Relations between events and objects and/or their attributes indicate how events are inferred from objects and/or object attributes. In addition, temporal event relations are also used in event definitions. Events have an interval during which they occur. An event has a name, a definition in terms of temporal event relations or spatial/temporal object relations, and role definitions of the objects taking part in the event. *Jump ball*, *rebound* and *free throw* are examples of events for the basketball domain.

$$Event: \begin{cases} \begin{bmatrix} name \Rightarrow [string]\,,\ eventDef \Rightarrow \{ED_i\}, \\ objectRole \Rightarrow \{OR_j\}, \\ temporalEventComp \Rightarrow \{TEC_l\} \end{bmatrix} \\ where \\ individual\,(ED_i, EventDefinition)\,, \\ individual\,(TEC_l, TemporalEventComponent)\,, \\ individual\,(OR_j, ObjectRole)\,, \\ at\ least\ one\ of\ i,\ l > 0. \end{cases} \tag{4.11}$$

where *hasTemporalEventComponent*, *hasEventDefinition* and *hasEventObjectRole* are properties of this class. *hasTemporalEventComponent* is used to define temporal relations between events which are used in the definition of other events. *hasEventDefinition* is utilized to associate events with event definitions. An event can be expressed with more than one event definition. Different definitions of an event can be utilized to increase the possibility to

detect the event. *hasEventObjectRole* is used to define the roles of objects taking part in the definition of an event. Each object can play different roles in different situations. For instance, a *player* is a *shooter* in a *shoot* event, an *assist maker* in an *assist* event.

4.2.4 Concept

Concepts have the widest meanings and they are disjoint from object and events. Concepts express some special meaning themselves. A concept can enclose objects, events and other concepts having narrower meaning than its meaning. Each concept has a relation with the component that can be placed in its meaning. *Attack* and *defense* are examples of concepts for the basketball domain.

$$Concept: \begin{cases} \left[\ name \Rightarrow [string]\,,\ conceptComp \Rightarrow \{CC_i\}\ \right] \\ where \\ individual\,(CC_i, ConceptComponent)\,,\ i > 0. \end{cases} \tag{4.12}$$

where *hasConceptComponent* is the only property of this class. *hasConceptComponent* is used to define the relation that is placed in concept's meaning. This relation is fuzzy and the degree of it denotes the degree of inclusion. Object, event and concept individuals are used in concept definitions. For instance, *attack* concept in a basketball match is related with *score*, *rebound* and *free throw* events. Whenever one of these events is extracted, it is inferred that *attack* concept happens with the relevance degree defined in its definition.

4.2.5 Spatial Relation

Spatial relations express the relative object positions between two objects such as *above*, *inside* or *far*. The categorization given in [41] is used to define spatial relation types between objects. In this categorization, spatial relation types are grouped under three categories as topological, distance and positional spatial relations. The individuals of this class are utilized by the individuals of *Spatial Relation Component* class.

$$SpatialRelation: \begin{cases} \left[\ type \Rightarrow \{T_i,\ P_j,\ D_k\}\ \right] \\ where \\ individual\,(T_i, TemporalSpatialChange)\,, \\ individual\,(P_j, PositionalSpatialChange)\,, \\ individual\,(D_k, DistanceSpatialChange)\,, \\ at\ least\ one\ of\ i,\ j,\ k > 0. \end{cases} \tag{4.13}$$

$$TopologicalSpatialRelation : \left\{ \left[relType \Rightarrow \left\{ \begin{array}{l} inside,\ partiallyInside, \\ disjoint,\ touch \end{array} \right\} \right] \right. \tag{4.14}$$

$$PositionalSpatialRelation : \left\{ \left[relType \Rightarrow \left\{ \begin{array}{l} rightSide,\ leftSide, \\ above,\ below \end{array} \right\} \right] \right. \tag{4.15}$$

$$DistanceSpatialRelation : \left\{ \left[reltype \Rightarrow \{far,\ near\} \right] \right. \tag{4.16}$$

4.2.6 Spatial Relation Component

Spatial Relation Component class is used to represent spatial relations between object individuals. It takes two object individuals and at most one spatial relation individual from each sub class of *Spatial Relation* class. This class is utilized in spatial change and event definition modeling. It is possible to define imprecise relations by specifying the membership value for the spatial relation individual used in its definition with a data property. For the basketball domain *Player under Hoop*, *Ball is near Player* are examples of *Spatial Relation Component* class individuals.

$$SpatialRelationComponent : \left\{ \begin{array}{l} \left[\begin{array}{l} name \Rightarrow [string], \\ object1 \Rightarrow \{O_i\},\ object2 \Rightarrow \{O_j\}, \\ spatialRelation \Rightarrow \{SR_k\}, \\ membershipValue \Rightarrow [\mu_l] \end{array} \right] \\ where \\ individual\,(O_i, Object), \\ individual\,(O_j, Object), \\ individual\,(SR_k, SpatialRelation), \\ 0 \leq \mu_l \leq 1, i \neq j. \end{array} \right. \tag{4.17}$$

where *hasSpatialRelation*, *hasObject*, *hasSubject* and *hasSpatialRelationMembershipValue* are properties of this class. *hasSpatialRelation* is used to define the spatial relation type individual between object individuals. *hasObject* is used to represent the first object individual in the spatial relation. *hasSubject* is used to represent the second object individual in the spatial relation. *hasSpatialRelationMembershipValue* is used to assign a membership value to the spatial relation between objects.

4.2.7 Spatial Change

Spatial Change class is utilized to express spatial relation changes between objects or spatial movements of objects. We use both types in the definition of *Spatial Change* class in order to model events.

Simply, objects refer to semantic real world entity definitions that are used to denote a coherent spatial region. Spatial regions representing objects have spatial relations between each other. These relations continuously change in time. This information is utilized in event definitions. An example can be given from the basketball domain: In the definition of *scoring* event, a spatial change component individual is used where its initial spatial relation component individual is *Ball is above Hoop* and its final spatial relation component individual is *Ball is below Hoop*.

More than one spatial relation change can be used to make an event definition. Temporal relations between spatial changes are used when more than one spatial change is needed for definition. This concept is explained under *Temporal Relations* and *Event Definition* classes. A spatial change also contains role definitions of the objects taking part in the event.

$$SpatialChange: \begin{cases} \left[\begin{array}{l} name \Rightarrow [string],\ initialSRC \Rightarrow \{SRC_i\}, \\ finalSRC \Rightarrow \{SRC_j\},\ objectRole \Rightarrow \{OR_k\}, \\ spatialMovement \Rightarrow \{SM_m\} \end{array}\right] \\ where \\ individual\,(SRC_i, SpatialRelationComponent)\,, \\ individual\,(SRC_j, SpatialRelationComponent)\,, \\ individual\,(OR_k, ObjectRole)\,, \\ individual\,(SM_m, SpatialMovement)\,, \\ exactly\ one\ of\ i,\ m > 0,\ if\ i > 0\ then\ j > 0. \end{cases} \tag{4.18}$$

where *hasInitialSpatialRelationComponent*, *hasFinalSpatialRelationComponent*, *hasSpatialMovementComponent*, and *hasSpatialChangeObjectRole* are properties of this class. *hasInitialSpatialRelationComponent* is used to represent the initial spatial relation component individual of the spatial change component. *hasFinalSpatialRelationComponent* is used to represent the final spatial relation component individual of the spatial change component. *hasSpatialMovementComponent* is used to define single object movements. *hasSpatialChangeObjectRole* is used to define object roles in spatial changes. This information is used in event definitions in order to extract roles played during event executions.

4.2.8 Spatial Change Period

Spatial changes have an interval that is designated by the spatial relation individuals used in their definitions. In the semantic content extraction process, spatial relations between objects are automatically extracted. Spatial relations are momentary situations but periods of spatial relations can be extracted from consecutive frames. Whenever the temporal situation between *Spatial Relation Component* individuals defined in a *Spatial Change* individual is satisfied, the *Spatial Change* individual is extracted and these periods are utilized to calculate the *Spatial Change* individual's interval. According to the meaning of the spatial change, periods of spatial relations should be included or discarded in the calculation of spatial change intervals. In order to address this need, we define *Spatial Change Period* class.

It has four individuals as *startToEnd*, *startToStart*, *endToStart* and *endToEnd*. *startToEnd* is used to mean that the first frame of the spatial change interval will be the first frame of the initial spatial relation component and the last frame of the spatial change interval will be the last frame of the final spatial relation component. *startToStart* is used to mean that the first frame of the spatial change interval will be the first frame of the initial spatial relation component and the last frame of the spatial change interval will be the first frame of the final spatial relation component. *endToStart* is used to mean that the first frame of the spatial change interval will be the last frame of the initial spatial relation component and the last frame of the spatial change interval will be the first frame of the final spatial relation component. *endToEnd* is used to mean that the first frame of the spatial change interval will be the last frame of the initial spatial relation component and the last frame of the spatial change interval will be the last frame of the final spatial relation component.

$$SpatialChangePeriod : \left\{ \left[relType \Rightarrow \left\{ \begin{array}{l} startToStart,\ startToEnd, \\ endToStart,\ endToEnd \end{array} \right\} \right] \right. \tag{4.19}$$

4.2.9 Spatial Movement

Second alternative to define a spatial change is using spatial movements. Spatial movements represent spatial changes of single objects. This class is used to define movement types. It has 5 individuals as; *moving to left*, *moving to right*, *moving up*, *moving down* and *stationary*. For the basketball domain, in the definition of *giving a pass* event, the movement of the ball to left or right is used as a spatial change. *Spatial Movement* class individuals are used by *Spatial Movement Component* class individuals.

$$SpatialMovement : \left\{ \left[relType \Rightarrow \left\{ \begin{array}{l} movesLeft,\ movesRight, \\ movesUp,\ movesDown \\ stationary \end{array} \right\} \right] \right. \tag{4.20}$$

4.2.10 Spatial Movement Component

Spatial Movement Component class is used to declare object movement individuals. *"Ball moves left"* is an example of an individual of this class is.

$$SpatialMovementComponent : \left\{ \begin{array}{l} \left[\begin{array}{l} name \Rightarrow [string]\,,\ Object \Rightarrow \{O_i\}, \\ spatialmovement \Rightarrow \{SM_j\} \end{array} \right] \\ where \\ individual\,(O_i, Object)\,, \\ individual\,(SM_j, SpatialMovement)\,. \end{array} \right. \tag{4.21}$$

where *hasMovingObject* and *hasSpatialMovement* are properties of this class. *hasMovingObject* is used to define the object individual that realizes the movement. *hasSpatialMovement* is used to define the direction of the movement.

4.2.11 Temporal Relation

Actually events are long-term temporal objects or object relation changes, which usually extend over tens or hundreds of frames. We use temporal relations to order *Spatial Changes* or *Events* in *Event Definitions*. Allen's temporal relationships [16] are used to express parallelism and mutual exclusion between components. For the basketball domain, in the definition of *scoring* event, the temporality between two spatial change individuals are used as; *"Ball passing through the Hoop" Spatial Change* individual occurs *after "Throwing Ball" Spatial Change* individual.

$$TemporalRelation : \left\{ \left[relType \Rightarrow \left\{ \begin{array}{l} before,\ during,\ finishes,\ meet, \\ overlap,\ starts,\ equal \end{array} \right\} \right] \right. \tag{4.22}$$

4.2.12 Temporal Event Component

Temporal Event Component class is used to define temporal relations between *Event* individuals. *Temporal Event Relation* individuals are used by *Event Definition* individuals. For example, *Shot Made* event occurs after *Pass* event is used in the definition of *Assist* event.

$$TemporalEventComponent: \begin{cases} \begin{bmatrix} name \Rightarrow [string], \\ initialE \Rightarrow \{E_i\},\ finalE \Rightarrow \{E_j\}, \\ tempRelation \Rightarrow \{TR_k\} \end{bmatrix} \\ where \\ individual\,(E_i, Event), \\ individual\,(E_j, Event), \\ individual\,(TR_k, TemporalRelation), \\ i \neq j. \end{cases} \tag{4.23}$$

where *hasTemporalEventRelation*, *hasFirstEvent* and *hasSecondEvent* are properties of this class. *hasTemporalEventRelation* is used to define the temporal relation type between events. *hasFirstEvent* is used to represent the first *Event* individual in the temporal relation. *hasSecondEvent* is used to represent the second *Event* individual in the temporal relation.

4.2.13 Temporal Spatial Change Component

Temporal Spatial Change Component class is used to define temporal relations between spatial changes in *Event* definitions. For instance, the temporal relation *after* is used between *Ball hits Hoop* and *Player jumps Spatial Change* individuals in the definition of *Rebound* event.

$$TemporalSpatialChange: \begin{cases} \begin{bmatrix} name \Rightarrow [string],\ tempRelation \Rightarrow \{TR_k\}, \\ initialSC \Rightarrow \{SC_i\},\ finalSC \Rightarrow \{SC_j\} \end{bmatrix} \\ where \\ individual\,(SC_i, SpatialChange), \\ individual\,(SC_j, SpatialChange), \\ individual\,(TR_k, TemporalRelation), \\ i \neq j. \end{cases} \tag{4.24}$$

where *hasTempSpatialChangeRelation*, *hasFirstSpatialChange* and *hasSecondSpatialChange* are properties of this class. *hasTemporalSpatialChangeRelation* is used to define the temporal relation type between *Spatial Change* individuals. *hasFirstSpatialChange* is used to represent the first *Spatial Change* individual in the temporal relation. *hasSecondSpatialChange* is used to represent the second *Spatial Change* individual in the temporal relation.

4.2.14 Event Definition

An event can have several definitions where each definition describes the event with a certainty degree. In other words, each event definition has a membership value for the event it defines that denotes the clarity of description. Event definitions contain individuals of *Spatial Change*, *Spatial Relation Component* or *Temporal Spatial Change Component* classes.

$$EventDefinition: \begin{cases} \left[\begin{array}{l} name \Rightarrow [string]\,,\ objectRole \Rightarrow \{OR_k\}, \\ spatialRelComp \Rightarrow \{SRC_j\},\ relevance \Rightarrow [\mu_i]\,, \\ tempSpatialChange \Rightarrow \{TSCC_l\}, \\ uniqueSpatialChange \Rightarrow \{USC_m\} \end{array}\right] \\ where \\ individual\,(SRC_j, SpatialRelationComponent)\,, \\ individual\,(OR_k, ObjectRole)\,, \\ individual\,(TSC_l, TemporalSpatialChangeComponent)\,, \\ individual\,(USC_m, SpatialChange)\,, \\ exactly\ one\ of\ j,\ l,\ m > 0. \end{cases} \tag{4.25}$$

where *hasUniqueSpatialChange*, *hasTemporalSpatialChangeComponent*, *hasEventSpatialRelationComponent*, *hasEventRelevance* and *hasEventDefinitionObjectRole* are properties of *Event Definition* class. Event definitions generally contain more than one *Spatial Change* individual which are temporally related with each other. *hasUniqueSpatialChange* is used for cases when single *Spatial Change* individual is enough to make the event definition. *hasTemporalSpatialChangeComponent* is used to model temporal spatial change relations. *hasEventSpatialRelationComponent* is used when a spatial relation between two objects is enough to make the event definition. *hasEventRelevance* is used to define the relevance of the definition to the event. *hasEventDefinitionObjectRole* is used to define object roles in the event definition. Object roles are used to extract roles occurred during the event execution.

4.2.15 Concept Component

Concept Component class is used to associate components to a concept semantically. This association is fuzzy and the degree of it denotes the degree of inclusion. If an object is used in the definition of a concept, the role of this object in this concept is also represented within this class. This class is utilized in the concept extraction process.

$$ConceptComponent : \begin{cases} \left[\begin{array}{l} name \Rightarrow [string]\,,\ relevance \Rightarrow [\mu_i]\,, \\ objectRole \Rightarrow \{OR_j\},\ component \Rightarrow \{COM_i\} \end{array} \right] \\ where \\ individual\,(COM_i, Component)\,, \\ individual\,(OR_j, ObjectRole)\,, \\ 0 \leq \mu_i \leq 1. \end{cases} \tag{4.26}$$

where *hasRelevance*, *hasComponent* and *hasConceptObjectRole* are properties of this class. *hasRelevance* is used to define the relevance degree of the component with the related concept. *hasComponent* is used to define the component which can be an *Object*, an *Event* or a *Concept* individual related with the concept. *hasConceptObjectRole* is used to define the role of the object that is used in the concept definition.

4.2.16 Object Role and Role

Object Role class is used to represent roles. An object may play different roles in different situations. Even in a unique event, it may commit different roles at different stages of the event. For example, *Player* takes *Assist Maker* role in *Assist* event and *Rebounder* role in *Rebound* event.

$$ObjectRole : \begin{cases} \left[\ name \Rightarrow [string]\,,\ role \Rightarrow \{R_i\},\ object \Rightarrow \{O_j\} \ \right] \\ where \\ individual\,(R_i, Role)\,,\ individual\,(O_j, Object) \end{cases} \tag{4.27}$$

where *hasRoledObject* and *hasRole* are properties of this class. *hasRoledObject* is used to define the *Object* individual in this relation. *hasRole* is used to define the role of the object. Role class is used to define all possible role types such as *Assist Maker*.

$$Role : \left\{ \ \left[\ name \Rightarrow [string] \ \right] \right. \tag{4.28}$$

4.2.17 Low Level Feature

All of the classes defined until now are introduced as having a functionality of temporal or spatial relations for modeling. *Low Level Feature* class adds low level modeling capability to the model.

$$LowLevelFeature : \left\{ \ \left[\ name \Rightarrow [string]\,, value \Rightarrow [float] \ \right] \right. \tag{4.29}$$

where *hasLowLevelFeatureValue* and *hasLowLevelFeatureName* are properties of this class. *hasLowLevelFeatureValue* is used to define the low level feature value. *hasLowLevelFeature-Name* is used to define the low level feature name.

4.2.18 Similarity

Similarity class is used to represent the relevance of a component to another component in a fuzzy manner. It occupies the similar component reference in its definition. Whenever a component which has a similarity relation with another component is extracted, the semantically related component is automatically extracted by using this similarity relation.

$$Similarity: \begin{cases} \left[\; name \Rightarrow [string] \,,\; relevance \Rightarrow [\mu_i] \,,\; simWith \Rightarrow \{COM_j\} \; \right] \\ where \\ individual\,(COM_j, Component) \,,\; 0 \leq \mu_i \leq 1. \end{cases} \tag{4.30}$$

where *hasSimilarityRelevance* and *hasSimilarityWith* are properties of this class. *hasSimilarityRelevance* is used to define the degree of relevance. *hasSimilarityWith* is used to define the *Component* individual that is similar to the related *Component* indiviual.

4.2.19 Object Composed of Relations

Object Composed Of Type class is used to define concept inclusion, membership and structural object relation types such as *isA, partOf, substanceOf, composedOf* and *memberOf.*

$$ObjectComposedOfType: \left\{ \left[\; relType \Rightarrow \left\{ \begin{array}{l} isA,\; memberOf,\; partOf, \\ composedOf,\; substanceOf \end{array} \right\} \right] \right. \tag{4.31}$$

Object Composed Of Group class is used to define the *Object Composed Of Type* relation with the parent *Object* individual.

$$ObjectComposedOfGroup: \begin{cases} \left[\begin{array}{l} name \Rightarrow [string] \,,\; type \Rightarrow \{OCT_i\}, \\ object \Rightarrow \{O_j\}, \end{array} \right] \\ where \\ individual\,(O_j, Object) \,, \\ individual\,(OCT_i, ObjectComposedOfType) \,. \end{cases} \tag{4.32}$$

where *hasComposedOfType* and *hasParentObject* are properties of this class. *hasComposed-OfType* is used to define the type of the relation. *hasParentObject* is used to define the parent *Object* individual of the composed of relation.

Object Composed Of Relation class is used to define relations like "*"Basketball Hoop* is part of *Basket*". It has a data property which defines the relevance degree of the *Object* individual to the reference *Object Composed-of Group* individual in its definition.

$$ObjectComposedOfRelation: \begin{cases} \begin{bmatrix} name \Rightarrow [string]\,,\ group \Rightarrow \{CG_i\}, \\ relevance \Rightarrow [\mu_i] \end{bmatrix} \\ where \\ individual\,(CG_i, ObjectComposedOfGroup) \\ 0 \leq \mu_i \leq 1. \end{cases} \tag{4.33}$$

where *hasObjectToParentRelevance* and *hasObjectComposedOfGroup* are properties of this class. *hasObjectToParentRelevance* is used to define the degree of relevance. *hasObject-ComposedOfGroup* is used to define the relation that specifies the parent object and the composed-of type information.

Object Composed-of classes are utilized to extract objects which have relations with other objects. Because there is no applied low-level extraction process, the spatial information of object instances extracted by using individuals of *Object Composed-Of* classes can not be fixed. Therefore, these classes are not utilized in the semantic content extraction process.

In Table 4.1, VISCOM class dependencies is listed. VISCOM classes and relations are given in Figure 4.1. VISCOM OWL code is given in Appendix A.

Table 4.1: VISCOM Class Dependencies

Class name	Dependent Classes
Component	Concept Component, Similarity
Object	Object Role, Spatial Movement Component, Spatial Relation Component, Object Composed of Relation
Event	Temporal Event Component
Concept	-
Spatial Relation	Spatial Relation Component
Spatial Change	Event Definition, Temporal Spatial Change Component
Spatial Movement	Spatial Movement Component
Temporal Relation	Temporal Spatial Change Component, Temporal Event Component
Event Definition	Event
Temporal Event Component	Event
Temporal Spatial Change Component	Event Definition
Object Role	Concept, Event, Event Definition, Spatial Change
Concept Component	Concept
Low Level Feature	Object
Similarity	Component
Spatial Movement Component	Spatial Change
Object Composed Of Relation	Object
Role	Object Role
Object Composed Of Type	Object Composed Of Relation
Spatial Relation Component	Event Definition, Spatial Change
Spatial Change Period	Spatial Change

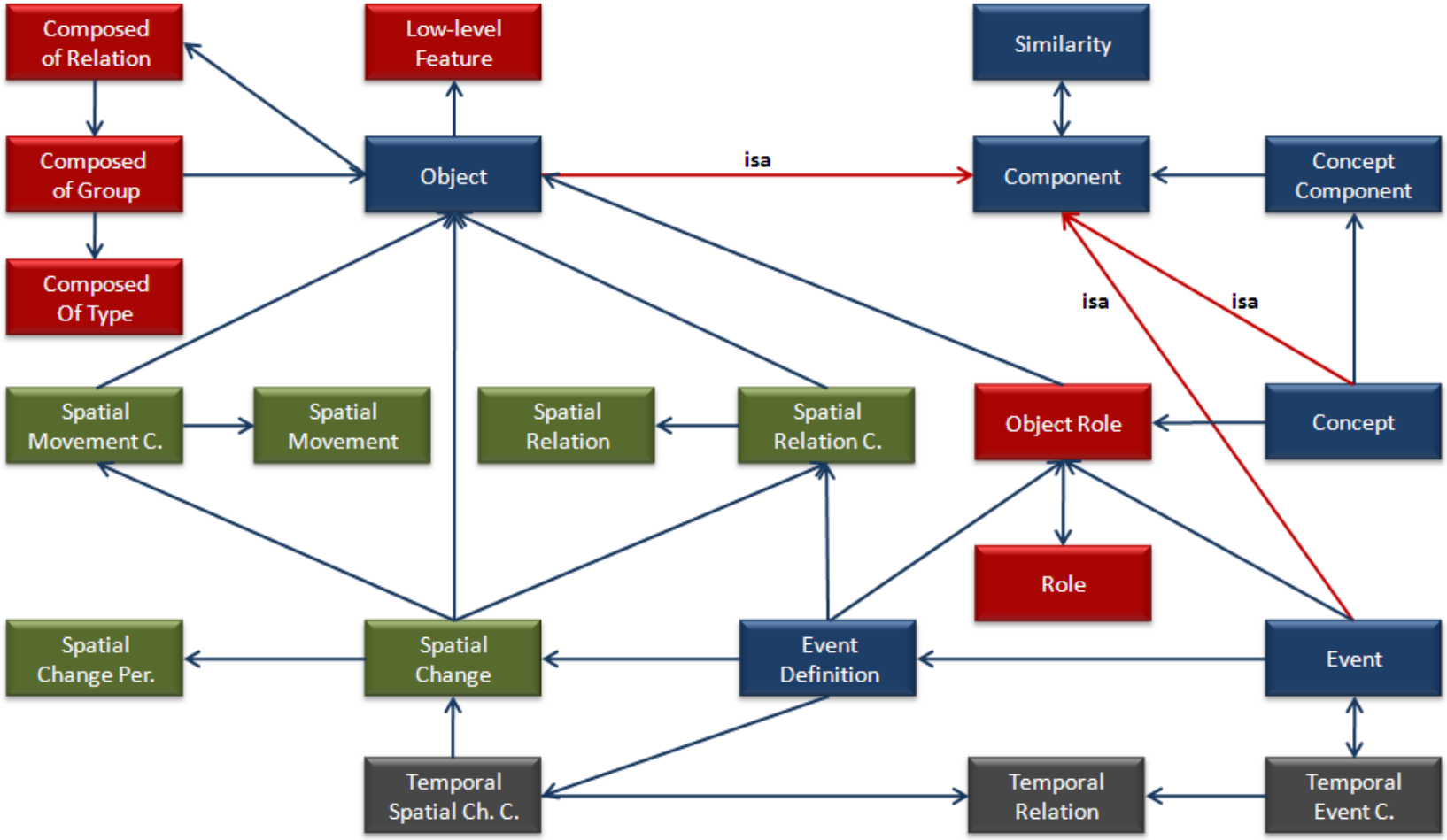


Figure 4.1: VISCOM Classes and Relations

4.3 Domain Ontology Construction with VISCOM

VISCOM is utilized as a meta model to construct domain ontologies. It has a very generic architecture that is applicable to construct ontologies for different domains. Basically, domain specific semantic contents are defined as individuals of VISCOM classes and properties.

$$DomainOntology: \begin{cases} \begin{bmatrix} metaModel \Rightarrow [VISCOM], \ domain \Rightarrow \{D_i\}, \\ classInds \Rightarrow \{CI_0 \ \cdots \ CI_k\}, \\ dataPropertyInds \Rightarrow \{DP_0 \ \cdots \ DP_l\}, \\ objectPropertyInds \Rightarrow \{OP_0 \ \cdots \ OP_m\} \end{bmatrix} \\ where \\ CIs \ are \ domain \ specific \ class \ individuals, \\ DPs \ are \ domain \ specific \ data \ property \ individuals, \\ OPs \ are \ domain \ specific \ object \ property \ individuals, \\ individual\,(D_i, Domain)\,. \end{cases} \tag{4.34}$$

Algorithm 1 presents the steps followed to construct a domain ontology by using VISCOM. The first step is defining semantic content types as individuals of *Object*, *Event* and *Concept* classes. The next step is defining all possible spatial relations between *Object* individuals as *Spatial Relation Component* individuals that can occur and be used to define an event individual. If there are object movement definitions that occur within an event, they are created as *Spatial Movement Component* class individuals. All of the *Spatial Relation Component* and *Spatial Movement Component* individuals are utilized to define *Spatial Change* individuals. Temporal relations between *Spatial Change* individuals are used to create *Temporal Spatial Change Component* individuals. *Spatial Change*, *Spatial Relation Component* and *Temporal Spatial Change Component* individuals are used to create *Event Definition* individuals. There are two possible ways of defining an *Event* individual. The first way is using *Event Definition* individuals. The second way is using *Temporal Event Component* individuals where events are defined with temporal relations between events. *Concept* individuals are defined with object, event and concept individuals. To achieve this, *Concept Component* individuals are created with *Object*, *Event* or *Concept* individuals. As the last step, *Similarity*, *Role* and *Object Role* individuals are created.

We have constructed an *Office Surveillance Ontology* and a *Basketball Ontology* by using VISCOM. We started ontology creation with *Object* (player, referee, coach, basket, hoop, free throw line, ball ..), *Event* (free throw, shoot, dunk, rebound, pass, assist, jump ball ...) and then *Concept* (attack, defense, match,...) individuals for basketball ontology. We

have analyzed spatial and temporal nature of events to define individuals of *Spatial Change* and *Spatial Movement Component* classes. We added temporal relation individuals between component individuals to the ontology after defining spatial relation individuals. Additional features such as *Similarity* and *Object Role* individuals were added to the ontology. Because the visual representation of the basketball ontology is too big, small portions of this ontology is illustrated in Figure 4.2, Figure 4.3 and Figure 4.4 with *Rebound* event, *Free throw made* event and *Attack* concept respectively.

Algorithm 1 Ontology Construction with VISCOM

Input : VISCOM

Output : Domain Ontology

1: define *object*, *event* and *concept* individuals.
2: define all possible *spatial relation* individuals that can occur within an *event* individual.
3: define all possible *object movement* individuals that can occur within an *event* individual.
4: use *spatial relation* and *movement* individuals to define *spatial change* individuals.
5: describe temporal relations between *spatial change* individuals as *temporal spatial change component* individuals.
6: make *event definitions* with *spatial change*, *spatial relation* and *temporal spatial change component* individuals.
7: **for all** *event* individual **do**
8: **if** an event can be defined with an event definition **then**
9: define *event* individual in terms of *event definition* individuals.
10: **end if**
11: **if** an event can be defined with temporal relations between other events **then**
12: define *event* individuals in terms of *event temporal relation* individuals.
13: **end if**
14: **end for**
15: **for all** *concept* individuals **do**
16: construct a relation with the *component* individual that can be placed in its meaning.
17: **end for**
18: define *similarity* individuals.
19: define all *object role* individuals taking place in *spatial change*, *event definition*, *event* and *concept* individuals.

4.4 Rule-based Extension

In addition to VISCOM, rules are utilized to extend the modeling capabilities of the dissertation. Rules consist of VISCOM domain individuals. Each rule has two parts as *body* and *head* where *body* part contains any number of domain class or property individuals and *head* part contains only one individual with a value, μ, representing the certainty of the definition given in the *body* part to represent the definition in the *head* part where $0 \leq \mu \leq 1$. The basic syntax of rules has parentheses and logical connectives $(\wedge, \vee, \neg, \forall, \exists, \supset)$ in both *body* and *head* parts.

$$Rule: \begin{cases} \left[\begin{array}{l} body \Rightarrow \{VCI_i, VDPI_j, VOPI_k, Connector\}, \\ head \Rightarrow \{VCI_m,\ \mu\} \end{array}\right] \\ where \\ individual\,(VCI_i, VISCOMClass\ Individuals)\,, \\ individual\,(VDPI_j, VISCOMDataProperty\ Individuals)\,, \\ individual\,(VOPI_k, VISCOMObjectProperty\ Individuals)\,, \\ individual\,(VCI_m, VISCOMClass\ Individuals)\,, \\ connector \Rightarrow \{\wedge, \vee, \neg, \forall, \exists, \supset\}. \end{cases} \tag{4.35}$$

Rule definitions are used for two different purposes. The first set of rules are defined to lower spatial relation computation cost. Inverse spatial relations and spatial relations that can be described in terms of other spatial relations are expressed with rule definitions. In the spatial relation extraction process, these rules are utilized to extract the content represented with the *head* part of the rule definition automatically. Rule definitions for *Below* positional relation and *Near* distance relation types are presented as examples of rule usage for this kind.

$$BelowRule: \begin{cases} \left[\begin{array}{l} hasObject(?SRC, ?O)\ \wedge\ hasSubject(?SRC, ?S)\ \wedge \\ hasSpatialRelation(?SRC, positionalAbove) \end{array}\right], \\ \\ \left[\ hasSpatialRelation(?NewSRC, positionBelow)\ \right] \\ where \\ individual\,(SRC, SpatialRelationComponent)\,, \\ individual\,(NewSRC, SpatialRelationComponent)\,, \\ individual\,(O, Object)\,,\ individual\,(S, Object)\,. \end{cases} \tag{4.36}$$

$$NearRule: \begin{cases} \left[\begin{array}{l} hasObject(?SRC, ?O) \ \wedge \ hasSubject(?SRC, ?S) \ \wedge \\ (hasSpatialRelation(?SRC, topologicalInside) \ \vee \\ hasSpatialRelation(?SRC, topologicalParInside) \ \vee \\ hasSpatialRelation(?SRC, positionalTouch)) \end{array}\right], \\ \left[\ hasSpatialRelation(?NewSRC, distanceNear)\ \right] \\ where \\ individual\,(SRC, SpatialRelationComponent)\,, \\ individual\,(NewSRC, SpatialRelationComponent)\,, \\ individual\,(O, Object)\,,\ individual\,(S, Object)\,. \end{cases} \tag{4.37}$$

Ontologies have classes and a set of relations between these classes that define the general structure of a domain. Nearly every domain has a number of irregular situations that can not be represented with the relation sets defined in the ontology. VISCOM is enriched with rule definitions where it is hard to define situations as a natural part of ontology. The second rule set is utilized to be able to define such complex situations more effectively and to make semantic content extraction.

Rules can contain any class/property individual defined in the ontology. In this way, events and concepts can be represented with rules. In fact, VISCOM is adequate to represent any kind of event definition in terms of spatial or/and temporal relations and similarity definitions. Rules give the opportunity to make the event definitions which contain a set of events or other class individuals defined in the domain ontology.

Concept individuals in VISCOM utilizes object, event and concept individuals in their definition. A relevance degree is used to represent how relevant the object, event or concept to the concept is. This representation can also be made with a fuzzy rule definition. An example rule definition is given below:

$$TalkingRule: \begin{cases} \left[\ Event(welcome)\ \right], \\ \left[\ Concept(talking)\ \wedge\ hasValue(0.8)\ \right] \end{cases} \tag{4.38}$$

Unfortunately, *Concept Component* individuals can take only one component individual in their definitions. Rules are utilized to make concept definitions which can be represented only with multiple individuals. Example rules of this kind are given below. In each rule, a

set of VISCOM class or property individual is used to define a concept with a degree.

$$ConceptRule: \begin{cases} \left[\ Object(?O) \wedge\ Event(?E)\ \right], \\ \\ \left[\ Concept(?C)\ \wedge\ hasValue(\mu)\ \right] \\ where \\ individual\,(O, Object)\,,\ individual\,(E, Event)\,, \\ individual\,(C, Concept)\,,\ 0 \leq \mu \leq 1. \end{cases} \tag{4.39}$$

$$ConceptRule2: \begin{cases} \left[\ hasConceptComponent(?X, ?Y) \wedge\ Event(?E)\ \right], \\ \\ \left[\ Concept(?C)\ \wedge\ hasValue(\mu)\ \right] \\ where \\ individual\,(C, Concept)\,,\ individual\,(E, Event)\,, \\ individual\,(X, Component)\,, \\ individual\,(Y, ConceptComponent)\,, \\ 0 \leq \mu \leq 1. \end{cases} \tag{4.40}$$

$$WorkingRule: \begin{cases} \left[\begin{array}{l} ConceptComponent(typingToSitting) \wedge \\ SpatialRelation(person,\ screen,\ near) \end{array}\right], \\ \\ \left[\ Concept(working)\ \wedge\ hasValue(0.7)\ \right] \end{cases} \tag{4.41}$$

$$BusyRule: \begin{cases} \left[\begin{array}{l} hasConceptComponent(?X, workingToSitting) \wedge \\ hasConceptComponent(?X, workingToPrinting) \end{array}\right], \\ \\ \left[\ Concept(busy)\ \wedge\ hasValue(0.65)\ \right] \\ where \\ individual\,(X, Component)\,. \end{cases} \tag{4.42}$$

Rule definitions strengthened the framework in terms of both semantic content representation and semantic content extraction.

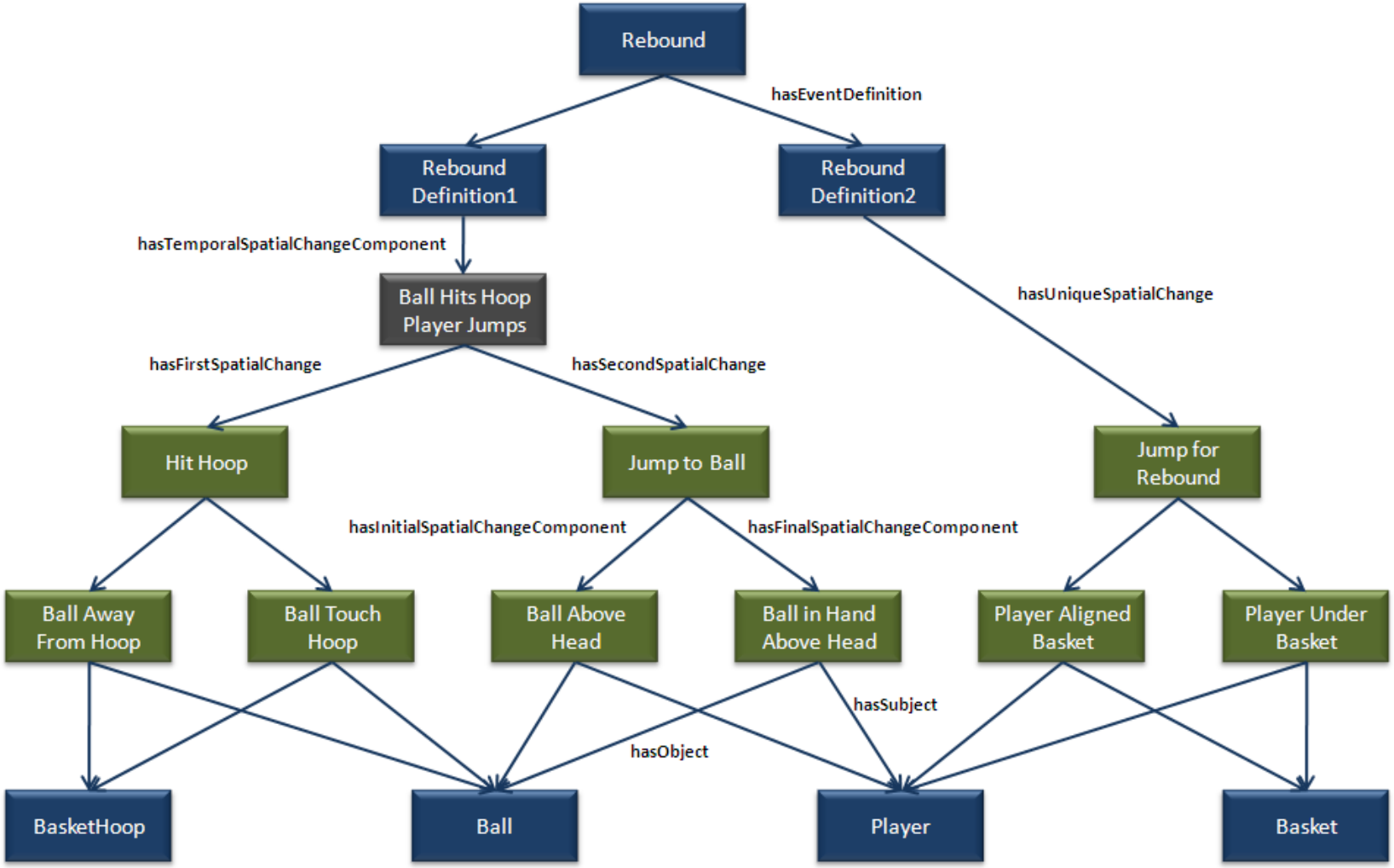


Figure 4.2: Rebound Event Representation

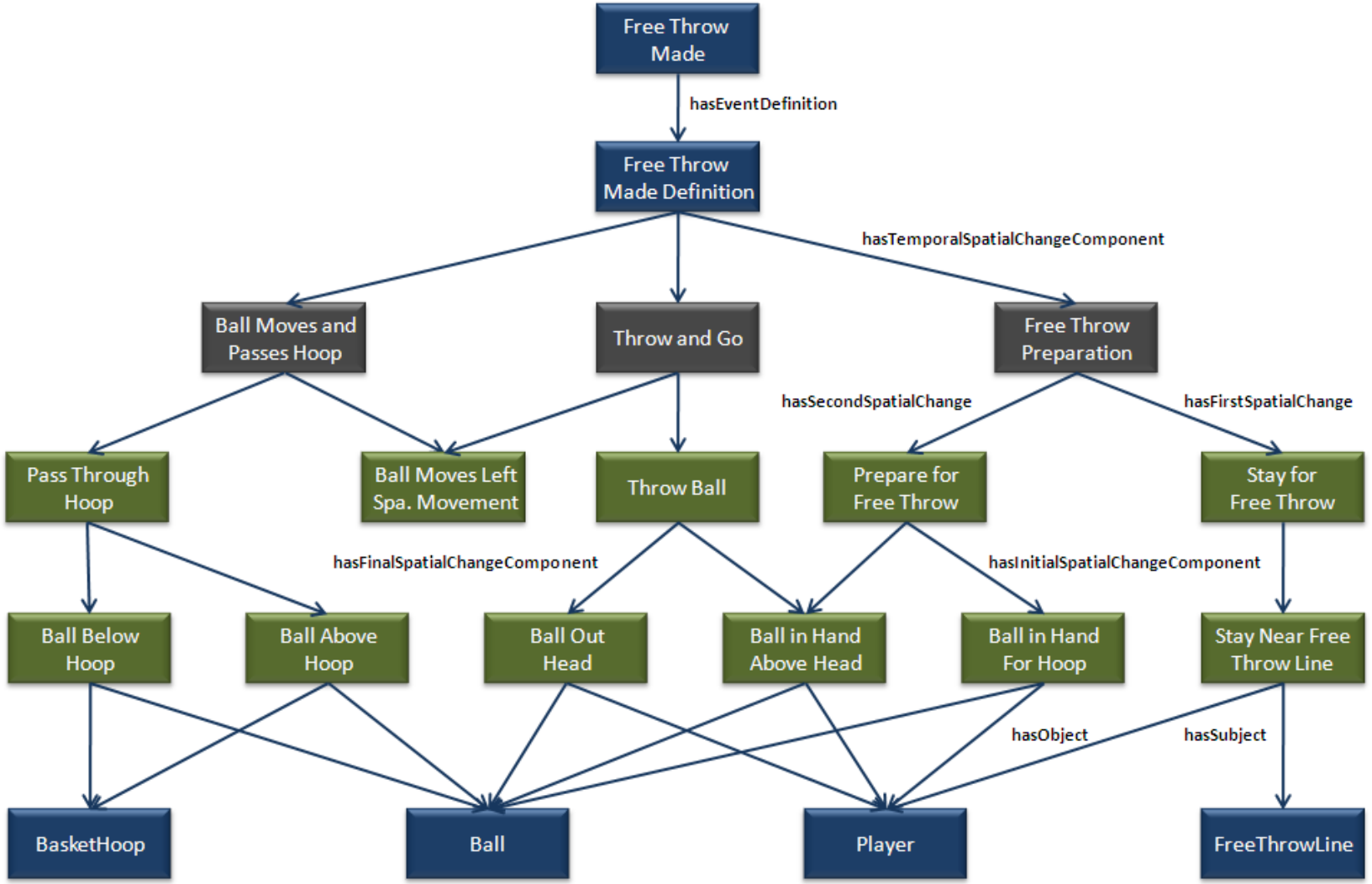


Figure 4.3: Free Throw Made Event Representation

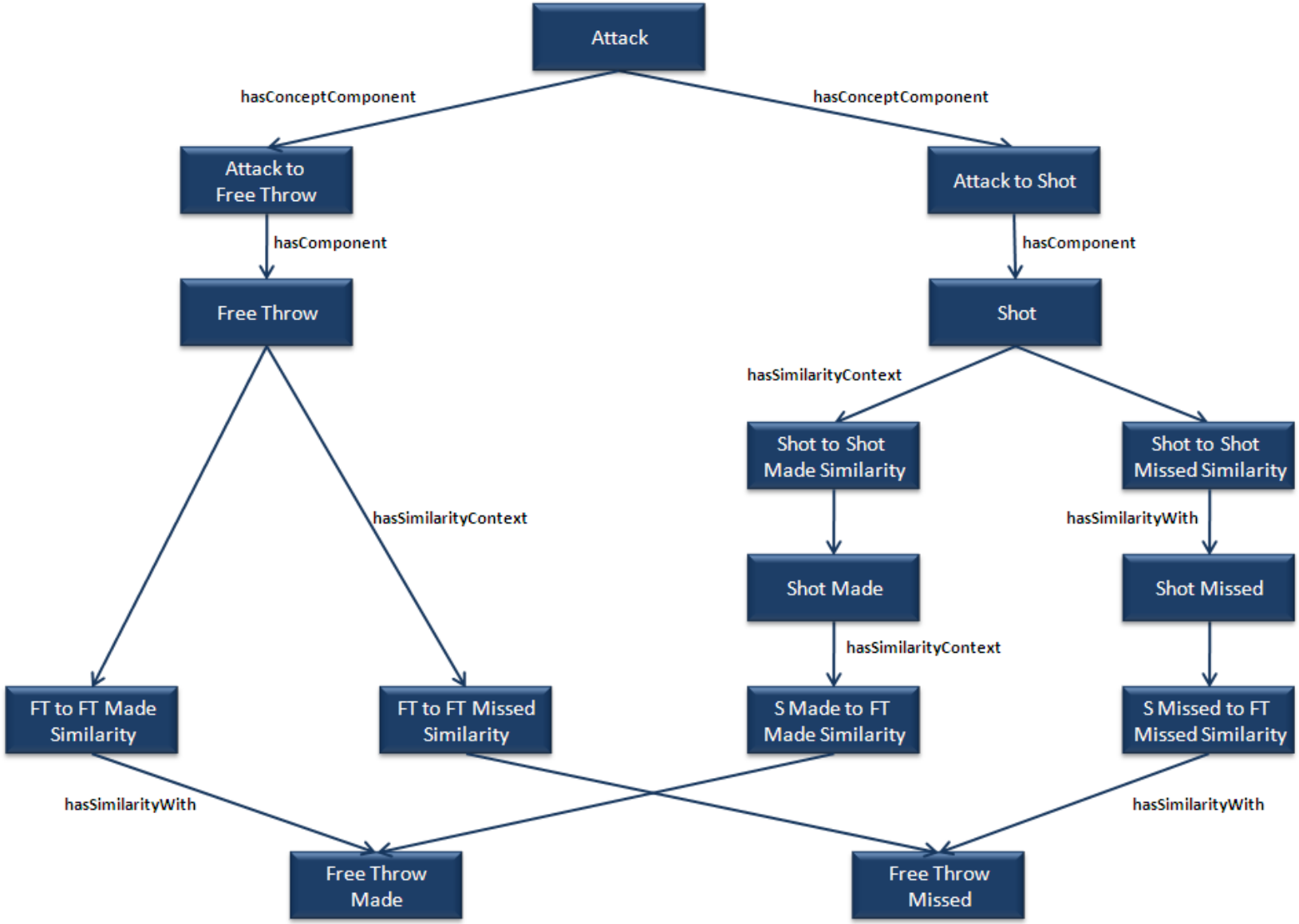


Figure 4.4: Attack Concept Representation

CHAPTER 5

AUTOMATIC SEMANTIC CONTENT EXTRACTION FROM VIDEOS

In this chapter, the architecture of the **A**utomatic **S**emantic **C**ontent **E**xtraction **F**ramework, ASCEF, is explained. The ultimate goal of ASCEF is to extract all of the semantic content existing in video instances. In order to achieve this goal, the framework utilizes the domain ontology and a set of rules defined for the related domain as input. As a result, the extraction process outputs a set of semantic contents.

$$ASCEF: \begin{cases} \begin{bmatrix} input \Rightarrow \{V_i, DO_j, R_j\}, \\ output \Rightarrow \{VSC_i\} \end{bmatrix} \\ where \\ individual\,(V_i, Video)\,, \\ individual\,(DO_j, DomainOntology)\,, \\ individual\,(R_j, Rule)\,, \\ individual\,(VSC_i, VideoSemanticContent)\,. \end{cases} \tag{5.1}$$

Semantic contents are basically object, event and concept instances taking part in video instances.

$$VideoSemanticContent: \begin{cases} \begin{bmatrix} video \Rightarrow \{V_n\},\ objects \Rightarrow \{OI_i\}, \\ events \Rightarrow \{EI_j\},\ concepts \Rightarrow \{CI_k\} \end{bmatrix} \\ where \\ individual\,(V_n, Video)\,, \\ individual\,(OI_i, ObjectInstance)\,, \\ individual\,(EI_j, EventInstance)\,, \\ individual\,(CI_k, ConceptInstance)\,, \\ i,\ j,\ k \geq 0,\ n = 1. \end{cases} \tag{5.2}$$

There are two main steps followed in the automatic semantic content extraction process. The first step is extracting and classifying object instances from important (representative) frames of shots of the video instances. The second step is extracting events and concepts by using domain ontology and rule definitions.

A genetic algorithm based object extraction and classification mechanism is utilized to make object extraction. Details of this process are explained in Section 5.1. A set of procedures are executed to extract semantically meaningful components in the automatic event and concept extraction process. The first semantically meaningful component is spatial relation instances between object instances. Details about the spatial relation calculation and extraction process are given in Section 5.2. Issues related to temporal relation calculations are described in Section 5.3. Each step in the event extraction process is automatic and details of this process is given in Section 5.4. *Concept Component* individual definitions in the domain ontology and extracted object, event and concept instances are utilized in the automatic concept extraction process that is explained in Section 5.5. At the end of this chapter, a RDF based semantic content data model that is utilized to store/access object, event, concept and semi-semantic content instances is described.

5.1 Object Extraction

Extracting objects from videos and finding their categories give a big support to the content based retrieval job. Manual object extraction methods which define the object instances in each video instance manually are inefficient and time consuming. Therefore, many semi-automatic and automatic object extraction methods are proposed to overcome these limitations. In order to meet the object extraction and classification need of this dissertation, a semi-automatic genetic algorithm based object extraction approach presented in [128] is utilized. In this section, brief information about this approach and its adaptation to the dissertation is given.

[128] proposes a mechanism that separates feature extraction from classification and attacks the problem as a categorization problem. N-Cut Image Segmentation [106] is utilized to segment the images to find candidate objects. A Genetic Algorithms (GA) based approach is used to classify candidate objects from image segments where object categories are defined with the Best Representative and Discriminative Feature (BRDF) model. Main components of the extraction process are illustrated in Figure 5.1 together with the utility components used for feature value normalization and feature importance determination issues.

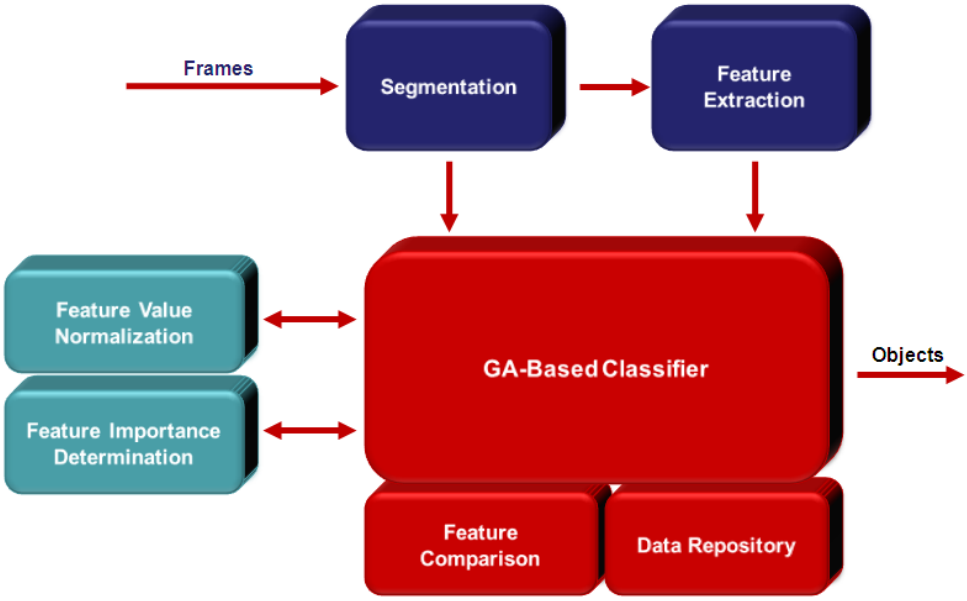


Figure 5.1: Object Extraction Components

Normalized-Cut Segmentation algorithm performs over-segmentation on images as a result of graph partitioning, which mostly yields objects or parts of objects. These segments are then combined to obtain objects. The process mainly aims at finding the maximum number of segments, each of which is different from each other, consistent in some manner and a part of some object. This is achieved by unifying segments into objects according to their representative, calculated according to the similarities of images with the same class, and discriminative, calculated according to the ability of features to distinguish between different object classes, features which are MPEG-7 descriptors [77] extracted from segments. MPEG-7 descriptors such as color, texture and shape descriptors are used as low-level visual features. These features are extracted by using the official software of MPEG, eXperimentation Model (MPEG-7 Reference Software) [85]. [128] models the fact that each visual feature has a different amount of effect in representing the object categories. While, for one category, color distribution and homogeneous texture are important, for another category edge histogram can be more important.

During the object extraction process, each segment is firstly assumed to be an object and tried to be classified by the GA based object classifier. In the second round, segments are tried to be unified with their neighbor segments. Then, the classifier tries to classify new object candidates. In each round, segments are unified and the classifier tries to classify new candidates. Instead of using the average values of the features from MPEG-7 descriptors of samples in the definition of object types (object classes); a set of feature values are stored

and a genetic algorithm mechanism is used to make the set more qualified. At each step, the classifier returns a membership value between 0 and 1 that represents the relevance of the extracted object to the object type it is classified. Extracted objects having a membership value lower than a predefined threshold value are discarded.

For each representative frame in shots, the described object extraction process is performed and a set of objects are extracted and classified. The extracted object instances are stored with their type, frame number, membership value and Minimum Bounding Rectangle (MBR) data that contains upper left corner point and edge lengths of a rectangle that bounds the extracted object. Object instances are used as input with the domain ontologies in the event and concept extraction process.

$$MemberShip : \begin{cases} \left[\ \mu \Rightarrow [float] \ \right] \\ where \\ 0 \leq \mu \leq 1. \end{cases} \tag{5.3}$$

$$MBR : \begin{cases} \left[\begin{array}{l} x \Rightarrow [integer] \, , \ y \Rightarrow [integer] \, , \\ width \Rightarrow [integer] \, , \ length \Rightarrow [integer] \end{array} \right] \end{cases} \tag{5.4}$$

$$ObjectInstance : \begin{cases} \left[\begin{array}{l} frameNo \Rightarrow [number] \, , \\ minumumBoundingRectangle \Rightarrow \{MBR_i\}, \\ membership \Rightarrow \{MSV_j\}, \ objectType \Rightarrow \{O_k\} \end{array} \right] \\ where \\ individual \left(O_k, Object\right) , \ individual \left(MBR_i, MBR\right) , \\ individual \left(MSV_j, MemberShip\right) . \end{cases} \tag{5.5}$$

5.2 Spatial Relation Extraction

Object instances show an irregular nature in terms of shape, which makes the spatial relation extraction process complex. In order to simplify the calculation process, objects are represented with the Minimum Bounding Rectangles (MBR) that surround the segment groups classified as objects. There can be n object instance (as regions) represented with R in a frame F, where $F = \{R_0, \ldots, R_n\}$. For each R, the upper left-hand corner point represented with P_{ul}, length and width of R are stored. The area inside R_i is represented with R_i^{α} where the edges of R_i are represented with R_i^{β}.

Every spatial relation extraction is stored as a *Spatial Relation Component* instance which contains the frame number, object instances, type of the spatial relation and a membership

value of the relation. A *Spatial Relation Component* instance is formally represented as:

$$SpatialRelationInstance: \begin{cases} \left[\begin{array}{l} object \Rightarrow \{O_i\},\ subject \Rightarrow \{S_j\}, \\ relationType \Rightarrow \{R_k\},\ frameNo \Rightarrow [number], \\ membership \Rightarrow \{MSV_m\} \end{array}\right] \\ where \\ individual\,(O_i, ObjectInstance), \\ individual\,(S_j, ObjectInstance), \\ individual\,(R_k, SpatialRelation), \\ individual\,(MSV_m, MemberShip). \end{cases} \tag{5.6}$$

Spatial relations are fuzzy relations and membership values for each relation type can be calculated according to the positions of objects relative to each other. Below, we explain how membership values $(\mu_{dis}, \mu_{top}, \mu_{pos})$ for each of the distance, topological and positional relation categories are calculated.

5.2.1 Topological Relations

Topological relation types are illustrated in Figure 5.2. The membership values for the topological relation types are calculated by using the Equation 5.7.

$$\mu_{top}(R_i, R_k) = \frac{(R_i^\alpha \cap R_k^\alpha)}{R_k^\alpha} \text{ [41]} \tag{5.7}$$

$\mu_{top}(R_i, R_k) = 1$ means region R_k is ***inside*** region R_i. $\mu_{top}(R_i, R_k) = 0 \wedge (R_i^\beta \cap R_k^\beta) = \emptyset$ means region R_k is ***disjoint*** with region R_i. $0 < \mu_{top}(R_i, R_k) < 1$ means region R_k is ***partially inside*** region R_i. $\mu_{top}(R_i, R_k) = 0 \wedge (R_i^\beta \cap R_k^\beta) \neq \emptyset$ means region R_k ***touches*** region R_i.

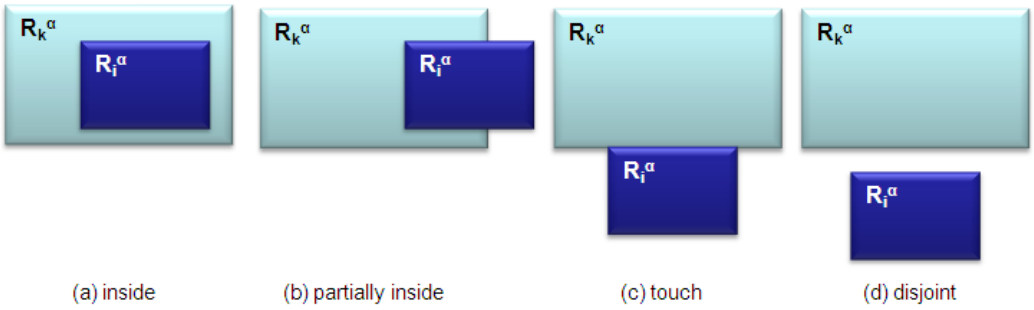


Figure 5.2: Topological Relation Types

5.2.2 Distance Relations

There are two distance relation types as *far* and *near*. The distance between the center points of the regions is utilized in order to calculate the distance relation membership values. When two regions have a *inside*, *partially inside* or *touch* topological relation, the distance relation membership values are directly assigned as $\mu_{far}(R_i, R_k) = 0$ and $\mu_{near}(R_i, R_k) = 1$. When there is a *disjoint* topological relation, using the distance between the center points of the regions sometimes causes problems. When there is one or two big sized regions, even regions are very close to each other, the center points can place far from each other. Such a situation is given with an example in Figure 5.3. To overcome this problem, the distance between two nearest points of regions are used in the calculation formulas of μ_{far} and μ_{near}.

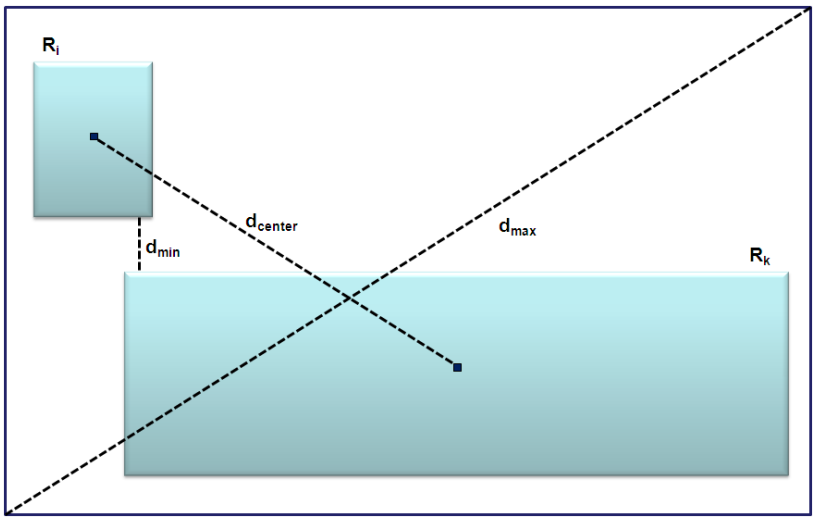


Figure 5.3: Distance Relation

In Figure 5.4, the membership function graphs of μ_{far} and μ_{near} are given. d represents the distance between two nearest points of region R_i and region R_k. A distance relation is extracted as *near* when $\mu_{near} > 0.5$ otherwise it is extracted as *far*.

5.2.3 Positional Relations

μ_{pos} is calculated as μ_{pos}^{above}, μ_{pos}^{below}, μ_{pos}^{left}, μ_{pos}^{right} values for each positional relation type. Center points of regions are used to calculate membership values as most of the studies such as [41, 116, 61] do. The center point of one of the regions is fixed as origin (0,0). The sinus of the angle(Φ) between the x coordinate and the line between two center points of regions

is calculated. This value is used to calculate μ_{pos}s with the following formulas:

$$\mu_{pos}^{right}(R_i, R_k) = \begin{cases} sin(\Phi + 90), & 0 < \Phi < 90 \vee 270 < \Phi < 360 \\ 0, & otherwise \end{cases} \tag{5.8}$$

$$\mu_{pos}^{left}(R_i, R_k) = \begin{cases} sin(\Phi - 90), & 90 < \Phi < 270 \\ 0, & otherwise \end{cases} \tag{5.9}$$

$$\mu_{pos}^{above}(R_i, R_k) = \begin{cases} sin(\Phi), & 0 < \Phi < 180 \\ 0, & otherwise \end{cases} \tag{5.10}$$

$$\mu_{pos}^{below}(R_i, R_k) = \begin{cases} sin(\Phi - 180), & 180 < \Phi < 360 \\ 0, & otherwise \end{cases} \tag{5.11}$$

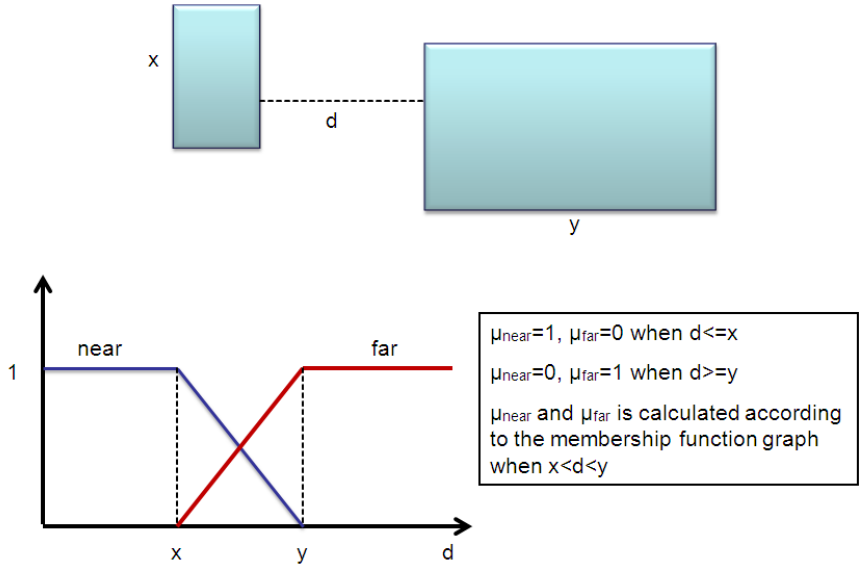


Figure 5.4: Distance Relation Membership Function Graph

In order to decrease the calculation costs, we divide each frame into nine parts by extending the edges of the first region as given in Figure 5.5. μ values are calculated according to the placement of the center point of the reference region. When the center point of the reference region places in parts 2, 4, 6 and 8, we assign $\mu_{pos}^{above} = 1$, $\mu_{pos}^{left} = 1$, $\mu_{pos}^{right} = 1$ and $\mu_{pos}^{below} = 1$ respectively. For example, in Figure 5.5, the center point of region B places in part 2 where $\mu_{pos}^{above} = 1$ and others are set to zero. For region C, center of gravity point places in part 6 where only $\mu_{pos}^{right} = 1$. For region D, center of gravity point places in part 7 where μ_{pos}^{below} and μ_{pos}^{left} is calculated according to the formulas given in this section. Part

5 means one region is in front of another region where we handle this situation with the topological relation *partiallyInside*.

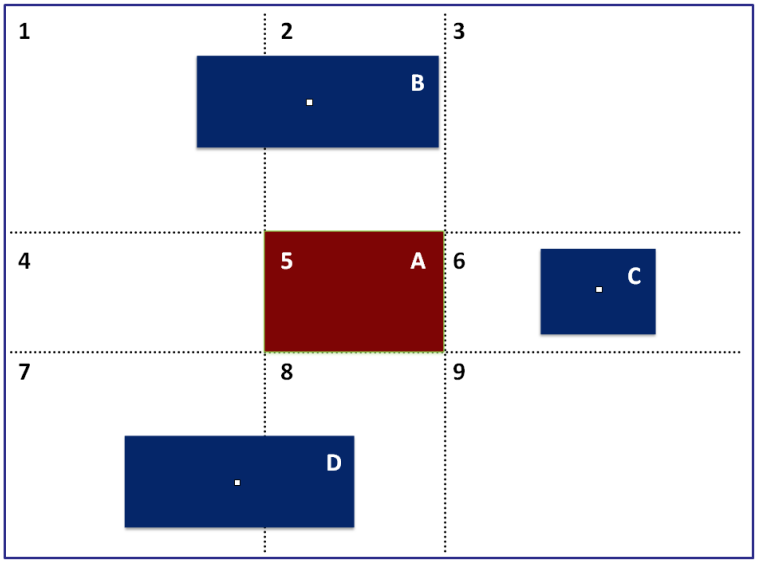


Figure 5.5: Positional Relation Calculation

More spatial relations as *Just Above, Just Below, Near Left, Near Right* can be represented with VISCOM. For instance, to define a *Just Above* spatial relation between two objects, an *Above* and a *Near* relation definition is made as a *Spatial Relation Component* individual. The membership value of a *Spatial Relation Component* instance having a *Just Above* relation is calculated by taking the minimum of the membership values of *Above* and *Near* relations ($min(\mu_{pos}^{above}, \mu_{near})$).

Rule definitions are also utilized in order to lower spatial relation computation costs. Whenever a spatial relation individual which is defined in the body part of a rule definition is extracted, the rule is executed and the spatial relation individual defined in the head part of the rule definition is directly extracted. In Chapter 4, rule examples for positional relation *below* and distance relation *near* are given.

5.3 Temporal Relation Extraction

Temporal representation and reasoning is an important facet in the design of video content models. In this framework, temporal relations are utilized in order to add temporality to sequence *Spatial Change* or *Events* individuals in the definition of *Event* individuals.

One of the well-known formalisms proposed for temporal reasoning is Allen's temporal interval algebra [16] which describes a temporal representation that takes the notion of a temporal interval as primitive. Allen's algebra is used to express parallelism and mutual exclusion between model components of VISCOM. Allen defined a set of 13 qualitative relations that may hold between two intervals $X = [x^-, x^+]$ and $Y = [y^-, y^+]$. Table 5.1 shows how Allen expressed these precise relations by means of constraints on the boundaries of the crisp intervals ($X = [x^-, x^+]$ and $Y = [y^-, y^+]$) involved.

Because the first seven relation types are enough to represent all other types, they are defined as temporal relation individuals in VISCOM. The formulas given in the definition column of Table 5.1 are used to extract temporal relations between instances.

Table 5.1: Allen's Temporal Interval Relations

Name	Notation	Definition
before	b(X,Y)	$x^+ < y^-$
overlaps	o(X,Y)	$x^- < y^-$ and $y^- < x^+$ and $x^+ < y^+$
during	d(X,Y)	$y^- < x^-$ and $x^+ < y^+$
meets	m(X,Y)	$x^+ = y^-$
starts	s(X,Y)	$x^- = y^-$ and $x^+ < y^+$
finishes	f(X,Y)	$x^+ = y^+$ and $y^- < x^-$
equals	e(X,Y)	$x^- = y^-$ and $x^+ = y^+$
after	bi(X,Y)	b(Y,X)
overlapped-by	oi(X,Y)	o(Y,X)
contains	di(X,Y)	d(Y,X)
met-by	mi(X,Y)	m(Y,X)
started-by	si(X,Y)	s(Y,X)
finished-by	fi(X,Y)	f(Y,X)

5.4 Event Extraction

Event instances are extracted after a sequence of extraction processes. Each extraction process outputs instances of a semantic content type defined as an individual in the domain ontology. The first semantically meaningful component is spatial relation instances between

object instances. Spatial relations are calculated and stored as *Spatial Relation Component* instances as described in Section 5.2. The second important semantic component that can be directly extracted from object instances is object movements. Object positions are traced through consecutive frames to find whether they follow a movement pattern. Object movements are stored as *Spatial Movement Component* instances. *Spatial Relation Component* and *Spatial Movement Component* instances are used to extract *Spatial Change* instances by using *Spatial Change* individual definitions in the domain ontology. Similar to the *Spatial Change* instance extraction, *Temporal Spatial Change Component* instance extraction is done by using temporal relations between *Spatial Change* instances and *Temporal Spatial Change Component* individual definitions in the domain ontology. *Event Definition* individuals are defined with *Spatial Change*, *Spatial Relation Component* and *Temporal Spatial Change Component* individuals in the domain ontology. Instances of these classes are utilized in order to extract *Event Definition* instances. *Event Definition* instances or temporal relations between previously extracted *Event* instances which are extracted as *Temporal Event Component* instances are used to extract *Event* instances. The last step in the event extraction process is executing rule definitions related with event individuals. Each step in the event extraction process is automatic. Algorithm 2 simply describes the whole event extraction process. In addition, relations between the extraction processes are illustrated in Figure 5.6.

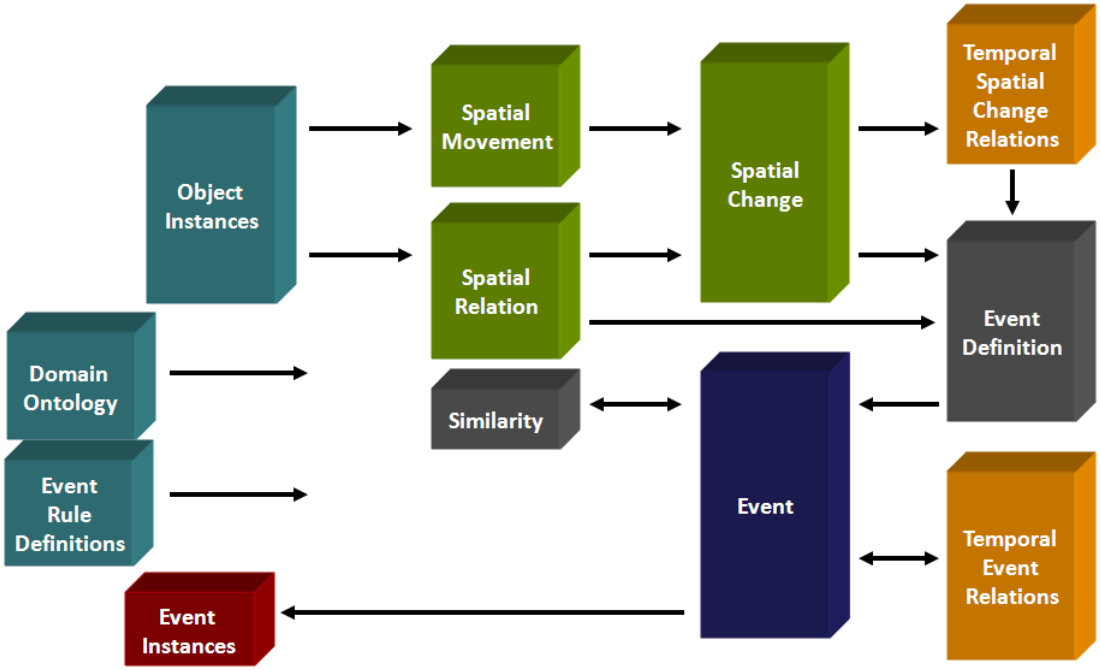


Figure 5.6: Event Extraction Process

Algorithm 2 Event Extraction Algorithm

Input : *Domain Ontology*, *Object Instances*

Output : *Event Instances*

1: **for all** *Spatial Relation Component* individuals in the ontology **do**
2: extract *Spatial Relation Component* instances that satisfy the individual definition.
3: execute *Spatial Relation* rule definitions
4: **end for**
5: **for all** *Spatial Movement Component* individuals in the ontology **do**
6: extract *Spatial Movement Component* instances that satisfy the individual definition.
7: **end for**
8: **for all** *Spatial Change* individuals in the ontology **do**
9: check if there are *Spatial Relation Component* or *Spatial Movement Component* instances that satisfy the individual definition.
10: **end for**
11: **for all** *Temporal Spatial Change* individuals in the ontology **do**
12: extract *Spatial Change* instances that satisfy the individual definition.
13: **end for**
14: **for all** *Event Definition* individuals in the ontology **do**
15: check if there are *Spatial Change*, *Spatial Relation* or *Temporal Spatial Change* instances that satisfy the individual definition.
16: **end for**
17: **for all** *Event* individuals in the ontology **do**
18: check if there are *Event Definition* instances that satisfy the individual definition.
19: **end for**
20: **for all** *Event* individuals which have *Temporal Event Component* individuals **do**
21: extract *Event* instances that satisfy the individual definition.
22: **end for**
23: **for all** *Similarity* individuals in the ontology **do**
24: extract *Event* instances that satisfy the individual definition.
25: **end for**
26: **for all** extracted *Event*, *Event Definition* and *Spatial Change* instances **do**
27: extract *Object Role* instances defined in individual definitions.
28: **end for**
29: execute all rules defined for *Event* individuals to extract additional events.

In order to picturize the event extraction process, screen shots of a free throw event from a basketball video are given in Figure 5.7. In the definition of free throw event, player, ball, free throw line and hoop object individuals and spatial relations between them and object movements are used. Object instances are extracted from each frame given in this figure by using the methodology defined in Section 5.1. In the first frame, there are two important *Spatial Relation Component* instances. The first, the player object stays near to the free throw line object and the second, the ball object places in the right of the player object. In the second frame, the ball object instance is seen above the player object instance where this situation is extracted as a *Spatial Relation Component* instance. Between the first two frames, a *Spatial Change* definition happens between the player and ball object instances. After the player throws ball object, it moves right which can be seen in the third frame. In the fourth frame, this movement becomes more obvious where a *Spatial Movement Component* instance is extracted. In the fifth frame, the ball object is above the hoop object where it is below the hoop object in the last frame. These two *Spatial Relation Component* instances following each other produce another *Spatial Change* instance. In this example, totally, three *Spatial Change* instances (*Spatial Movement Component* individuals are used in *Spatial Change* individual definitions) exist and they happen in the order that is described above. The temporal relations between the *Spatial Change* instances are calculated and a *Temporal Spatial Change Component* instance is extracted which is used in the definition of free throw event. At last, it is used to extract the free throw *Event* instance.

Both the ontology model and the semantic content extraction process is developed considering uncertainty issues. For the semantic content representation, VISCOM ontology introduces fuzzy classes and properties. *Spatial Relation Component*, *Event Definition*, *Similarity*, *Object Composed Of Relation* and *Concept Component* classes are fuzzy classes as they give opportunity to make fuzzy definitions. Object instances have membership values as an attribute which represents the relevance of the given MBR to the object type. Spatial relation calculations return fuzzy results and *Spatial Relation Component* instances are extracted with membership values.

We can describe how fuzzy concepts are handled in the *Spatial Change* instance calculations with an example. It can be assumed that X and Y are objects taking role in the definition of the *Spatial Change* individual *SC*. *SC* is defined in the ontology as: *SC* = (X stands left of Y) followed by (X is above Y). The membership value for the *Spatial Change* individual *SC* is calculated as: $\mu_{SC} = min(\mu_{leftside}(X, Y), \mu_{above}(X, Y))$.

(a) Ball is in the right of the Player

(b) Ball is above the Player

(c) Ball goes right

(d) Ball is far from the Player

(e) Ball is above the Hoop

(f) Ball is below the Hoop

Figure 5.7: Free Throw Event Screen Shots

The predefined relevance value usage such as in *Event Definition* class is another dimension where uncertainty is considered. For instance, an *Event Definition* individual *ED* is extracted with a membership value μ_{ED}. If, *ED* has a relevance value for represent-

ing an event E as μ_{ED_E}, then the membership value of an *Event* E instance is calculated as: $\mu_E = \mu_{ED} * \mu_{ED_E}$

During the extraction process, all of the fuzzy definitions and calculations are considered and the semantic content is extracted with a certainty degree between 0 and 1. At the end of the extraction process, an extracted event instance is represented with a type, a frame set representing the event's interval, a membership value that represents the possibility of the event realization in the extracted event period and the roles of the objects taking part in the event. *Frame Set* is used to represent the frame interval of instances. Formal representation of *Frame Set* and *Event Instance* is given as:

$$FrameSet : \begin{cases} \begin{bmatrix} start \Rightarrow [integer], \ end \Rightarrow [integer], \\ video \Rightarrow \{V_i\} \end{bmatrix} \\ where \\ individual\,(V_i, Video), \ start \neq \oslash, \ end \neq \oslash. \end{cases} \tag{5.12}$$

$$EventInstance : \begin{cases} \begin{bmatrix} frameSet \Rightarrow \{FS_i\}, \ membership \Rightarrow \{MSV_j\}, \\ eventType \Rightarrow \{E_k\}, \ objectRole \Rightarrow \{OR_m\} \end{bmatrix} \\ where \\ individual\,(FS_i, FrameSet), \\ individual\,(MSV_j, MemberShip), \\ individual\,(E_k, Event), \ individual\,(OR_m, ObjectRole). \end{cases} \tag{5.13}$$

5.5 Concept Extraction

In the concept extraction process, *Concept Component* individuals and extracted object, event and concept instances are used. *Concept Component* individuals relate objects, events and concepts with concepts. When an object or event that is used in the definition of a concept individual is extracted, the related concept instance is automatically extracted with the relevance degree given in its definition. In addition, *Similarity* individuals are utilized in order to extract more concepts from the extracted components. The last step in the concept extraction process is executing concept rule definitions.

Concept Extraction Algorithm given as Algorithm 3 simply describes the whole concept extraction process. In addition, relations between the concept extraction processes are illustrated in Figure 5.8.

Algorithm 3 Concept Extraction Algorithm

Input : *Domain Ontology*, *Object Instances*, *Event Instances*

Output : *Event Instances*, *Concept Instances*

1: **for all** *Concept Component* individuals in the ontology **do**
2: check is there are *Object* or *Event* instances that satisfy the individual definition.
3: extract *Object Role* instances defined as *Object Role* individuals.
4: **end for**
5: **for all** *Similarity* individuals in the ontology **do**
6: extract *Concept* instances that satisfy the individual definition.
7: **end for**
8: execute all rules defined for *Concept* individuals.

Similar to the event extraction, concepts are extracted with a membership value between 0 and 1. The following example explains how component membership values are used to calculate concept membership values. *Event* individual E and *Object* individual O are related components with the *Concept* individual C. Event E and Object O have relevance values for representing the concept C as μ_{E_C} and μ_{O_C} respectively. When an event E instance is extracted with a membership value μ_E and an object O instance is extracted with a membership value μ_O, the membership value for concept C instance is calculated with the following equation: $\mu_C = max((\mu_E * \mu_{E_C}), (\mu_O * \mu_{O_C}))$.

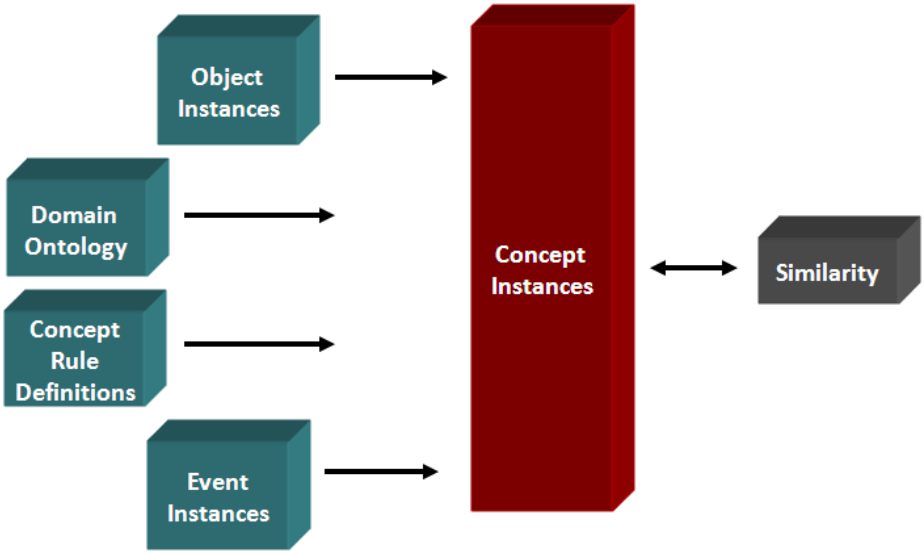


Figure 5.8: Concept Extraction Process

Concept instances use the frame interval of events or objects that are taking part in their definition. A concept has a type, an interval as a frameset, a membership value which represents the possibility of the concept realization in the extracted concept period and the roles of objects taking part in the concept.

$$ConceptInstance: \begin{cases} \left[\begin{array}{l} frameSet \Rightarrow \{FS_i\},\ membership \Rightarrow \{MSV_j\}, \\ conceptType \Rightarrow \{C_k\},\ objectRole \Rightarrow \{OR_m\} \end{array}\right] \\ where \\ individual\left(FS_i, FrameSet\right), \\ individual\left(MSV_j, MemberShip\right), \\ individual\left(C_k, Concept\right), \\ individual\left(OR_m, ObjectRole\right). \end{cases} \tag{5.14}$$

The overall architecture of ASCEF is given in Figure 5.9. After object extraction and classification, the extraction algorithms defined in Section 5.4 and Section 5.5 are applied with relation calculations defined in Section 5.2 and Section 5.3. Spatial, temporal and similarity relations defined in domain ontologies, rule definitions and extracted instances are used together in the semantic extraction process.

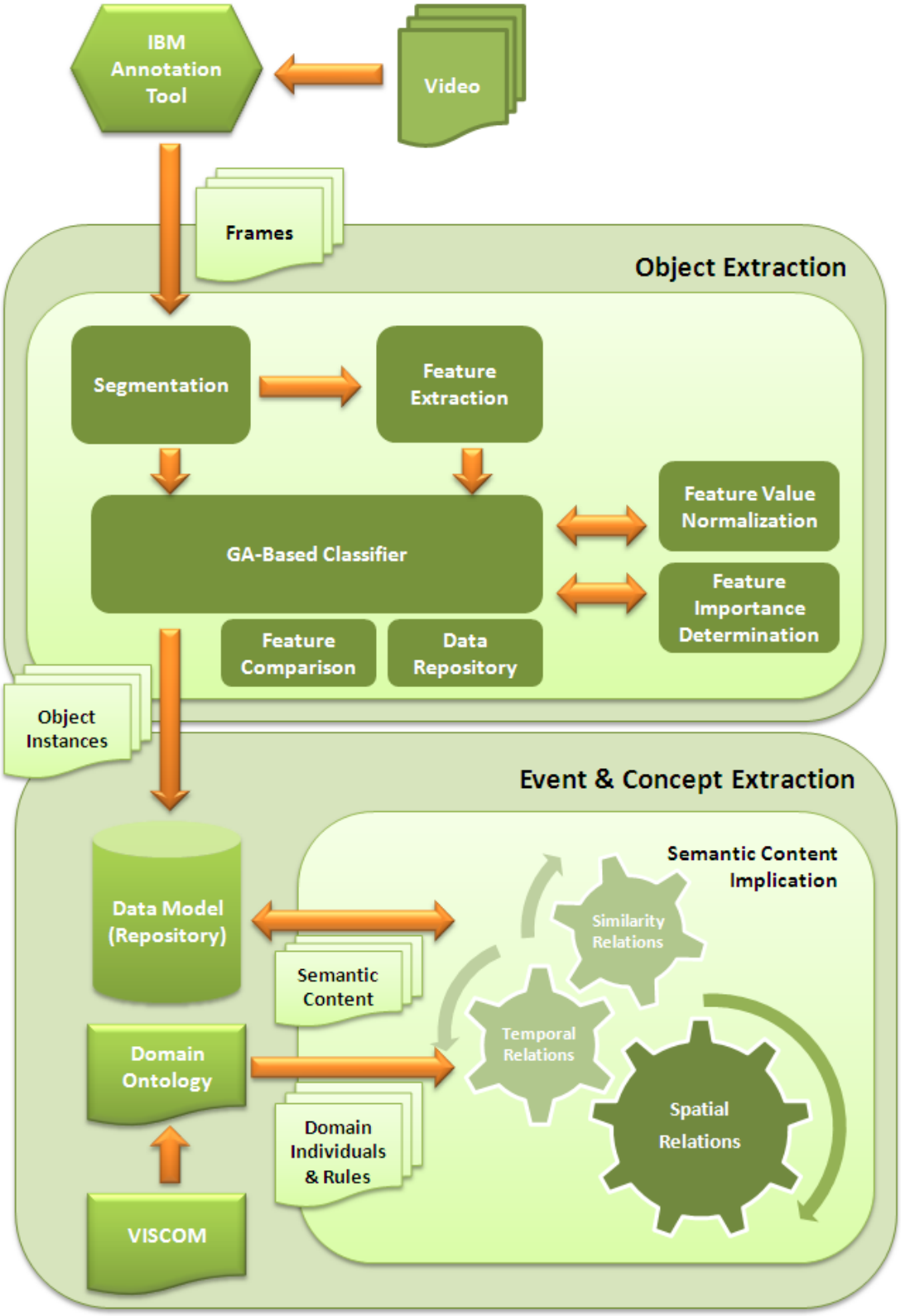


Figure 5.9: Automatic Semantic Content Extraction Framework(ASCEF)

CHAPTER 6

EMPIRICAL STUDY

After describing the entire system, this chapter expresses the empirical studies on the system. Organization of the chapter is as follows: Firstly, brief information about the standards, tools and libraries which are utilized during the implementation is given. Secondly, implementation details and query capabilities of the framework are introduced. Then, the tests performed on the implemented system are given with the results. Lastly, the evaluation of the test results in terms of semantic content extraction for office surveillance and basketball videos are stated.

6.1 Standards, Tools and Libraries

This section is reserved for presenting the standards, tools and libraries utilized in the modeling and extraction phases of the entire system. Starting with the chosen ontology and rule representation languages, ontology editor and management preferences are presented.

Ontologies must be embedded in standard knowledge representation frameworks to exploit available inference engines. In this dissertation, OWL is chosen as the semantic markup language and Semantic Web Rule Language (SWRL)[59] is utilized to make rule definitions.

SWRL rules reason about OWL individuals, primarily in terms of OWL classes and properties. They can also refer explicitly to OWL individuals and support literals and the common same-as and different-from concepts. Many things that cannot be expressed in OWL can be defined easily with SWRL rules. Furthermore, aggregations like count, sum, max, avg can be expressed with SWRL rules.

In order to capture imprecision in rules, a fuzzy extension of SWRL is used. In this extension, OWL individuals include a specification of the *degree* (a truth value between 0 and 1) of confidence with which an individual is an instance of a given class or property.

Protégé [10] platform is utilized as the ontology editor. It is a free, open source ontology editor and knowledge-base framework. The Protégé platform supports two main ways of modeling ontologies. The first one is the Protégé-Frames editor that enables users to build and populate ontologies that are frame-based. The second one is the Protégé-OWL editor which enables users to build ontologies for the Semantic Web, in particular in the W3C's Web Ontology Language (OWL). In this dissertation, Protégé-OWL editor is used to:

- load and save domain ontologies,
- edit and visualize classes, properties, and SWRL rules,
- define logical class characteristics as OWL expressions,
- edit OWL individuals.

Protégé-OWL is tightly integrated with a number of libraries which are supposed to handle ontology deployment and management issues. Jena2 [6] library, which is an open source Java framework for building semantic applications, is used in this study. It provides a programmatic environment for RDF, RDFS, OWL and SPARQL and includes a rule-based inference engine. Through the Ontology API, Jena aims to provide a consistent programming interface for ontology application development, independent of the ontology language. All of the state information remains encoded as RDF triples stored in a RDF model. The ontology API adds a set of convenience classes and methods that make it easier to write programs that manipulate the RDF statements. Thus, domain ontologies are processed by using Jena instructions.

In addition to these, during the object extraction process, I-frames are obtained by using IBM MPEG Annotation Tool [73] that provides facilities to extract shots, keyframes, I-frames from videos.

6.2 Implementation

The system is aimed to be platform independent. To provide platform independency, the main flow of the system and the modules are implemented in Java. Because a convenient system should provide the user a single entry point to perform all of the processes as a single process for simplicity, it is aimed to hide the relations between operations and also external components. Concretely, in the expected system, the user only gives the video instance, the corresponding domain ontology and rule definitions to the system and gets the results.

The implemented system can be analyzed in two parts; Functional Application and Browser Based Graphical User Interface (GUI). Functional application is the core of the system that performs all operations like object extraction, event extraction, concept extraction and rule executions. A graphical user interface is prepared for all operations to be controlled by the user. The GUI provides the following two capabilities; performing semantic content extraction and querying on the extracted content.

6.2.1 Extraction Module

The extraction module of GUI provides the following capabilities; selecting input video files, I-frame folders, domain ontology files, rule definition files and VISCOM OWL file, importing object instances of a video, executing the extraction process and saving the extraction results to the database. A screenshot of the main page of the GUI is given in Figure 6.1.

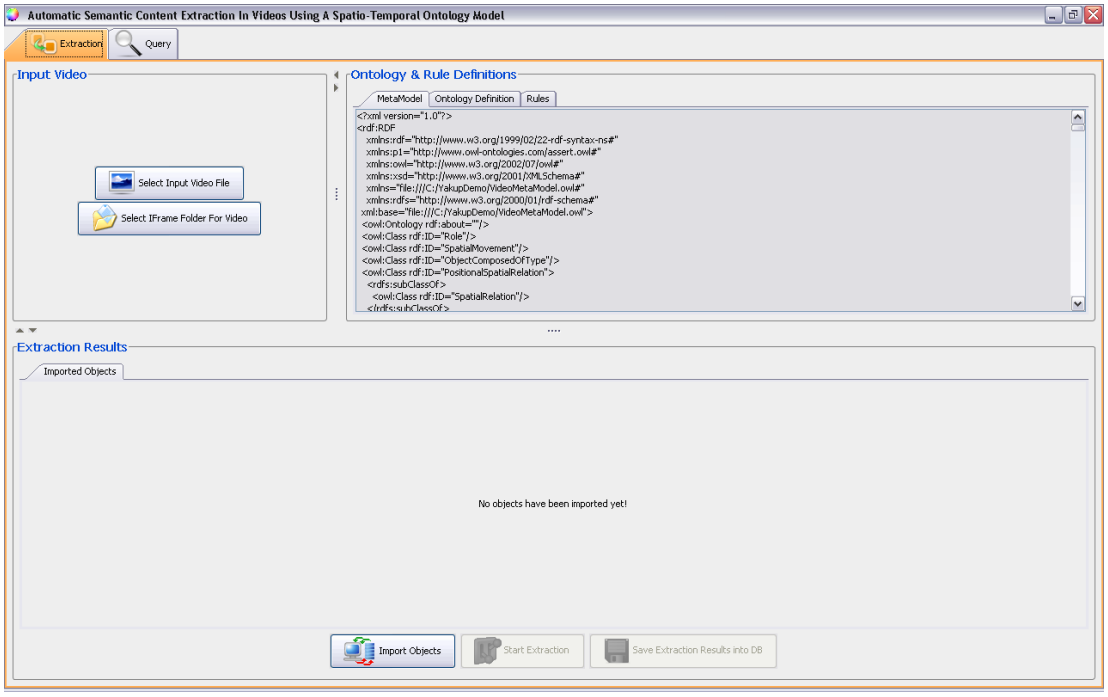


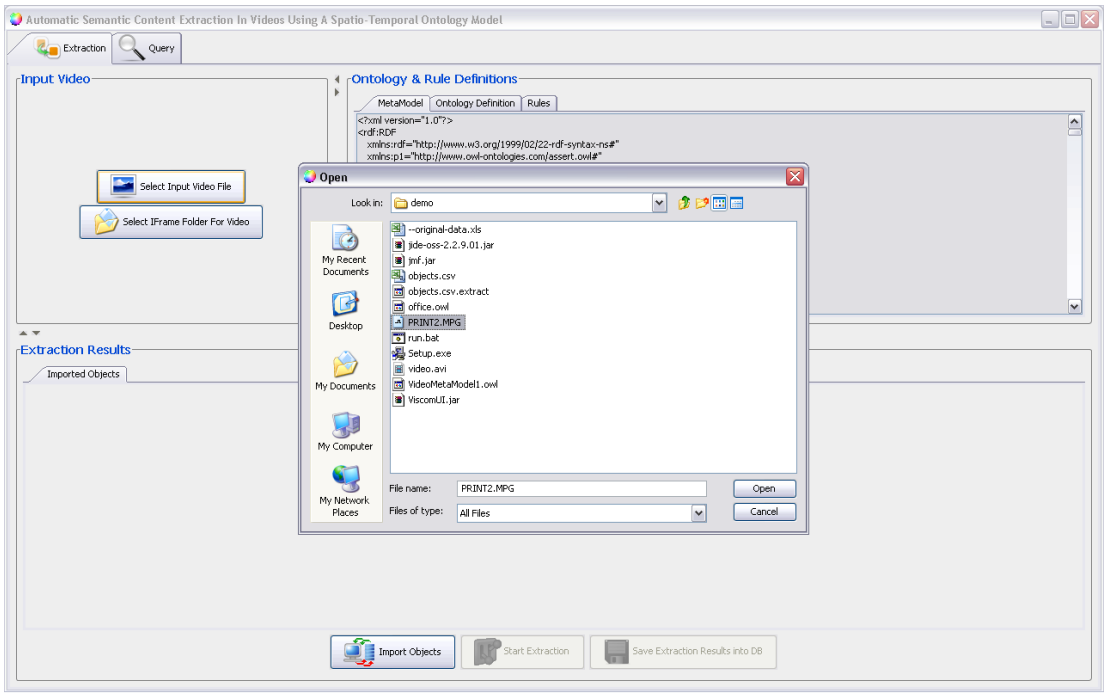
Figure 6.1: Automatic Semantic Content Extraction GUI

The extraction module performs three main functionalities. The first one is importing source files. The second one is importing the metamodel, domain ontology and rule definition files. The third one is activating the extraction process and storing the extracted content to the database.

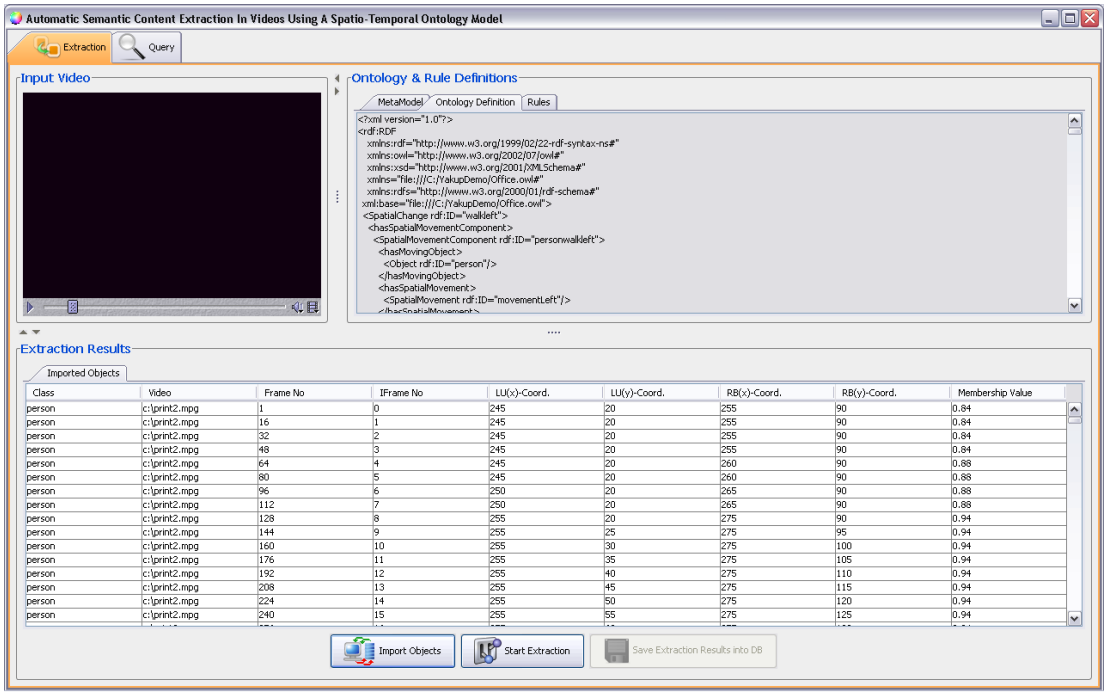
The extraction process starts with selecting a video instance or a folder which contains all I-frames of a video instance. The third alternative for source importing is giving all object instances in a comma-separated file compatible with the format which is accepted by the software. Two screenshots for the source file importing operation are given in Figure 6.2.

Secondly, metamodel and the domain ontology of the selected video instance is imported as OWL files. Additionally, rule definitions are imported in a separate file. Screenshots of the GUI which are captured just after the domain ontology and rule definition import operations are given in Figure 6.3. In another tab, the generic VISCOM OWL file is also displayed.

The extraction process from object extraction to concept extraction is executed by pressing the *Start Extraction* button. All of the extracted content such as objects, events, concepts and all other semantic content instances of VISCOM classes are displayed in the *Extraction Results* part of the screen. Moreover, they can be stored in a database for further query operations. Screenshots of the GUI which are captured during and after the extraction process are given in Figure 6.4.

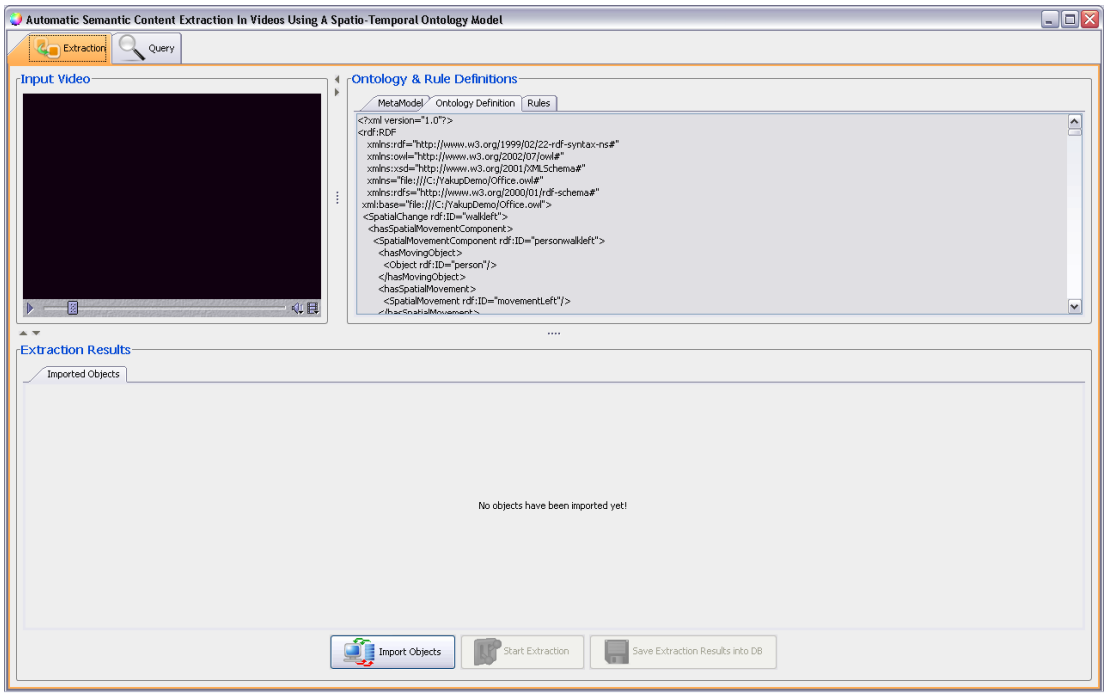


(a) Video File Selection

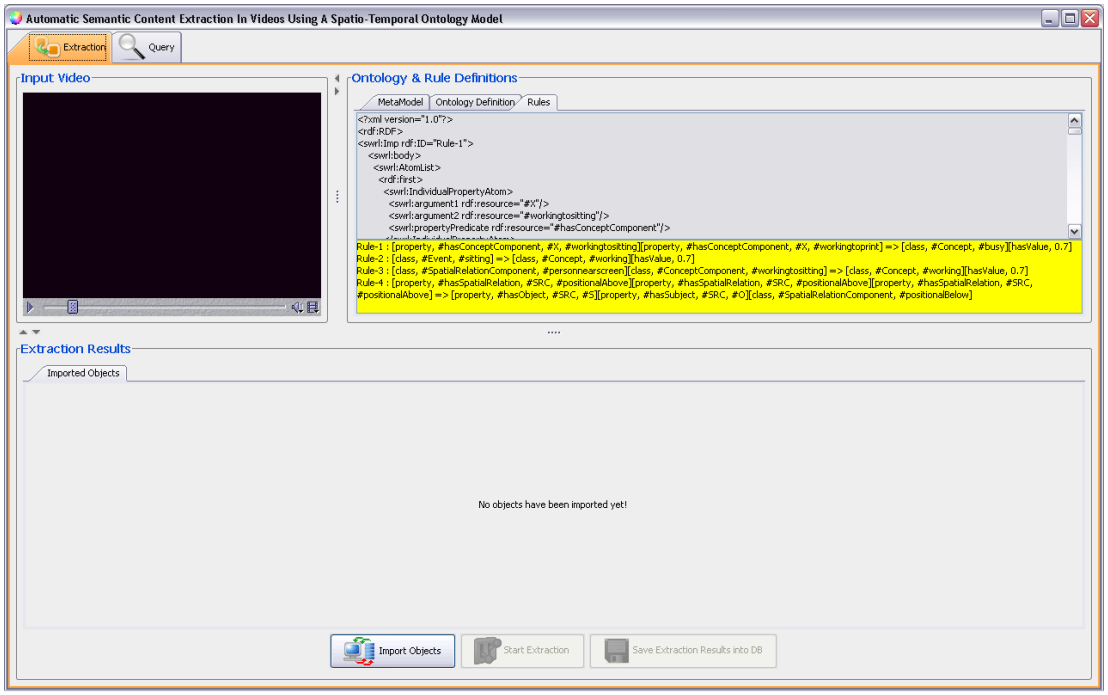


(b) Importing Object Instances

Figure 6.2: Video and Object Instances Import Screenshots

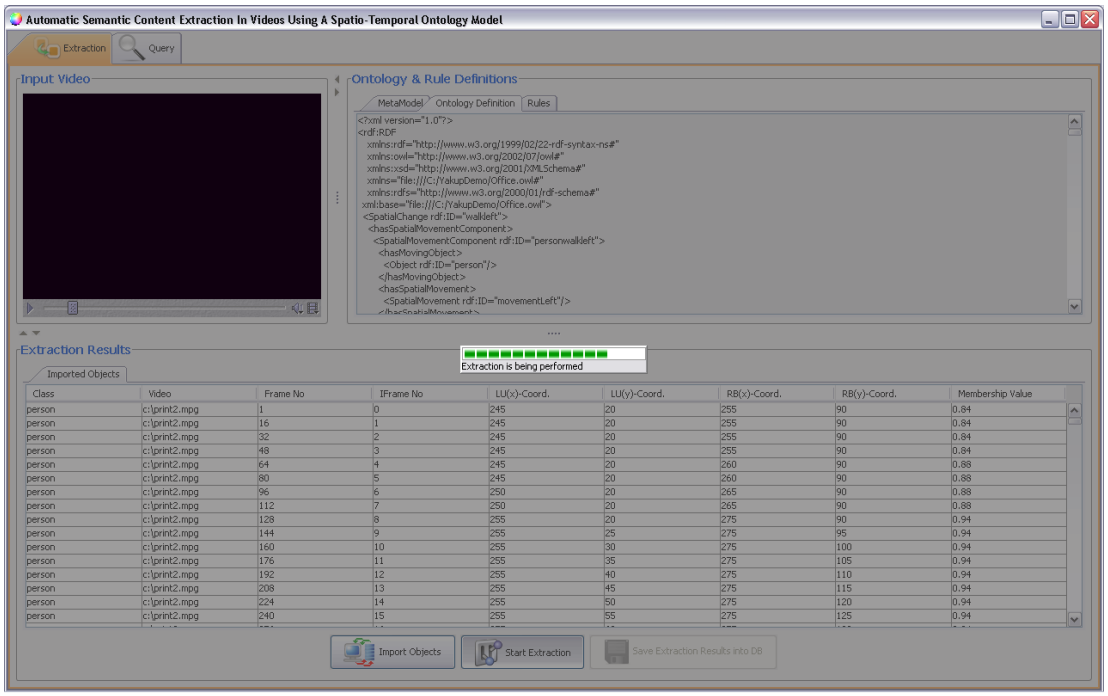


(a) Domain Ontology File Selection

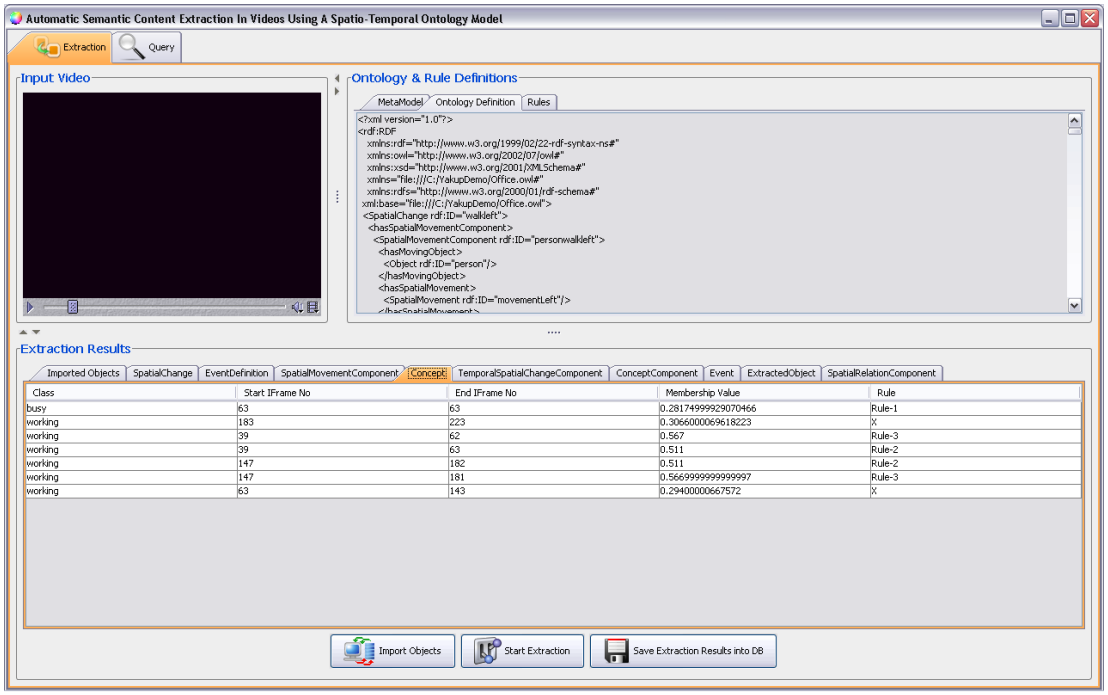


(b) Rule Definition File Selection

Figure 6.3: Domain Ontology and Rule Definition File Selection Screenshots



(a) Semantic Content Extraction Execution



(b) Semantic Content Extraction Results

Figure 6.4: Semantic Content Extraction Screenshots

6.2.2 Query Module

The Query module enables users to search through the extracted semantic content from videos. All of the semantic/semi-semantic content type instances can be stored in a database after the extraction process. By using the query module, users can search the stored content with basically 4 query types as semantic content, spatial relation, temporal relation and trajectory queries.

Semantic Content Query

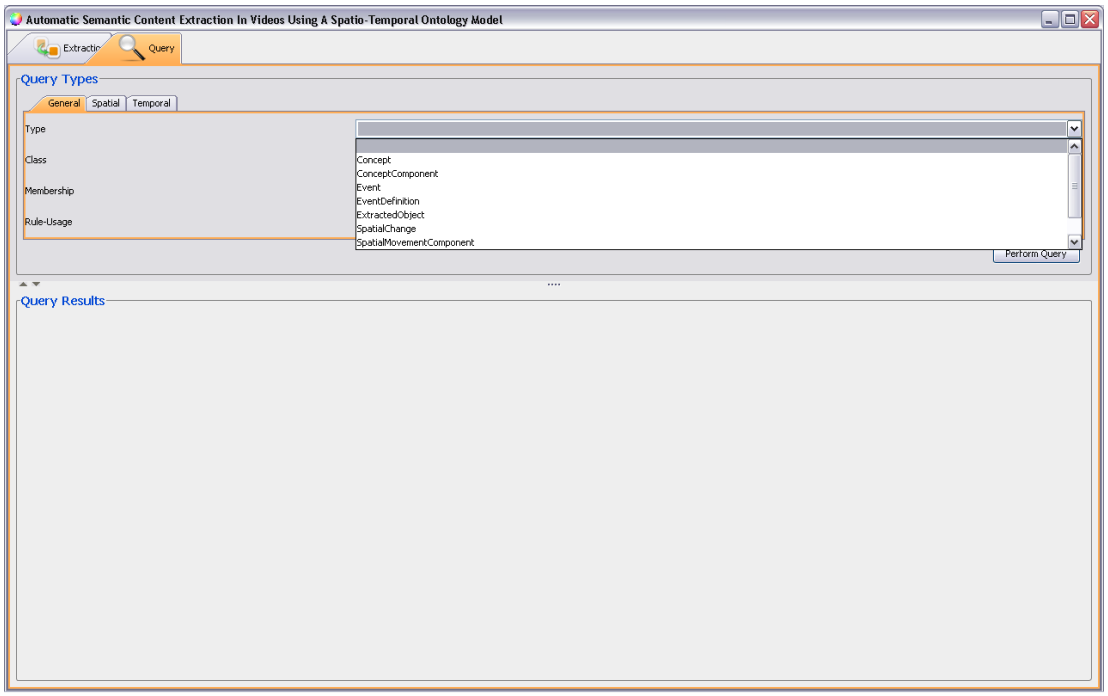
Semantic content query enables users to retrieve instances of all class individuals defined in VISCOM. Object, event, concept, event definition, spatial change and concept component instances can be queried in a video interval, video or video database. All occurrences of the queried content are listed and the related portion of the video can be played. Also, by giving a video interval or video, all of the extracted content within the given source can be retrieved. Two screenshots which are captured from the general semantic content query GUI are given in Figure 6.5. The user can define a value representing the threshold value for the extraction certainty. Below, we give some query examples of this query type:

- Retrieve all *printer* object instances.
- Retrieve all *writing on board* event instances with a threshold value greater than 0,7.
- Retrieve all *working* concept instances.
- Retrieve all of the semantic content in the video.

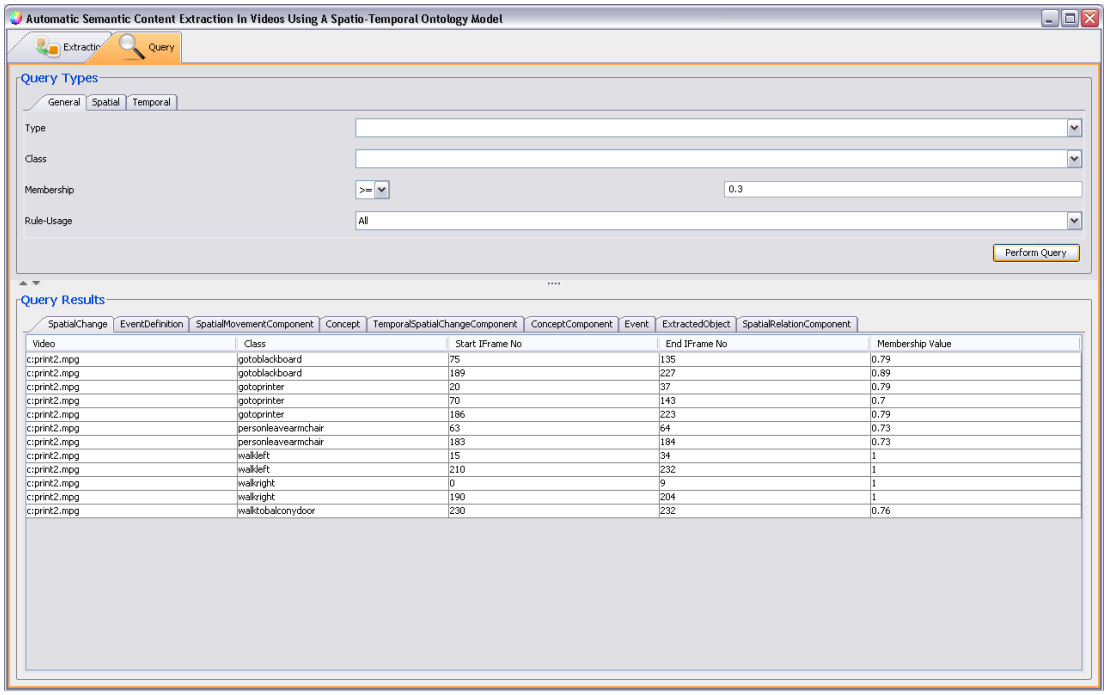
Spatial Relation Query

Spatial relation query enables users to retrieve any spatial relation between objects. In addition, spatial relations between any kind of semantic content individuals such as events and concepts can also be queried. This is achieved by using the object positions taking role in the definition of individuals. A screenshot of the spatial relation query GUI is given in Figure 6.6. Below, we give some query examples of this query type:

- Retrieve all instances where *person* is *near* to the *door* with a threshold value greater than 0,4.
- Retrieve all *writing on board* event instances happening above a *typing* event instance.
- Retrieve all *walking* event instances near the *board* object instance.



(a) Semantic Content Type Selection



(b) Semantic Content Query Results

Figure 6.5: Semantic Content Query Screenshots

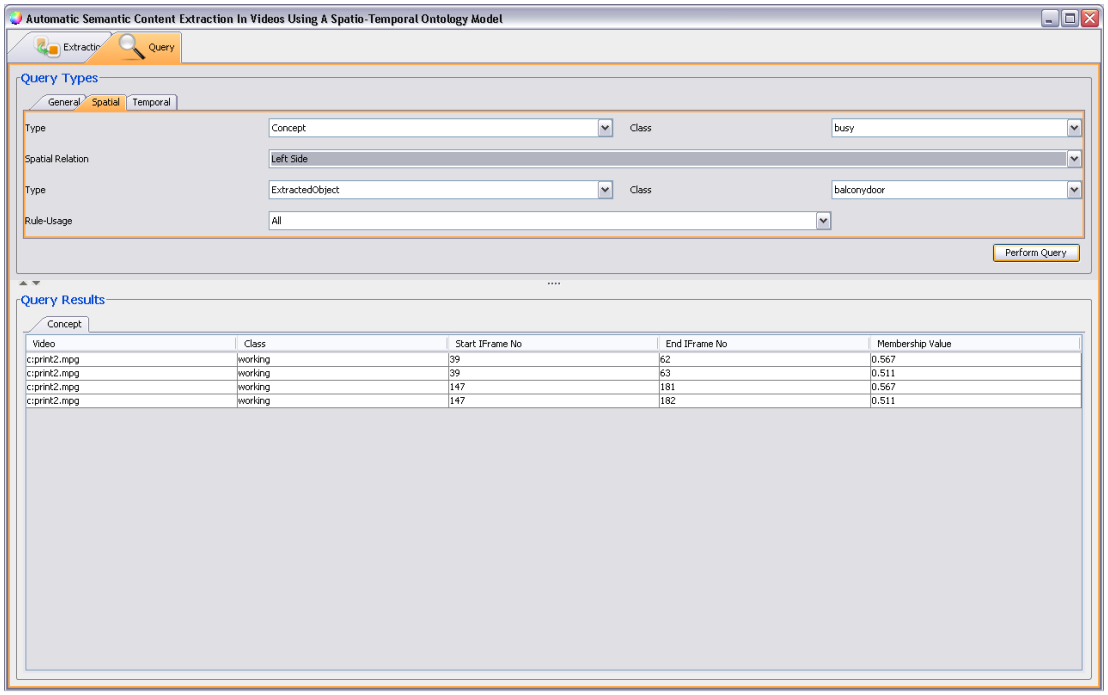


Figure 6.6: Spatial Relation Query Screenshot

Temporal Relation Query

Temporal relation query enables users to query temporal relations between objects, events and concepts. For each extracted instance, its temporal information is stored which enables the user to compare the temporal relations between any kind of the semantic content. A screenshot of the temporal relation query GUI is given in Figure 6.7. Below, we give some query examples of this query type:

- Retrieve all *writing on board* event instances happening before *printing* event instances.
- Retrieve all *printing* events instances happening during a *working* concept instance.

Object Trajectory Query

Object trajectory query enables users to make two types of trajectory queries. First query type uses *Spatial Movement* semantic content type instances extracted during the extraction process. Object movements such as moving left, moving right, moving up and moving down can be queried with this type. This type of query is executed through the general query screen which is given in Figure 6.5. Additionally, object movements which have a destination object in their definition can be queried. In order to query such trajectories, in the spatial query tab, *Spatial Movement Component* is chosen from the '*type*' drop down. After choosing

Spatial Movement Component as the type, any object trajectory can be queried by choosing the moving object from the object list in the '*from*' drop down and the destination object from the '*to*' drop down. A screenshot of the trajectory query GUI designed for this kind of trajectories is given in Figure 6.8. Below, we give some query examples of this query type:

- Retrieve all *person* instances moving left.
- Retrieve all *person* instances moving to a printer instance.
- Retrieve all spatial movements in this video.

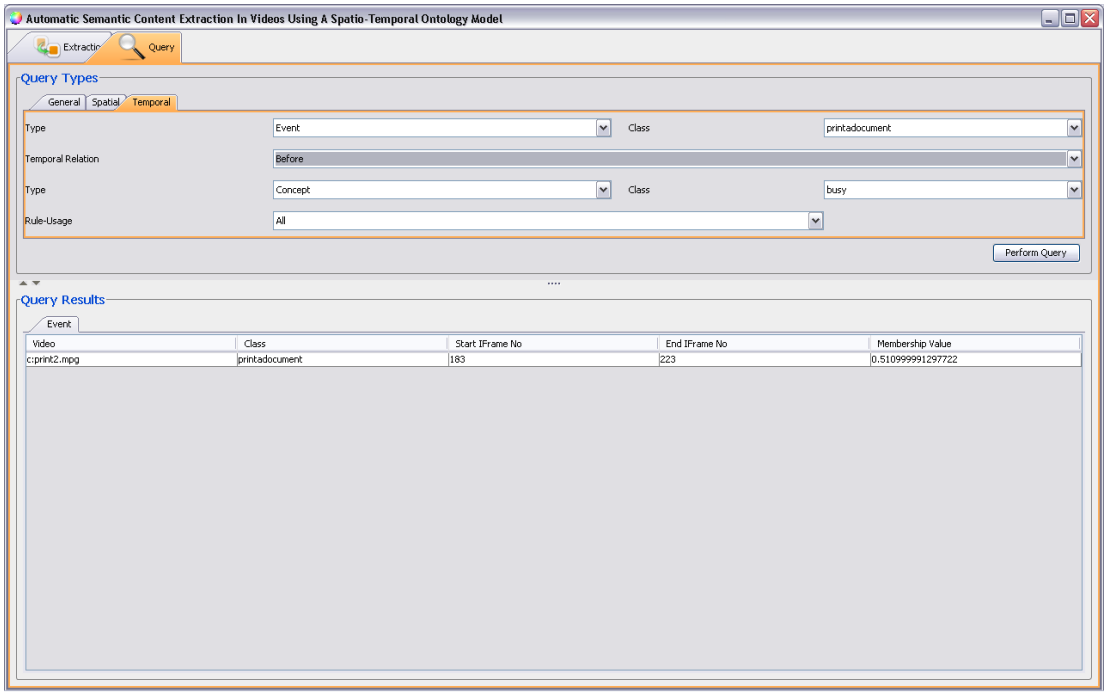


Figure 6.7: Temporal Relation Query Screenshot

6.2.3 Other Facts on Implementation

Other facts on implementation are as follows:

- The implementation is developed on a Windows XP, Centrino Duo 1.66 GHz, 1 GB RAM machine. Also all tests are performed on the same machine.
- The Java implementation is developed with Java Development Kit (JDK) version 1.5.0_9 on IntelliJ IDEA 6.0.4

- The Java implementation contains 5962 lines of code in 32 files.
- As the web server of the application, JBoss AS 4.2.1.GA is used.
- The user interface is developed on Java by using library Google Web Toolkit Version 1.4.59.
- Following Java libraries are used:
  - Apache Jakarta Commons FileUpload Library v1.2, for uploading file into the web server
  - Apache Jakarta Commons IO Library v1.3.2, for file input output with the web server
  - Jakarta Commons Math Library Version 1.1, for statistical calculations
- In the browser based UI, High Performance JavaScript Graphics Library v. 3.01 of Walter Zorn is used.
- For compiling C/C++ files of XM Software, Microsoft Visual Studio 6.0 is used.

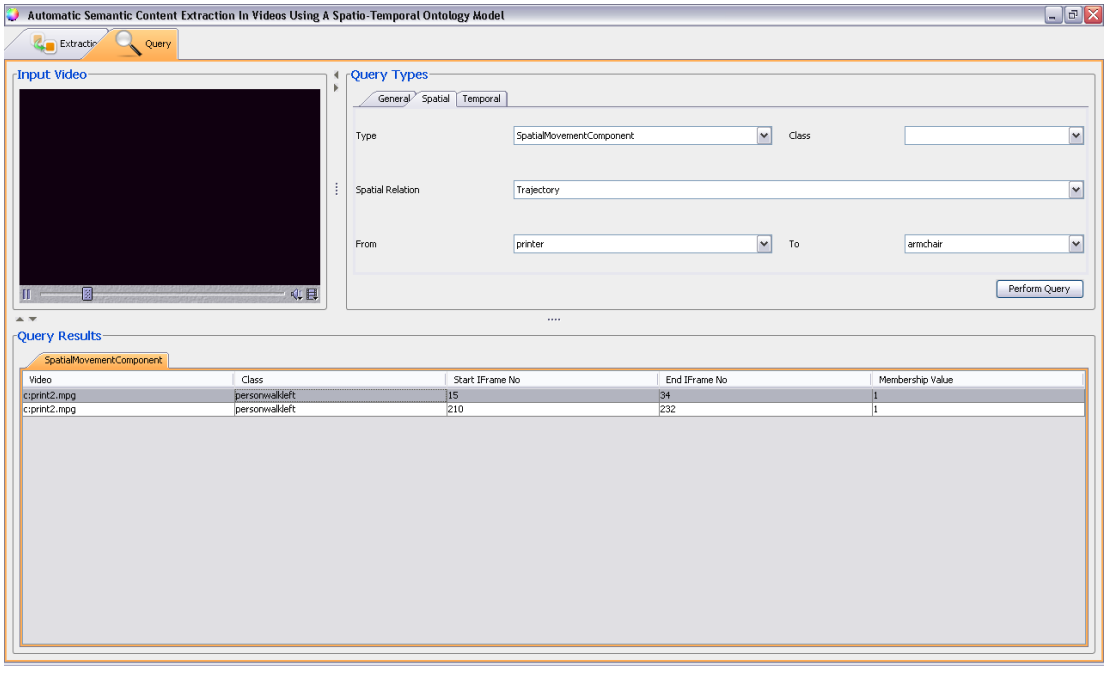


Figure 6.8: Object Trajectory Query Screenshot

6.3 Tests, Results and Evaluation

The experimental part of the system contains tests on office surveillance and basketball videos. In this section, details about the tests, test results, the evaluation done for the performance of the system in terms of semantic content extraction and comparison of the test results with other studies' results are given.

Precision and recall rates and Boundary Detection Accuracy (BDA) [125] score which are important metrics to see the performance of the retrieval systems are utilized to evaluate the success of the proposed framework. A semantic content is accepted as a correctly extracted semantic content when its interval intersects with the manually extracted semantic content interval. In addition, precision and recall rates are calculated according to the detected content boundary/interval compared with the manually labeled boundary/interval with the formulas given below.

$$Prec_{int} = \frac{\tau_{mb} \cap \tau_{db}}{\tau_{db}} \tag{6.1}$$

$$Rec_{int} = \frac{\tau_{mb} \cap \tau_{db}}{\tau_{mb}} \tag{6.2}$$

$$BDA = \frac{\tau_{mb} \cap \tau_{db}}{max(\tau_{mb}, \tau_{db})} \tag{6.3}$$

where τ_{db} and τ_{mb} are the automatically detected event/concept interval and the manually labeled event/concept interval respectively.

Initially, automatic semantic content extraction framework was tested with five 10 minute length office surveillance videos. Totally, 1026 I-frames were extracted and utilized in the extraction process. Object, event and concept instances in the test videos and their number of occurrences are given in Table 6.1.

Three tests were conducted to evaluate the success of the semantic content extraction framework on office surveillance videos. In the first test, object extraction was done automatically and all of the semantic content extraction process explained in this dissertation was executed. Video shots which have single event instances were used during the tests. In the second test, missing or misclassified object instances after the automatic object extraction process were manually extracted or corrected. Same video set which was used in the first test was utilized for this test. In the third test, video shots having multiple semantic content instances were used.

Semantic content name, type of the semantic content, membership value of the semantic content(μ), manually extracted semantic content number, correctly extracted semantic content number, false extraction number, miss extraction number, precision rates, recall rates

and BDA scores for the semantic contents existing in the test videos are given in result tables Table 6.2, Table 6.3 and Table 6.4.

Table 6.1: Semantic Content List for Office Surveillance Ontology

Name	Type	Occurrence	Name	Type	Occurrence
Talk	Concept	1	**Armchair**	Object	2
Work	Concept	5	**Board**	Object	1
Enter the Office	Event	2	**Person**	Object	2
Cast	Event	3	**Room Door**	Object	1
Exit to Balcony	Event	1	**Balcony Door**	Object	1
Print	Event	1	**Screen**	Object	2
Sit	Event	6	**Book**	Object	1
Type	Event	5	**Cabinet**	Object	1
Walk	Event	23	**Table**	Object	2
Welcome	Event	1	**Printer**	Object	1
Write On Board	Event	1	**Tripod**	Object	1
Put Book in Cabinet	Event	1	**Telephone**	Object	1

Results for the first test are given in Table 6.2. Out of 50 semantic content, 45 of them were correctly extracted during this test. When we have examined the data produced during the extraction process, we detected the fact that some of the object instances were misclassified or not extracted with the automatic object extraction process. *Sitting*, *typing*, *exiting to balcony* and *printing* event instances were not extracted because of missing or misclassified object instances. *Welcome* event, has a complex definition which has multiple objects from the same object type. In the definition of this event, two people approach each other and move away after a period. One of the required spatial relation instance for this event was missed by the extraction process which inhibited the extraction of this event.

Moreover, there were five wrong extractions as three *walking*, one *casting* and one *typing* event instances. Object movements were utilized in the definition of *walking* event. According to the object position changes, the conditions defined for walking event were satisfied. But, these movements were not significant movements which can be evaluated as a walking event. In order to detect small movements which were utilized in other event individual

definitions, the actual threshold value which was used in the implementation was proper for movement calculations. However, it caused extra extractions for some event individuals. A similar situation happened for *casting* event. A *typing* event instance was extracted by using the similarity individual definition between *sitting* and *typing* event individuals in the ontology. For this case, there was a sitting event instance but there was no typing event instance.

Both precision and recall rates were calculated as 90.00% and BDA score was calculated as 78.59%, which shows the success of the proposed framework. The higher the BDA score and precision-recall rates are, the better the performance is.

Results for the second test are given in Table 6.3. In this test, we have manually added the object instances which were misclassified or not extracted with the automatic object extraction process. *Sitting*, *typing*, *exiting to balcony* and *printing* event instances which were not extracted during the first test were extracted after this addition. Thereby, recall rate has increased to 98.00% where the only semantic content that could not be extracted was *welcome* event instance. Precision rate and BDA score were slightly increased according to the values obtained in the previous test.

The last test was made with video shots having multiple semantic content instances. Results for this test are given in Table 6.4. All of the semantic content was extracted successfully where recall rate was calculated as 100.00%. Precision and BDA scores showed similar rates with the previous tests.

Test results for basketball domain are given in Table 6.5. Manually annotated object instances were utilized and 0.90 was defined as the membership value for object instances for basketball domain tests. There were four event types and a concept type. Only one *rebound* event instance was not extracted. In the test videos, during the *rebound* event, most of the time, ball instances can not be recognized because of the player instances in the rebound event. Also, a *rebound* event was wrongly extracted because of the *similarity* class individual defined for *rebound* and *jumpball* event individuals.

Test results are also compared with the results of two recent studies. The first one is a multi-modal framework for semantic event extraction from basketball games based on webcasting text and broadcast video in [134]. In this study, an unsupervised clustering based method instead of pre-defined keywords to automatically detect event from web-casting text and a statistical approach instead of finite state machine to detect event boundary in the video are proposed. The second study proposes a method to detect events involving multiple agents in a video and to learn their structure in terms of temporally related chain of sub-

events [52]. The principal assumption made in this work is that the events are composed of highly correlated chain of sub-events. They evaluate their proposal's success with surveillance videos. In Table 6.6, comparisons of the results of this dissertation and mentioned studies are given. Both for basketball and surveillance videos similar or better precision and recall rates and BDA scores were obtained when compared with the results of these studies. The only exception was the rebound event because of the reasons given above.

In order to evaluate the effect of rule usage to the semantic content extraction, two different set of rules were defined. First, positional relation *Below*, positional relation *Left* and distance relation *Near* rules were defined in order to see the effect of rules on spatial relation computation cost. The videos that were used during these tests contain totally 10342 spatial relation instances where 1504 of them are spatial relation instances that have *Below*, *Left* or *Near* as the spatial relation type. The spatial relation instances having these spatial relation types were extracted by using the rule definitions.

Initially, spatial relation computation time was calculated for the case where no rule definition was made. Then, rules were defined one by one and computation times were calculated after each rule definition addition. As it can be seen in Figure 6.9, spatial relation computation times were decreased with the increase in the number of rules definitions. Because rule processing is less costly than spatial relation computation in terms of time, time elapsed during spatial relation computation process was lowered.

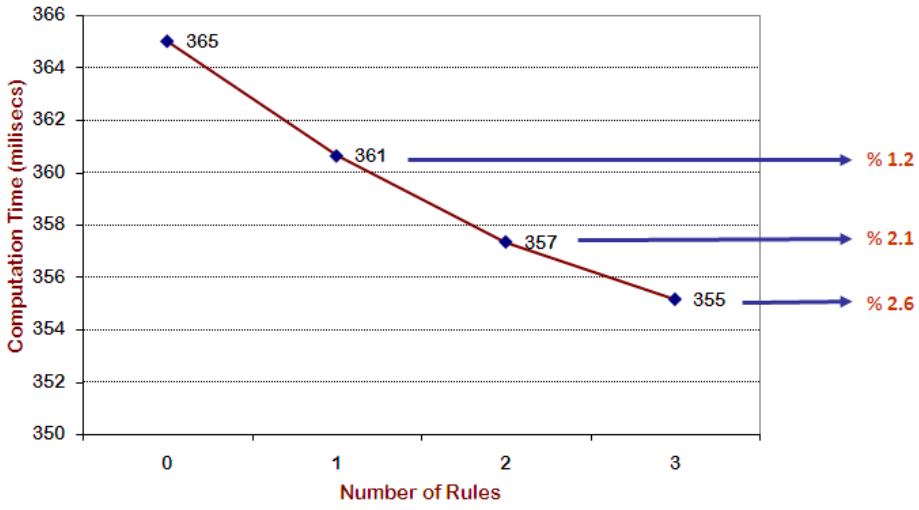


Figure 6.9: Rule Effect on Spatial Relation Computation Cost

As the second test, *working* and *busy* concept rule definitions which are given in Section 4.4 were used in the concept extraction process. Two more concept instances, one working and one busy concept, were extracted by using the rule definitions together with the domain ontology. In this way, the number of extracted event and concept instances can be increased by defining additional rules. Some complex situations such as in the *busy* concept rule definition are easily expressed with rule definitions by using the individuals defined in the domain ontology. In order to define such cases with the representation capabilities of VISCOM, we have to define extra semi-semantic individuals directly related with the *busy* concept. This increases both the execution time of the extraction process and the complexity of the ontology.

All these tests show that the proposed ontology-based automatic semantic content extraction framework is successful for both event and concept extraction. There are two points that must be ensured to achieve this success. The first one is to obtain object instances correctly. Whenever a missing or misclassified object instance occurs in the object instance set that is used by the framework as input, extraction success decreases. The second issue is to use a well and correctly-defined domain ontology. Wrong, extra or missing definitions in the domain ontology also decrease the extraction success. In our tests, we have encountered such cases because of the wrong *Similarity* class individual definitions for *rebound* event in basketball domain and *typing* event in office domain.

Table 6.2: Precision-Recall Values and BDA Scores for Office Surveillance Videos

Name	Type	μ	Manual	Correct	False	Missed	Prec(%)	Recall(%)	$Prec_{int}$(%)	Rec_{int}(%)	BDA(%)
Talk	Concept	0.44	1	1	-	-	100.00	100.00	41.67	100.00	41.67
Work	Concept	0.37	5	5	-	-	100.00	100.00	94.26	98.01	93.36
Enter the Office	Event	0.92	2	2	-	-	100.00	100.00	92.31	100.00	92.31
Cast	Event	0.89	3	3	1	-	75.00	100.00	83.97	100.00	83.97
Exit to Balcony	Event	-	1	-	-	1	0.00	0.00	0.00	0.00	0.00
Print	Event	-	1	-	-	1	0.00	0.00	0.00	0.00	0.00
Sit	Event	0.70	6	5	-	1	100.00	83.33	87.44	69.17	86.33
Type	Event	0.58	5	4	1	1	80.00	80.00	65.80	72.03	65.23
Walk	Event	1.0	23	23	3	-	88.46	100.00	77.23	98.91	76.91
Welcome	Event	-	1	-	-	1	0.00	0.00	0.00	0.00	0.00
Write On Board	Event	0.63	1	1	-	-	100.00	100.00	100.00	96.77	96.77
Put Book in Cabinet	Event	0.59	1	1	-	-	100.00	100.00	95.24	100.00	95.24
Total		0.82	50	45	5	5	90.00	90.00	80.71	80.13	78.59

Table 6.3: Precision-Recall Values and BDA Scores for Office Surveillance Videos (Missing-Misclassified Objects Manually Given)

Name	Type	μ	Manual	Correct	False	Missed	Prec(%)	Recall(%)	$Prec_{int}$(%)	Rec_{int}(%)	BDA(%)
Talk	Concept	0.44	1	1	-	-	100.00	100.00	41.67	100.00	41.67
Work	Concept	0.37	5	5	-	-	100.00	100.00	94.26	98.01	93.36
Enter the Office	Event	0.92	2	2	-	-	100.00	100.00	92.31	100.00	92.31
Cast	Event	0.90	3	3	1	-	75.00	100.00	83.97	100.00	83.97
Exit to Balcony	Event	0.61	1	1	-	-	100.00	100.00	97.22	100.00	97.22
Print	Event	0.51	1	1	-	-	100.00	100.00	97.50	97.50	97.50
Sit	Event	0.64	6	6	-	-	100.00	100.00	87.01	98.63	86.23
Type	Event	0.54	5	5	1	-	83.33	100.00	63.14	99.05	62.76
Walk	Event	1.0	23	23	3	-	88.46	100.00	77.23	98.91	76.91
Welcome	Event	-	1	-	-	1	0.00	0.00	0.00	0.00	0.00
Write On Board	Event	0.63	1	1	-	-	100.00	100.00	100.00	96.77	96.77
Put Book in Cabinet	Event	0.59	1	1	-	-	100.00	100.00	95.24	100.00	95.24
Total		0.80	50	49	5	1	90.74	98.00	80.39	94.32	79.90

Table 6.4: Precision-Recall Values and BDA Scores for Office Surveillance Videos (Multiple Events In Every Shot)

Name	Type	μ	Manual	Correct	False	Missed	Prec(%)	Recall(%)	$Prec_{int}$(%)	Rec_{int}(%)	BDA(%)
Talk	Concept	0.46	2	2	1	-	66.67	100.00	42.22	100.00	42.22
Work	Concept	0.33	4	4	1	-	80.00	100.00	76.19	100.00	76.19
Enter the Office	Event	0.92	3	3	-	-	100.00	100.00	92.59	100.00	92.59
Print	Event	0.51	2	2	-	-	100.00	100.00	95.23	100.00	95.23
Sit	Event	0.73	3	3	-	-	100.00	100.00	95.23	100.00	95.23
Type	Event	0.58	2	2	1	-	66.67	100.00	47.62	100.00	47.62
Walk	Event	1.0	10	10	-	-	100.00	100.00	89.61	97.40	87.34
Write On Board	Event	0.65	3	3	-	-	100.00	100.00	96.67	100.00	96.67
Total		0.75	29	29	3	-	90.62	100.00	77.62	99.24	77.62

Table 6.5: Precision-Recall Values and BDA Scores for Basketball Videos

Name	Type	μ	Manual	Correct	False	Missed	Prec(%)	Recall(%)	$Prec_{int}$(%)	Rec_{int}(%)	BDA(%)
Rebound	Event	0.66	2	1	1	1	50.00	50.00	62.50	55.56	62.50
Jump Ball	Event	0.73	2	2	-	-	100.00	100.00	94.23	100.00	97.23
Free Throw	Event	0.72	2	2	-	-	100.00	100.00	95.00	100.00	95.00
Attack	Concept	0.65	2	2	-	-	100.00	100.00	94.65	100.00	94.65
Total		0.69	8	7	1	1	87.50	87.50	88.42	92.36	89.34

Table 6.6: Comparison with Recent Semantic Content Extraction Studies

	Domain	Methodology	Relations	Precision(%)	Recall(%)	BDA(%)
Hakeem	Surveillance	Sub-event Graphs	Temporal Relations	86.70	87.00	-
Our Study	Surveillance	Ontology	Spatial/Temporal Relations	90.00	90.00	78.56
Zhang	Basketball-Rebound	Web-casting Text	Text-video Alignment	98.50	99.2	90.5
Our Study	Basketball-Rebound	Ontology	Spatial/Temporal Relations	62.50	55.56	62.50
Zhang	Basketball-FreeThrow	Web-casting Text	Text-video Alignment	100.00	100.00	83.00
Our Study	Basketball-FreeThrow	Ontology	Spatial/Temporal Relations	95.00	100.00	95.00
Zhang	Basketball-Jumper	Web-casting Text	Text-video Alignment	89.30	100.00	93.5
Our Study	Basketball-Jumpball	Ontology	Spatial/Temporal Relations	94.23	100.00	97.23

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

The primary aim of this research was to develop a framework as an automatic semantic content extraction system for videos. The innovative idea is to utilize domain ontologies generated with a domain independent ontology-based semantic content meta model and a set of rule definitions.

Ontology-based semantic meta model, VISCOM, uses objects and spatial/temporal relations between objects in event and concept definitions. This enables VISCOM to model events and concepts related with other objects and events. Because VISCOM is domain independent, classes and relations are generic and functional for any domain. Domain ontologies are generated by defining domain specific components such as objects, events and concepts as individuals of VISCOM classes.

In addition to this, rule definitions are utilized to strengthen the modeling capabilities of the meta model. Rule definitions are used in order to be able to define complex situations more effectively. They are also utilized to lower spatial relation calculation costs.

In order to achieve the semantic content extraction goal, video instances are processed through a set of extraction processes. Domain ontologies and rule definitions are utilized during these processes as inference mechanisms. First, object instances are extracted and classified from important (representative) frames with a genetic algorithm based object extraction and classification mechanism. Then, spatial relations between object instances and temporal relations between semantically meaningful VISCOM class instances are extracted in order to find instances of event and concept individuals defined in the domain ontology.

Automatic Semantic Content Extraction Framework, ASCEF, contributes in a number of ways to semantic video modeling and semantic content extraction research area. First of all,

all of the semantic content extraction process is done in an automatic manner. In addition, a generic ontology-based semantic meta model for videos is proposed. Moreover, the semantic content representation capability and extraction success are improved by adding fuzziness in class, relation and rule definitions. An automatic genetic algorithms based object extraction method is integrated to the proposed system to capture all of the semantic content types. In every process of the framework, ontology-based modeling and extraction capabilities are utilized.

As an empirical study, we have performed a number of experiments for event and concept extraction in basketball and office surveillance videos. We have obtained satisfactory precision and recall rates and BDA scores in terms of object, event and concept extraction. There are two points that must be ensured to achieve this success. The first one is obtaining object instances correctly. The second issue is using a well and correctly constructed domain ontology.

A platform and domain independent application for the proposed system has also been implemented. Throughout the experiments by using the implemented application, the proposed system achieved better performances compared to the other semantic content extraction approaches. Furthermore, the test results clearly showed the success of the framework.

The model and semantic content extraction solution can be utilized in various areas, such as surveillance, sports and news video applications. The following issues are research issues that can be conducted as future work:

- The effect of using fuzzy classes, relations and rule definitions on the success of the semantic content extraction process can be evaluated.
- Temporal relations can be defined and extracted considering uncertainty.
- More tests on different domains can be done to evaluate the adequacy of VISCOM and the proposed semantic content extraction framework, ASCEF.

REFERENCES

[1] Apollo: an ontology editor tool. http://apollo.open.ac.uk/. [Online; accessed 05-June-2008].

[2] Chimaera: a software system for creating and maintaining distributed ontologies on the web. http://www.ksl.stanford.edu/software/chimaera/. [Online; accessed 05-June-2008].

[3] Daml: The darpa agent markup language. http://www.daml.org/. [Online; accessed 05-May-2008].

[4] Dolce : a descriptive ontology for linguistic and cognitive engineering. http://www.loa-cnr.it/DOLCE.html. [Online; accessed 05-June-2008].

[5] Gfo: General formal ontology. http://www.onto-med.de/en/theories/gfo/index.html. [Online; accessed 05-June-2008].

[6] Jena: a semantic web framework. http://www.hpl.hp.com/semweb/. [Online; accessed 15-April-2007].

[7] Kaon: The karlsruhe ontology and semantic web framework. http://kaon.semanticweb.org/. [Online; accessed 05-June-2008].

[8] Ontolingua: a distributed collaborative environment to browse, create, edit, modify, and use ontologies. http://www.ksl.stanford.edu/software/ontolingua/. [Online; accessed 05-June-2008].

[9] Opencyc: a general knowledge base reasoning engine. http://www.opencyc.org/. [Online; accessed 05-June-2008].

[10] Protégé ontology editor. http://protege.stanford.edu/. [Online; accessed 05-May-2006].

[11] Rdf: Resource description framework. http://www.w3.org/RDF/. [Online; accessed 03-July-2007].

[12] Rdfstore: a model/statement centric approach to create, manage and query rdf models. http://rdfstore.sourceforge.net/. [Online; accessed 05-June-2008].

[13] Sesame: an open source framework for storage, inferencing and querying of rdf data. http://www.openrdf.org/. [Online; accessed 05-June-2008].

[14] Sumo: Suggested upper merged ontology. http://www.ontologyportal.org/. [Online; accessed 05-June-2008].

[15] Webonto: a java applet to browse and edit knowledge models over the web. http://kmi.open.ac.uk/projects/webonto/. [Online; accessed 05-June-2008].

[16] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, 1983.

[17] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer (Cooperative Information Systems)*. The MIT Press, April 2004.

[18] Andrew D. Bagdanov, Marco Bertini, Alberto Del Bimbo, Carlo Torniai, and Giuseppe Serra. Semantic annotation and retrieval of video events using multimedia ontologies. In *Proc. of IEEE International Conference on Semantic Computing (ICSC)*, Irvine, California (USA), September 2007. IEEE Computer Society.

[19] Liang Bai, Song Yang Lao, Gareth Jones, and Alan F. Smeaton. Video semantic content analysis based on ontology. In *IMVIP 2007 - Proceedings of the 11th International Machine Vision and Image Processing Conference*, pages 117–124, 2007.

[20] Liang Bai, Songyang Lao, Weiming Zhang, Gareth J. F. Jones, and Alan F. Smeaton. A semantic event detection approach for soccer video based on perception concepts and finiste state machines. In *WIAMIS '07: Proceedings of the Eight International Workshop on Image Analysis for Multimedia Interactive Services*, page 30, Washington, DC, USA, 2007. IEEE Computer Society.

[21] Jie Bao, Yu Cao, Wallapak Tavanapong, and Vasant Honavar. Integration of domain-specific and domain-independent ontologies for colonoscopy video database annotation. In Hamid R. Arabnia, editor, *IKE*, pages 82–90. CSREA Press, 2004.

[22] Faisal I. Bashir and Ashfaq A. Khokhar. Video content modeling techniques: An overview. http://ece.ut.ac.ir/classpages/S84/TopicsinDatabase/Multimediaview-IEEEMM-v3.1.pdf, 2002. [Online; accessed 01-September-2006].

[23] Marco Bertini, Alberto Del Bimbo, and Carlo Torniai. Enhanced ontologies for video annotation and retrieval. In HongJiang Zhang, John Smith, and Qi Tian, editors, *Multimedia Information Retrieval*, pages 89–96. ACM, 2005.

[24] Marco Bertini, Alberto Del Bimbo, and Carlo Torniai. Automatic annotation and semantic retrieval of video sequences using multimedia ontologies. In Klara Nahrstedt, Matthew Turk, Yong Rui, Wolfgang Klas, and Ketan Mayer-Patel, editors, *ACM Multimedia*, pages 679–682. ACM, 2006.

[25] John Binder, Daphne Koller, Stuart J. Russell, and Keiji Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29(2-3):213–244, 1997.

[26] Stephan Bloehdorn, Kosmas Petridis, Nikos Simou, Vassilis Tzouvaras, Yannis Avrithis, Siegfried Handschuh, Yiannis Kompatsiaris, Steffen Staab, and Michael G. Strintzis. Knowledge representation for semantic multimedia content analysis and reasoning. In *Proceedings of the European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (EWIMT)*, 2004.

[27] Bob Bolles and Ram Nevatia. A hierarchical video event ontology in owl. Technical Report PNNL-14981, ARDA Challenge Workshop, 2004. https://rrc.mitre.org//nwrrcOWL-events-final-report.pdf [Online; accessed 01-April-2008].

[28] John S. Boreczky and Lawrence A. Rowe. Comparison of video shot boundary detection techniques. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 170–179, 1996.

[29] Francois Bremond, Nicolas Maillot, Monique Thonnat, and Van-Thinh Vu. Ontologies for video events. Technical report, Institut National de Recherche en Informatique et en Automatique, 2004. http://www−sop.inria.fr/orion/personnel/VanThinh.Vu/publi−cations/INRIA_Orion_RR_Ontology_2004.pdf [Online; accessed 01-March-2007].

[30] Dan Brickley and R.V. Guha. Rdf vocabulary description language 1.0: Rdf schema. Technical Report, W3C, 2004.

[31] L. Chaudron C. Castel and C. Tessier. What is going on? a high level interpretation of sequences of images. In *Workshop on Conceptual Descriptions from Images, European Conf. Computer Vision*, pages 13–27, 1996.

[32] Min Chen, Chengcui Zhang, and Shu-Ching Chen. Semantic event extraction using neural network ensembles. In *ICSC '07: Proceedings of the International Conference on Semantic Computing*, pages 575–580, Washington, DC, USA, 2007. IEEE Computer Society.

[33] W. Chen and D. S. Warren. C-logic of complex objects. In *PODS '89: Proceedings of the eighth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 369–378, New York, NY, USA, 1989.

[34] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. Daml+oil reference description. Technical Report, W3C, 2001. [Online; accessed 05-May-2006].

[35] S. Dasiopoulou, Papastathis V. K., V. Mezaris, Kompatsiaris I., and Strintzis M. G. An ontology framework for knowledge-assisted semantic video analysis and annotation. In *Proc. 4th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2004) at the 3rd International Semantic Web Conference (ISWC)*, 2004.

[36] Stamatia Dasiopoulou, Vasileios Mezaris, Ioannis Kompatsiaris, V.-K. Papastathis, and Michael G. Strintzis. Knowledge-assisted semantic video object detection. *IEEE Trans. Circuits Syst. Video Techn.*, 15(10):1210–1224, 2005.

[37] Larry S. Davis, Sandor Fejes, David Harwood, Yaser Yacoob, Ismail Haratoglu, and Michael J. Black. Visual surveillance of human activity. In *ACCV (2)*, pages 267–274, 1998.

[38] Nevenka Dimitrova, HongJiang Zhang, Behzad Shahraray, M. Ibrahim Sezan, Thomas S. Huang, and Avideh Zakhor. Applications of video-content analysis and retrieval. *IEEE MultiMedia*, 9(3):42–55, 2002.

[39] Dragan Djuric, Dragan Gasevic, and Vladan Devedzic. Ontology modeling and mda. *Journal of Object Technology*, 4(1):109–128, 2005.

[40] M. E. Donderler. *Data Modeling and Querying for Video Databases.* PhD thesis, Bilkent University, Turkey, 2002.

[41] Max J. Egenhofer and John R. Herring. A mathematical framework for the definition of topological relationships. In *Proceedings 4th. int. Symp on Spatial Data handling*, pages 803–813, 1990.

[42] Ahmet Ekin, A. Murat Tekalp, and Rajiv Mehrotra. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 12(7):796–807, 2003.

[43] J. Fan, W. Aref, A. Elmagarmid, M. Hacid, M. Marzouk, and X.Zhu. Multiview: Multilevel video content representation and retrieval. *Journal of Electronic Imaging*, 10(4):895–908, 2001.

[44] Jianping Fan, Ahmed K. Elmagarmid, Xingquan Zhu, Walid G. Aref, and Lide Wu. Classview: hierarchical video shot classification, indexing, and accessing. *IEEE Transactions on Multimedia*, 6(1):70–86, 2004.

[45] Myron Flickner, Harpreet S. Sawhney, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by image and video content: The qbic system. *IEEE Computer*, 28(9):23–32, 1995.

[46] Nagia Ghanem, Daniel DeMenthon, David Doermann, and Larry Davis. Representation and recognition of events in surveillance video using petri nets. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 7*, page 112, Washington, DC, USA, 2004.

[47] Hiranmay Ghosh, Santanu Chaudhury, Karthik Kashyap, and Brindaduti Maiti. Ontology specification and integration for multimedia applications. In *Ontologies in the Context of Information Systems*. Springer US, 2007.

[48] Christine Golbreich. Combining content-based retrieval and description logics reasoning. In *In Proceedings of the First International Workshop on Semantic Web Annotations for Multimedia (SWAMM'06)*, 2006.

[49] N. Guarino. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy.* IOS Press, Amsterdam, The Netherlands, The Netherlands, 1998.

[50] Nicola Guarino and P. Giaretta. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. J. I. Mars, editor, *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pages 25–32. IOS, 1995.

[51] Mohand-Said Hacid, Cyril Decleir, and Jacques Kouloumdjian. A database approach for modeling and querying video data. *IEEE Trans. Knowl. Data Eng.*, 12(5):729–750, 2000.

[52] Asaad Hakeem and Mubarak Shah. Multiple agent event detection and representation in videos. In Manuela M. Veloso and Subbarao Kambhampati, editors, *AAAI*, pages 89–94. AAAI Press / The MIT Press, 2005.

[53] Samira Hammiche, Salima Benbernou, Mohand-Said Hacid, and Athena Vakali. Semantic retrieval of multimedia data. In Shu-Ching Chen and Mei-Ling Shyu, editors, *MMDB*, pages 36–44. ACM, 2004.

[54] Gaurav Harit, Santanu Chaudhury, and Hiranmay Ghosh. Using multimedia ontology for generating conceptual annotations and hyperlinks in video collections. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 211–217, Washington, DC, USA, 2006. IEEE Computer Society.

[55] Alex Hauptmann. A video indexing ontology using fuzzy metadata. http://www.informedia.cs.cmu.edu/. [Online; accessed 16-May-2008].

[56] Jeff Heflin, James Hendler, Sean Luke, Carolyn Gasarch, Qin Zhendong, Lee Spector, and David Rager. Shoe-simple html ontology extensions. http://www.cs.umd.edu/projects/plus/SHOE/. [Online; accessed 05-May-2008].

[57] Somboon Hongeng, Ramakant Nevatia, and François Brémond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129–162, 2004.

[58] I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. van Harmelen, M. Klein, S. Staab, R. Studer, and E. Motta. The ontology inference layer oil. Technical Report. [Online; accessed 05-May-2006].

[59] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. Swrl: A semantic web rule language. Technical Report, W3C, 2004. [Online; accessed 12-March-2008].

[60] Xian-Sheng Hua, Lie Lu, and HongJiang Zhang. Automatic music video generation based on temporal pattern analysis. In Henning Schulzrinne, Nevenka Dimitrova, Angela Sasse, Sue B. Moon, and Rainer Lienhart, editors, *ACM Multimedia*, pages 472–475. ACM, 2004.

[61] Po-Whei Huang and Chu-Hui Lee. Image database design based on 9d-spa representation for spatial relations. *IEEE Trans. on Knowl. and Data Eng.*, 16(12):1486–1496, 2004.

[62] Chung Hee Hwang. Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. In Enrico Franconi and Michael Kifer, editors, *Proceedings of the 6th International Workshop on Knowledge Representation meets Databases (KRDB'99), Linköping, Sweden, July 29-30, 1999*, volume 21 of *CEUR Workshop Proceedings*, pages 14–20. CEUR-WS.org, 1999.

[63] I.Kompatsiaris, V.Mezaris, and M.G.Strintzis. Multimedia content indexing and retrieval using an object ontology. In Wiley G.Stamou, editor, *Multimedia Content and Semantic Web Methods, Standards and Tools*. Springer US, New York, NY, 2004.

[64] Stephen S. Intille and Aaron F. Bobick. Recognizing planned, multiperson action. *Computer Vision and Image Understanding: CVIU*, 81(3):414–445, 2001.

[65] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):852–872, 2000.

[66] Alejandro Jaimes, Belle L. Tseng, and John R. Smith. Modal keywords, ontologies, and reasoning for video understanding. In Erwin M. Bakker, Thomas S. Huang, Michael S. Lew, Nicu Sebe, and Xiang Sean Zhou, editors, *CIVR*, volume 2728 of *Lecture Notes in Computer Science*, pages 248–259. Springer, 2003.

[67] Michael Kifer and Georg Lausen. F-logic: a higher-order language for reasoning about objects, inheritance, and scheme. *SIGMOD Rec.*, 18(2):134–146, 1989.

[68] I. Kompatsiaris, Y. Avrithis, P. Hobson, and M.G. Strinzis. Integrating knowledge, semantics and content for user-centred intelligent media services: the acemedia project. Proc. of Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '04), Lisboa, Portugal, 2004.

[69] R Kowalski and M Sergot. A logic-based calculus of events. *New Gen. Comput.*, 4(1):67–95, 1986.

[70] Mesru Köprülü, Nihan Kesim Cicekli, and Adnan Yazici. Spatio-temporal querying in video databases. *Inf. Sci.*, 160(1-4):131–152, 2004.

[71] D. B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of ACM*, 38(11):33–38, 1995.

[72] Riccardo Leonardi and Pierangelo Migliorati. Semantic indexing of multimedia documents. *IEEE MultiMedia*, 9(2):44–51, 2002.

[73] C.Y. Lin, B.L. Tseng, and J.R. Smith. Ibm mpeg-7 annotation tool version 1.5.1, 2003. http://www.alphaworks.ibm.com/tech/videoannex [Online; accessed 15-April-2007].

[74] Chien Yong Low, Qi Tian, and Hongjiang Zhang. An automatic news video parsing, indexing and browsing system. In *MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia*, pages 425–426, New York, NY, USA, 1996. ACM.

[75] M. Maier. A logic for objects. In *inProc. Workshop on Foundations of Deductive Databases and Logic Programming,Washington, DC*, pages 6–26, 1986.

[76] Oge Marques and Borko Furht. Muse: A content-based image search and retrieval system using relevance feedback. *Multimedia Tools Appl.*, 17(1):21–50, 2002.

[77] José Martínez. Mpeg-7 overview (version 10). Requirements ISO/IEC JTC1 /SC29 /WG11 N6828, International Organisation For Standardisation, Oct 2003. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm/ [Online; accessed 04-April-2008].

[78] Gérard G. Medioni, Isaac Cohen, François Brémond, Somboon Hongeng, and Ramakant Nevatia. Event detection and analysis from video streams. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(8):873–889, 2001.

[79] Vasileios Mezaris, Ioannis Kompatsiaris, Nikolaos V. Boulgouris, and Michael G. Strintzis. Real-time compressed-domain spatiotemporal segmentation and ontologies for video indexing and retrieval. *IEEE Trans. Circuits Syst. Video Techn.*, 14(5):606–621, 2004.

[80] Vasileios Mezaris, Ioannis Kompatsiaris, and Michael G. Strintzis. Region-based image retrieval using an object ontology and relevance feedback. *EURASIP J. Appl. Signal Process.*, 2004(1):886–901, 2004.

[81] George A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[82] M. Missikoff and F. Taglino. Symontox: A web-ontology tool for ebusiness domains. In *WISE '03: Proceedings of the Fourth International Conference on Web Information Systems Engineering*, page 343, Washington, DC, USA, 2003. IEEE Computer Society.

[83] Nicolas Moënne-Loccoz, François Brémond, and Monique Thonnat. Recurrent bayesian network for the recognition of human behaviors from video. In James L. Crowley, Justus H. Piater, Markus Vincze, and Lucas Paletta, editors, *ICVS*, volume 2626 of *Lecture Notes in Computer Science*, pages 68–77. Springer, 2003.

[84] R. J. Morris and D. C. Hogg. Statistical models of object interaction. *International Journal of Computer Vision*, 37(2):209–215, 2000.

[85] Moving Picture Experts Group (MPEG). Mpeg-7 reference software experimentation model, 2003. http://standards.iso.org/ittf/PubliclyAvailableStandards/c035364_ISO_IEC_15938-6(E)_Reference_Software.zip [Online; accessed 01-April-2007].

[86] S. Muller-Schneiders, T. Jager, H. S. Loos, and W. Niem. Performance evaluation of a real time video surveillance system. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 137–143, 2005.

[87] Milind R. Naphade and Thomas S. Huang. Detecting semantic concepts using context and audiovisual features. In *IEEE Workshop on Detection and Recognition of Events in Video*, pages 92–98, 2001.

[88] R. Nevatia, J. Hobbs, and B. Bolles. An ontology for video event representation. In *Computer Vision and Pattern Recognition Workshop, 2004 Conference on*, page 119, 2004.

[89] Ram Nevatia, Tao Zhao, and Somboon Hongeng. Hierarchical language-based representation of events in video streams. In *Proc. Workshop Event Mining (in conjunction with IEEE Int'l Conf. Computer Vision and Pattern Recognition [CVPR])*, page 39. IEEE CS Press, 2003.

[90] María Auxilio Medina Nieto. Boemie: An overview of ontologies. Technical report, Universidad Las Americas Puebla, 2003. https://starlab.vub.ac.be/teaching/ontologies_overview.pdf [Online; accessed 05-May-2006].

[91] Nuria Oliver, Ashutosh Garg, and Eric Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *Comput. Vis. Image Underst.*, 96(2):163–180, 2004.

[92] Nuria M. Oliver, Barbara Rosario, and Alex Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[93] Vinay Chaudhri Jerome Thomere SRI International Peter Karp, Pangea Systems. Xol ontology exchange language. http://www.xml.com/pub/r/888, 2000. [Online; accessed 05-May-2008].

[94] M. Petkovic and W. Jonker. An overview of data models and query languages for content-based video retrieval. In *International Conference on Advances in Infrastructure for E-Business, Science, and Education on the Internet, l'Aquila, Italy*. Scuola Superiore G. Reiss Romoli, August 2000.

[95] M. Petkovic and W. Jonker. Content-based retrieval of spatio-temporal video events. In *Managing Information Technology in a Global Economy: Proceedings of the 12th Information Resources Management Association International Conference (IRMA 2001), Toronto, Canada*, Hershey, 2001. Idea Group Publishing.

[96] Milan Petkovic and Willem Jonker. Content-based video retrieval by integrating spatio-temporal and stochastic recognition of events. In *IEEE Workshop on Detection and Recognition of Events in Video*, pages 75–82, 2001.

[97] S. Petridis and N. Tsapatsoulis. Boemie: D2.1 methodology for semantics extraction frommultimedia content. Technical Report FP6-027538 D2.1, National Centre for Scientific Research (NCSR), 2006. http://www.boemie.org/files/BOEMIE-d2_1-v2.pdf [Online; accessed 05-June-2008].

[98] Claudio S. Pinhanez and Aaron F. Bobick. Human action detection using pnf propagation of temporal constraints. In *CVPR*, pages 898–904. IEEE Computer Society, 1998.

[99] R. Polana and R.C. Nelson. Detecting activities. In *DARPA93*, pages 569–574, 1993.

[100] Ramakant Nevatia Pradeep Natarajan. Edf: A framework for semantic annotation of video. In *Tenth IEEE International Conference on Computer Vision Workshops (ICCVW'05)*, page 1876, 2005.

[101] D. Reidsma, J. Kuper, T. Declerck, H. Saggion, and H. Cunningham. Cross-Document Ontology-Based Information Extraction for Multimedia Retrieval. In *Proceedings of ICCS'03*, pages 41–48, Desden, 2003.

[102] P. Remagnino, J. Orwell, and G.A. Jones. Visual interpretation of people and vehicle behaviours using a society of agents. In *in: Italian Association for Artificial Intelligence*, pages 333–342, 1999.

[103] P. Remagnino, T. Tan, and K. Baker. Agent orientated annotation in model based visual surveillance. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, pages 857–862, Washington, DC, USA, 1998. IEEE Computer Society.

[104] Nathanaël Rota and Monique Thonnat. Activity recognition from video sequences using declarative models. In Werner Horn, editor, *ECAI*, pages 673–680. IOS Press, 2000.

[105] David A. Sadlier and Noel E. O'Connor. Event detection in field sports video using audio-visual features and a support vector machine. *IEEE Trans. Circuits Syst. Video Techn.*, 15(10):1225–1233, 2005.

[106] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[107] Michael K. Smith, Chris Welty, and Deborah L. McGuinness. Owl web ontology language guide. Technical Report, W3C, 2004. [Online; accessed 15-April-2008].

[108] Cees Snoek and Marcel Worring. Multimedia event-based video indexing using time intervals. *IEEE Transactions on Multimedia*, 7(4):638–647, 2005.

[109] Dan Song, Hai Tao Liu, Miyoung Cho, Hanil Kim, and PanKoo Kim. Domain knowledge ontology building for semantic video event description. In *CIVR*, pages 267–275, 2005.

[110] York Sure, Michael Erdmann, Jürgen Angele, Steffen Staab, Rudi Studer, and Dirk Wenke. Ontoedit: Collaborative ontology engineering for the semantic web. In I. Horrocks and J. Hendler, editors, *Proceedings of the First International Semantic Web Conference 2002 (ISWC 2002), June 9-12 2002, Sardinia, Italia*, volume 2342 of *LNCS*, pages 221–235. Springer, 2002.

[111] B. Swartout, P. Ramesh, K. Knight, and T. Russ. Toward distributed use of large-scale ontologies. *AAAI Symposium on Ontological Engineering*, 1997.

[112] Tanveer Fathima Syeda-Mahmood and Savitha Srinivasan. Detecting topical events in digital video. In *ACM Multimedia*, pages 85–94, 2000.

[113] Karthik Thatipamula, Santanu Chaudhury, and Hiranmay Ghosh. Specifying spatio temporal relations for multimedia ontologies. In Sankar K. Pal, Sanghamitra Bandyopadhyay, and Sambhunath Biswas, editors, *PReMI*, volume 3776 of *Lecture Notes in Computer Science*, pages 527–532. Springer, 2005.

[114] David Thirde, Mark Borg, James Ferryman, Florent Fusier, Valery Valentin, Francois Bremond, and Monique Thonnat. Video event recognition for aircraft activity monitoring. In *Proceedings of 8th IEEE International Conference on Intelligent Transportation Systems*, Vienna, Austria, 13-16 September 2005. IEEE Computer Society.

[115] Chrisa Tsinaraki, Panagiotis Polydoros, Fotis Kazasis, and Stavros Christodoulakis. Ontology-based semantic indexing for mpeg-7 and tv-anytime audiovisual content. *Multimedia Tools Appl.*, 26(3):299–325, 2005.

[116] M. Vazirgiannis. Uncertainty handling in spatial relationships. In *SAC '00: Proceedings of the 2000 ACM symposium on Applied computing*, pages 494–500, New York, NY, USA, 2000. ACM.

[117] Julio César Arpírez Vega, Óscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Webode in a nutshell. *AI Magazine*, 24(3):37–48, 2003.

[118] Shankar Vembu, Malte Kiesel, Michael Sintek, and Stephan Baumann. Towards bridging the semantic gap in multimedia annotation and retrieval. In *Proceedings of the First International Workshop on Semantic Web Annotations for Multimedia*, 2006.

[119] Van-Thinh Vu, François Brémond, and Monique Thonnat. Automatic video interpretation: A novel algorithm for temporal scenario recognition. In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 1295–1302. Morgan Kaufmann, 2003.

[120] Van-Thinh Vu, François Brémond, and Monique Thonnat. Automatic video interpretation: A recognition algorithm for temporal scenarios based on pre-compiled scenario models. In James L. Crowley, Justus H. Piater, Markus Vincze, and Lucas Paletta, editors, *ICVS*, volume 2626 of *Lecture Notes in Computer Science*, pages 523–533. Springer, 2003.

[121] Howard Wactlar. Auto-summarization and visualization over multiple video documents and libraries. Technical Report NSF Cooperative Agreement No. IIS-9817496, Carnegie Mellon University School of Computer Science, 2001.

[122] Kasun Wickramaratna, Min Chen, Shu-Ching Chen, and Mei-Ling Shyu. Neural network based framework for goal event detection in soccer videos. In *ISM*, pages 21–28. IEEE Computer Society, 2005.

[123] Wikipedia. Ontology (information science). http://en.wikipedia.org [Online; accessed 05-June-2008].

[124] Andrew D. Wilson and Aaron F. Bobick. Recognition and interpretation of parametric gesture. In *ICCV*, pages 329–336, 1998.

[125] Changsheng Xu, Jinjun Wang, Kongwah Wan, Yiqun Li, and Lingyu Duan. Live sports event detection based on broadcast video and web-casting text. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 221–230, New York, NY, USA, 2006. ACM.

[126] Yakup Yildirim and Adnan Yazici. Ontology-supported video modeling and retrieval. In *Adaptive Multimedia Retrieval*, pages 28–41, 2006.

[127] Yakup Yildirim, Turgay Yilmaz, and Adnan Yazici. Ontology-supported object and event extraction with a genetic algorithms approach for object classification. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 202–209, New York, NY, USA, 2007. ACM.

[128] Turgay Yilmaz. Object Extraction from Images/Videos Using a Genetic Algorithm Based Approach. Master's thesis, Computer Engineering Department, METU, Ankara, Turkey, January 2008.

[129] Xinguo Yu, Changsheng Xu, Hon Wai Leong, Qi Tian, Qing Tang, and Kongwah Wan. Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In Lawrence A. Rowe, Harrick M. Vin, Thomas Plagemann, Prashant J. Shenoy, and John R. Smith, editors, *ACM Multimedia*, pages 11–20. ACM, 2003.

[130] R. Zabih, K. Mai, and J. Miller. A robust method for detecting cuts and dissolves in video sequences. In *Proc. 3rd Int'l Conf. Multimedia (ACM Multimedia 95)*,, New York, NY, USA, 1995. ACM Press.

[131] Lotfali Asker Zadeh. Fuzzy sets. In *Information and Control*, pages 338–353, 1965.

[132] Lihi Zelnik-Manor and Michal Irani. Event-based analysis of video. In *CVPR (2)*, pages 123–130. IEEE Computer Society, 2001.

[133] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu. Video parsing, retrieval and browsing: an integrated and content-based solution. In *MULTIMEDIA '95: Proceedings of the third ACM international conference on Multimedia*, pages 15–24, New York, NY, USA, 1995. ACM.

[134] Yifan Zhang, Changsheng Xu, YongRui, Jinqiao Wang, and Hanqing Lu. Semantic event extraction from basketball games using multi-modal analysis. In *Proc. of IEEE ICME 2007*, pages 2190–2193, Beijing, China, 2007.

[135] Hakan Öztarak. Structural and event based multimodal video data modeling. Master's thesis, Computer Engineering Department, METU, Ankara, Turkey, January 2006.

APPENDIX A

VISCOM OWL CODE

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns="file:///G:/TEZ/Tez/Model/VideoMetaModel.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="file:///G:/TEZ/Tez/Model/VideoMetaModel.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Role"/>
  <owl:Class rdf:ID="SpatialMovement"/>
  <owl:Class rdf:ID="ObjectComposedOfType"/>
  <owl:Class rdf:ID="PositionalSpatialRelation">
    <owl:disjointWith>
      <owl:Class rdf:ID="TopologicalSpatialRelation"/>
    </owl:disjointWith>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="SpatialRelation"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ObjectRole">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
```

```
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasRoledObject"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasRole"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Event">
  <owl:disjointWith>
    <owl:Class rdf:ID="Concept"/>
  </owl:disjointWith>
  <owl:disjointWith>
    <owl:Class rdf:ID="Object"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Component"/>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasEventDefinition"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
```

```
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasTemporalEventComponent"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasEventObjectRole"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="TemporalRelation"/>
<owl:Class rdf:ID="ConceptComponent">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasComponent"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
```

```
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasConceptObjectRole"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="hasRelevance"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="TemporalSpatialChangeComponent">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasSecondSpatialChange"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      <owl:onProperty>
```

```
        <owl:ObjectProperty rdf:ID="hasFirstSpatialChange"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasTemporalSubEventRelation"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="LowLevelFeature">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="hasLowLevelFeatureName"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="hasLowLevelFeatureValue"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
```

```
</owl:Class>
<owl:Class rdf:ID="SpatialChangePeriod"/>
<owl:Class rdf:ID="ObjectComposedOfGroup">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasParentObject"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasComposedOfType"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:ID="TemporalEventComponent">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasFirstEvent"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
```

```
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasSecondEvent"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasTemporalEventRelation"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="DistanceSpatialRelation">
  <rdfs:subClassOf rdf:resource="#SpatialRelation"/>
</owl:Class>
<owl:Class rdf:ID="Similarity">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="hasSimilarityRelevance"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
```

```
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasSimilarityWith"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Component">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="hasSynonymName"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasSimilarContext"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#TopologicalSpatialRelation">
  <rdfs:subClassOf rdf:resource="#SpatialRelation"/>
  <owl:disjointWith rdf:resource="#PositionalSpatialRelation"/>
</owl:Class>
```

```
<owl:Class rdf:ID="EventDefinition">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="hasEventRelevance"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasEventDefinitionObjectRole"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:ID="hasUniqueSpatialChange"/>
      </owl:onProperty>
      <owl:maxCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
      <owl:onProperty>
```

```
        <owl:ObjectProperty rdf:ID="hasTemporalSpatialChangeComponent"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasEventSpatialRelationComponent"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Concept">
  <owl:disjointWith rdf:resource="#Event"/>
  <owl:disjointWith>
    <owl:Class rdf:about="#Object"/>
  </owl:disjointWith>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasConceptComponent"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Component"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
      <owl:onProperty>
```

```
        <owl:ObjectProperty rdf:ID="hasConceptOccurence"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  ></rdfs:comment>
</owl:Class>
<owl:Class rdf:ID="SpatialChange">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasSpatialChangePeriod"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:maxCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasInitialSpatialRelationComponent"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:maxCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:maxCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasFinalSpatialRelationComponent"/>
      </owl:onProperty>
    </owl:Restriction>
```

```
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:minCardinality rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="hasSpatialChangeObject"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="hasSpatialChangeOccurence"/>
    </owl:onProperty>
    <owl:minCardinality rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:minCardinality rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="hasSpatialChangeObjectRole"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:ObjectProperty rdf:ID="hasSpatialMovementComponent"/>
    </owl:onProperty>
```

```
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="ObjectComposedOfRelation">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasObjectComposedOfGroup"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:ID="hasObjectToParentRelevance"/>
      </owl:onProperty>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="SpatialMovementComponent">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasSpatialMovement"/>
```

```
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasMovingObject"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="SpatialRelationComponent">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      <owl:onProperty>
        <owl:DatatypeProperty rdf:ID=
        "hasSpatialRelationMembershipValue"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasSpatialRelation"/>
      </owl:onProperty>
      <owl:maxCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">3</owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
```

```
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasObject"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasSubject"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Object">
  <rdfs:subClassOf rdf:resource="#Component"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasObjectLowLevelFeature"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype=
      "http://www.w3.org/2001/XMLSchema#int">0</owl:minCardinality>
```

```
      <owl:onProperty>
        <owl:FunctionalProperty rdf:ID="hasObjectComposedOfRelation"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <owl:disjointWith rdf:resource="#Concept"/>
  <owl:disjointWith rdf:resource="#Event"/>
</owl:Class>
<owl:ObjectProperty rdf:about="#hasSpatialRelation">
  <rdfs:domain rdf:resource="#SpatialRelationComponent"/>
  <rdfs:range rdf:resource="#SpatialRelation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasObject">
  <rdfs:domain rdf:resource="#SpatialRelationComponent"/>
  <rdfs:range rdf:resource="#Object"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasEventSpatialRelationComponent">
  <rdfs:range rdf:resource="#SpatialRelationComponent"/>
  <rdfs:domain rdf:resource="#EventDefinition"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSpatialMovement">
  <rdfs:domain rdf:resource="#SpatialMovementComponent"/>
  <rdfs:range rdf:resource="#SpatialMovement"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasParentObject">
  <rdfs:range rdf:resource="#Object"/>
  <rdfs:domain rdf:resource="#ObjectComposedOfGroup"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasInitialSpatialRelationComponent">
  <rdfs:domain rdf:resource="#SpatialChange"/>
  <rdfs:range rdf:resource="#SpatialRelationComponent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasTemporalEventComponent">
  <rdfs:domain rdf:resource="#Event"/>
```

```
  <rdfs:range rdf:resource="#TemporalEventComponent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasConceptOccurence">
  <rdfs:domain rdf:resource="#Concept"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasRoledObject">
  <rdfs:domain rdf:resource="#ObjectRole"/>
  <rdfs:range rdf:resource="#Object"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasMovingObject">
  <rdfs:domain rdf:resource="#SpatialMovementComponent"/>
  <rdfs:range rdf:resource="#Object"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasFirstEvent">
  <rdfs:domain rdf:resource="#TemporalEventComponent"/>
  <rdfs:range rdf:resource="#Event"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSubject">
  <rdfs:range rdf:resource="#Object"/>
  <rdfs:domain rdf:resource="#SpatialRelationComponent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSpatialChangePeriod">
  <rdfs:range rdf:resource="#SpatialChangePeriod"/>
  <rdfs:domain rdf:resource="#SpatialChange"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasObjectLowLevelFeature">
  <rdfs:range rdf:resource="#LowLevelFeature"/>
  <rdfs:domain rdf:resource="#Object"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasEventDefinitionObjectRole">
  <rdfs:domain rdf:resource="#EventDefinition"/>
  <rdfs:range rdf:resource="#ObjectRole"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSpatialChangeObjectRole">
```

```
  <rdfs:range rdf:resource="#ObjectRole"/>
  <rdfs:domain rdf:resource="#SpatialChange"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasFinalSpatialRelationComponent">
  <rdfs:range rdf:resource="#SpatialRelationComponent"/>
  <rdfs:domain rdf:resource="#SpatialChange"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasEventDefinition">
  <rdfs:range rdf:resource="#EventDefinition"/>
  <rdfs:domain rdf:resource="#Event"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasObjectComposedOfGroup">
  <rdfs:range rdf:resource="#ObjectComposedOfGroup"/>
  <rdfs:domain rdf:resource="#ObjectComposedOfRelation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSecondSpatialChange">
  <rdfs:range rdf:resource="#SpatialChange"/>
  <rdfs:domain rdf:resource="#TemporalSpatialChangeComponent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasConceptObjectRole">
  <rdfs:range rdf:resource="#ObjectRole"/>
  <rdfs:domain rdf:resource="#Concept"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSimilarityWith">
  <rdfs:range rdf:resource="#Component"/>
  <rdfs:domain rdf:resource="#Similarity"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasTemporalSpatialChangeComponent">
  <rdfs:domain rdf:resource="#EventDefinition"/>
  <rdfs:range rdf:resource="#TemporalSpatialChangeComponent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSpatialChangeObject">
  <rdfs:range rdf:resource="#Object"/>
  <rdfs:domain rdf:resource="#SpatialChange"/>
```

```
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasComposedOfType">
  <rdfs:domain rdf:resource="#ObjectComposedOfGroup"/>
  <rdfs:range rdf:resource="#ObjectComposedOfType"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasFirstSpatialChange">
  <rdfs:domain rdf:resource="#TemporalSpatialChangeComponent"/>
  <rdfs:range rdf:resource="#SpatialChange"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSpatialChangeOccurence">
  <rdfs:domain rdf:resource="#SpatialChange"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSecondEvent">
  <rdfs:range rdf:resource="#Event"/>
  <rdfs:domain rdf:resource="#TemporalEventComponent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasTemporalEventRelation">
  <rdfs:range rdf:resource="#TemporalRelation"/>
  <rdfs:domain rdf:resource="#TemporalEventComponent"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasRole">
  <rdfs:range rdf:resource="#Role"/>
  <rdfs:domain rdf:resource="#ObjectRole"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSimilarContext">
  <rdfs:domain rdf:resource="#Component"/>
  <rdfs:range rdf:resource="#Similarity"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasComponent">
  <rdfs:domain rdf:resource="#ConceptComponent"/>
  <rdfs:range rdf:resource="#Component"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasEventObjectRole">
  <rdfs:domain rdf:resource="#Event"/>
```

```
  <rdfs:range rdf:resource="#ObjectRole"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasSpatialMovementComponent">
  <rdfs:range rdf:resource="#SpatialMovementComponent"/>
  <rdfs:domain rdf:resource="#SpatialChange"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasConceptComponent">
  <rdfs:range rdf:resource="#ConceptComponent"/>
  <rdfs:domain rdf:resource="#Concept"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#hasTemporalSubEventRelation">
  <rdfs:domain rdf:resource="#TemporalSpatialChangeComponent"/>
  <rdfs:range rdf:resource="#TemporalRelation"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="#hasRelevance">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#ConceptComponent"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasSimilarityRelevance">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#Similarity"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasEventRelevance">
  <rdfs:domain rdf:resource="#EventDefinition"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasLowLevelFeatureValue">
  <rdfs:domain rdf:resource="#LowLevelFeature"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasSpatialRelationMembershipValue">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="#SpatialRelationComponent"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasSynonymName">
  <rdfs:domain rdf:resource="#Component"/>
```

```
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasLowLevelFeatureName">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#LowLevelFeature"/>
</owl:DatatypeProperty>
<owl:FunctionalProperty rdf:about="#hasObjectToParentRelevance">
  <rdf:type rdf:resource=
  "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
  <rdfs:domain rdf:resource="#ObjectComposedOfRelation"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#hasUniqueSpatialChange">
  <rdfs:range rdf:resource="#SpatialChange"/>
  <rdfs:domain rdf:resource="#EventDefinition"/>
  <rdf:type rdf:resource=
  "http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:about="#hasObjectComposedOfRelation">
  <rdf:type rdf:resource=
  "http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:range rdf:resource="#ObjectComposedOfRelation"/>
  <rdfs:domain rdf:resource="#Object"/>
</owl:FunctionalProperty>
<SpatialMovement rdf:ID="stationary"/>
<SpatialMovement rdf:ID="movementRight"/>
<DistanceSpatialRelation rdf:ID="distanceFar"/>
<TemporalRelation rdf:ID="temporalMeets"/>
<ObjectComposedOfType rdf:ID="isA"/>
<PositionalSpatialRelation rdf:ID="positionalRightSide"/>
<TemporalRelation rdf:ID="temporalStarts"/>
<SpatialChangePeriod rdf:ID="endToStart"/>
<TemporalRelation rdf:ID="temporalOverlap"/>
<PositionalSpatialRelation rdf:ID="positionalBelow"/>
<ObjectComposedOfType rdf:ID="memberOf"/>
```

```
  <SpatialMovement rdf:ID="movementDown"/>
  <TemporalRelation rdf:ID="temporalDuring"/>
  <TopologicalSpatialRelation rdf:ID="topologicalDisjoint"/>
  <TemporalRelation rdf:ID="temporalFinishes"/>
  <TemporalRelation rdf:ID="temporalEqual"/>
  <ObjectComposedOfType rdf:ID="composedOf"/>
  <PositionalSpatialRelation rdf:ID="positionalLeftSide"/>
  <TopologicalSpatialRelation rdf:ID="topologicalInside"/>
  <PositionalSpatialRelation rdf:ID="positionalAbove"/>
  <SpatialChangePeriod rdf:ID="startToEnd"/>
  <TopologicalSpatialRelation rdf:ID="topologicalPartlyInside"/>
  <SpatialMovement rdf:ID="movementLeft"/>
  <ObjectComposedOfType rdf:ID="substanceOf"/>
  <ObjectComposedOfType rdf:ID="partOf"/>
  <SpatialChangePeriod rdf:ID="startToStart"/>
  <TemporalRelation rdf:ID="temporalBefore"/>
  <SpatialMovement rdf:ID="movementUp"/>
  <TopologicalSpatialRelation rdf:ID="topologicalTouch"/>
  <SpatialChangePeriod rdf:ID="endToEnd"/>
  <DistanceSpatialRelation rdf:ID="distanceNear"/>
</rdf:RDF>
```

VITA

PERSONAL INFORMATION

Surname, Name: Yıldırım, Yakup
Nationality: Turkish (TC)
Date and Place of Birth: 2 August 1975 , Balıkesir
Marital Status: Married
Phone: +90 312 473 39 85
email: yy@alumni.bilkent.edu.tr

EDUCATION

Degree	Institution	Year of Graduation
MS	METU Computer Engineering	2000
BS	Bilkent University Computer Engineering	1997
High School	Kayseri Science High School	1993

WORK EXPERIENCE

Year	Place	Enrollment
2008-Present	NC3A	Senior Scientist
2003-2008	Havelsan A.Ş.	Software Manager
2002-2003	ISF Yazılım	Software Engineer
1999-2002	Cybersoft A.Ş.	Software Engineer
1997-1999	Dept. of Computer Engineering, METU	Teaching Assistant

FOREIGN LANGUAGES

Advanced English, Intermediate Dutch

PUBLICATIONS

1. N. K. Cicekli and Y. Yildirim. Formalizing Workflows using the Event Calculus. *Proceedings of the 11th Intl.Conf., DEXA 2000*, London, UK, pp. 222-231, Sept. 2000.

2. Y.Yildirim and A.Yazici. Ontology-Supported Video Modeling and Retrieval. *AMR 2006*, S. Marchand-Maillet et al. (Eds.): Lecture Notes for Computer Science (LNCS) 4398, (Springer-Verlag) pp. 28-41, 2007.

3. Y.Yildirim, T. Yilmaz and A. Yazici. Ontology-supported Object and Event Extraction with a Genetic Algorithms Approach for Object Classification. *CIVR '07: Proceedings of the 6th ACM Int. Conference on Image and Video Retrieval*, pp. 202-209, 2007.

4. Y.Yildirim, A. Yazici, T. Yilmaz. Automatic Semantic Content Extraction in Videos using a Spatio-Temporal Ontology Model. submitted to *Multimedia Tools and Applications* journal.

RESEARCH INTERESTS

Ontology-based modeling, video content extraction, multimedia applications.