INTEGRATION
OF
FUZZY OBJECT-ORIENTED
MULTIMEDIA DATABASE COMPONENTS

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

BY

UTKU DEMİR

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

NOVEMBER 2010

Approval of the thesis:

INTEGRATION
OF
FUZZY OBJECT-ORIENTED
MULTIMEDIA DATABASE COMPONENTS

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

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural And Applied Sciences** _____________________

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** _____________________

Prof. Dr. Adnan Yazıcı
Supervisor, **Computer Engineering, METU** _____________________

Examining Committee Members:

Assoc. Prof. Dr. Ahmet Coşar
Computer Engineering, METU _____________________

Prof. Dr. Adnan Yazıcı
Computer Engineering, METU _____________________

Asst. Prof. Dr. Murat Koyuncu
Information Systems Engineering, Atılım University _____________________

Asst. Prof. Dr. Pınar Şenkul
Computer Engineering, METU _____________________

M.S.Turgay Yılmaz
HAVELSAN _____________________

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 **:** Utku DEMİR

Signature **:**

ABSTRACT

INTEGRATION
OF
FUZZY OBJECT-ORIENTED
MULTIMEDIA DATABASE COMPONENTS

Demir, Utku

Ms., Department of Computer Engineering

Supervisor: Prof. Dr. Adnan YAZICI

November 2010, 119 pages

Improvements in technology have increased the amount of human interactive systems that support visual and audial operations. Besides many others, especially the recent entertainment industry has been built on the digital world and processing large collections of multimedia materials. Having huge amount multimedia data revealed the need for efficient and effective ways of modeling, storing, addressing, and retrieving such huge data, mostly, the semantic contents in it. Although there are some database management systems that support multimedia objects by some add-ons or extensions, they are far away from satisfying user requests, which are usually based on similarity and semantic contents.

In this study, the requirements of multimedia databases are analyzed. To satisfy such requirements, a database architecture which is specialized for multimedia objects is constructed and a conceptual data model, handling semantic multimedia contents is implemented. In the architecture, a semantic information extractor, a high-dimensional index structure and an intelligent fuzzy object-oriented database component are integrated through a coordinator structure. The proposed architecture is realized and a prototype system is implemented by tightly coupling several multimedia modules.

Key words: Fuzzy, Multimedia, Database

ÖZ

BULANIK, NESNEYE DAYALI ÇOKLU ORTAM VERİTABANI BİLEŞENLERİNİN ENTEGRASYONU

Demir, Utku
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Adnan Yazıcı
Kasım 2010, 119 sayfa

Teknolojideki gelişmeler, görsel ve işitsel öğeleri destekleyen etkileşimli systemlerin miktarını arttırmıştır. Diğer birçoklarıyla beraber özellikle eğlence sektörü, dijital bir dünya üzerine kurulmuş olup, büyük miktarlarda çoklu ortam verisi işlemektedir. Büyük miktarda çoklu ortam verisine sahip olmak, bu devasa verinin, çoğunlukla da içindeki mantıksal içeriğin etkin ve işe yarar bir şekilde modellenmesi, saklanması, adreslenmesi ve erişilmesi ihtiyacını doğurmuştur. Bir takım yama ve eklentilerle çoklu ortam verilerini destekleyen veritabanları olsa da, bunlar, genellikle benzerlik ve mantıksal içeriğe dayanan kullanıcı isteklerini karşılamaktan oldukça uzaktır.

Bu çalışmada, çoklu ortam veritabanlarının gereksinimleri analiz edilmiştir. Bu gereksinimleri karşılamak için, çoklu ortam nesnelerine özel bir veri tabanı mimarisi kurulmuş ve çoklu ortam verilerinin mantıksal içeriğini ele alan bir kavramsal veri modeli gerçekleştirilmiştir. Bu yapıda, bir mantıksal bilgi çıkarıcı, bir çok boyutlu index yapısı ve akıllı, bulanık, nesneye dayalı bir veritabanı, bir koordinatör bileşen aracılığıyla bütünleşmiştir. Ayrıca önerilen yapı, birkaç çoklu ortam modülünün uygun bir şekilde bir araya getirilmesiyle somutlaştırılmış ve prototip bir sistem geliştirilmiştir.

Anahtar Kelimeler: Bulanık, Çoklu Ortam, Veritabanı

To My Family …

ACKNOWLEDGMENTS

I would like to present my deepest appreciation to my supervisor, Prof. Dr. Adnan Yazıcı for his valuable guidance, advice and insight during this study. It is an honor for me to share his knowledge and wisdom.

I would like to thank all members of Ceng Multimedia Research Group for their technical guidance and support.

I am very grateful to my commanders Sait Cengiz Özsoy, Kadir Murat Biçer and Bora Araz, and all my colleagues, for their patience and insight throughout this thesis.

Finally, I am very thankful to my beloved family, my dear wife Müge and our sweet daughter Ece, 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 this work. I am also grateful to my parents who always encouraged me to finish the thesis.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DDL	: Description Definition Language
JavaWS	: Java Web Start
RTSP	: Real Time Stream Protocol
RTP	: Real Time Protocol
AIFF	: Audio Interchange File Format
AVI	: Audio Video Interleave
MIDI	: Musical Instrument Digital Interface
MPEG	: Moving Picture Experts Groups
XML	: Extensible Markup Language
HTML	: Hyper Text Markup Language
OO	: Object-Oriented
DBMS	: Database Management System
RDBMS	: Relational Database Management System
OODBMS	: Object-Oriented Database Management System
BLOB	: Binary Large Objects
BRDF	: Best Representative and Discriminative Feature
GA	: Genetic Algorithm
XM	: Experimental Model
SEI	: Service Endpoint Interface
WSDL	: Web Service Description Language
SOAP	: Simple Object Access Protocol
UDDI	: Universal Description, Discovery and Integration
CLIPS	: C Language Integrated Production System
AV	: Audiovisual
JRE	: Java Runtime Environment

CHAPTER 1

INTRODUCTION

With the improvements in technology, large capacity and fast multimedia devices became cheaper and people, obtaining those hardware, have began to store and process large volumes of multimedia data excessively. Due to the natural properties of multimedia objects like having k-dimensional data, complicated non-standard uncertain objects, temporal and spatial aspects of these objects, etc. ; conventional database approaches are not adequate for handling multimedia content in most cases. Therefore, in order to access and retrieve desired portion of multimedia content easily, the need for multimedia databases has been raised. These multimedia frameworks should be able to:

- extract concepts, objects, events and relations between them,
- efficiently store and index semantic contents,
- have some specialized extended capabilities that provide high level abstractions for flexible queries, such as content-based and query-by-example type searches.

When multimedia data is in discussion, three different types of information gain importance:

- metadata or cataloging information as *textual content*, such as date, title and author,
- the knowledge you are exposed or inferred as high level *semantic content*,
- the low level features such as fundamental frequency, harmonicity, dominant color, region shape and edge histogram as *audiovisual content* [1].

Although managing and retrieving textual and audiovisual content is easier and straightforward, users are usually interested in the meaning of the content, in other words, the high level semantic information in multimedia data.

Concepts, objects, spatial/temporal relations and interactions between these entities, namely events, are the basic components of semantic content [2]. The process of extracting these entities, in other words annotating semantic information, is a challenging research area on multimedia domain. Using manual annotation techniques for extracting information from multimedia data is mostly boring, exhausting, slow and resource consuming, therefore, limited, inefficient and subjective to user. Also, most of the time, retrieving previously annotated data is not satisfying, since it is restricted to pre-extracted information. Therefore, an automatic, preferably real-time annotation system is required. Since it has no direct relation with raw multimedia data, which is a collection of pixels and signals, it is very difficult to extract semantic content automatically from multimedia data [3]. To overcome these difficulties, many studies [4-9] offer different ways for extracting semantic information.

Semantic entities, extracted from multimedia data should be modeled in order to represent relations and interactions between them. Due to the complex and compound structure of multimedia data, there are many studies offering semantic content modeling such as [1, 10-13] in literature. Using an object-oriented methodology and handling uncertain / fuzzy information are some common considerations of these studies, proposing semantic information modeling.

Assumptions made by most existing semantic data models that the information stored inside the database is precise and the requirements are crisp, are mostly not valid for many knowledge intensive real world applications. Because, users are usually interested in the semantic meanings of data, which are mostly subjective contents, hence, include uncertain and fuzzy information. Moreover, most of the time, users are not able to express themselves precisely or even do not really know what they are looking for when querying multimedia data. Therefore, a multimedia database should consider uncertainty.

The logical and structural designs of databases are determined by the data types stored and query types mostly executed. Traditional database management systems support primitive types (number, character, date, etc.) and have indexing functionality on these types. However, semantic multimedia contents have multi-dimensional information that cannot be represented by such primitive objects.

Indexing multi-dimensional data is an emerging research domain. In applications of database systems for multimedia like CAD, molecular biology, medical imaging, time series processing and many others; efficient retrieval of specific information in large data sets is crucial and required as the most basic functionality. Since multimedia objects contain huge amount of information and exist in compound form, long, nested and hierarchical transactions, hence slow queries are inevitable with traditional approach. Therefore, some fast indexing mechanisms are required for locating and retrieving multimedia objects accurately in an efficient way.

To achieve a reasonable quality of service, a database management system, containing semantic multimedia content should consider all above functionalities. Besides, since the semantic content is very subjective to user and extracting all possible semantic entities is not reasonable, it should have some rule or knowledge base capabilities to infer new information using formerly extracted contents.

1.1 Motivation and Contributions

Although, there exist various researches about multimedia databases as stated in Chapter 2, most of them ignore uncertainty and fuzziness while integrating database and knowledge base technologies. Moreover, object extraction, identification, and classification, are usually neglected in such systems. In addition, most multimedia database related studies usually target a small research area in a limited domain.

The main motivation for this thesis is the need of an intelligent system for accessing desired portions of multimedia objects efficiently and accurately. In this study, a fully qualified multimedia database framework that supports annotation of objects, events and concepts as well as providing all the basic functionalities of conventional database management systems, such as indexing, querying and retrieving is presented. Compared with the existing retrieval systems, the major contributions and advantages of the proposed study are as follows:

1. A generic conceptual model for multimedia data, supporting fuzzy information is provided. The model;
   - covers hierarchical structure and temporal segmentation,
   - represents semantic content as entities and relations between them,
   - considers uncertain and incomplete information.

   Since the model is generic, it can be applied to all domains and can be extended to support domain specific entities.

2. A multimedia database architecture is proposed in a component-oriented approach. A semantic information extractor, a high-dimensional index structure and an intelligent database are conceptually gathered around a coordinator structure to build a multimedia database architecture. Thin client technology is recommended for the proposed architecture in order to support distributed nature of data and variety of multimedia application environments.

3. A sample implementation of the proposed architecture is presented. By coupling some multimedia modules, a complete system that supports automated semantic content extraction and utilizes a multi-dimensional index structure is constructed. The system is gradually developed in a component-oriented manner, such that each component can be replaced by alternative modules without affecting others. Since no domain specific modules and structures are employed, and MPEG-7 specifications are considered in all steps; the framework can easily be adapted to any domain with minor changes.

4. An ontology for football domain, describing events and concepts, based on the integrated semantic video meta-model, is developed. Relations and relevant object definitions are provided in the ontology. This ontology is

used in cooperation with the event extraction module to detect concepts and events.

5. Web services for the developed multimedia database framework are presented for interoperability issues with other systems. Any external system may benefit the querying facilities of the implemented multimedia framework using the web service interface of the system.

6. In this study, a prototype client application is developed for testing and evaluating presented architecture.

1.2 Thesis Outline

The rest of the thesis is organized as follows; in Chapter 2, studies related with the topics dealt in this thesis are explained. Information about tools and technologies utilized in this thesis are given in Chapter 3. In Chapter 4, proposed multimedia database architecture and the recommended semantic data model for this architecture are analyzed in detail. Chapter 5 introduces the empirical study executed and gives brief information about the developed system while explaining integration steps in detail. Abilities of the architecture and implemented system, created ontology and developed web services are given in this chapter. Performance tests and their results are given in Chapter 6. The last chapter provides conclusions and gives future directions.

CHAPTER 2

REVIEW OF THE LITERATURE

Researches on multimedia databases gain importance since traditional database systems are not able to satisfy users, asking information about the semantic content in multimedia applications. Since a whole multimedia database framework covers a huge study area, many studies concentrate only on a specific subject in this domain. Also, because of the reason they comprised lots of the multimedia object types (image, audio, text, temporal and spatial events), most studies are based on video objects.

In order to develop a complete framework for multimedia database, some sub-research topics, such as modeling multimedia data, semantic information extraction and multi-dimensional indexing, should be inspected. In this chapter, a review of literature survey, executed during the development of this study is presented. To clarify each subject, they are briefly explained in separate sections.

Studies, related with multimedia object modeling are explained in the first section. Some studies in this section cover, uncertainty and fuzziness in multimedia objects and some of them take knowledge intensive modeling into consideration. Second section aims to discuss researches about semantic information extraction, in other words, object and event detection based studies. Although, some researches uses complex annotation techniques, most of them are based on segmentation and classification approach. Studies, proposing index structures for multi-dimensional objects are given in the next section.

2.1 Multimedia Data Modeling

The recent education and entertainment industry are built on the digital world and most of such user interactive systems support visual and audial operations. The increasing interest on the digital media has brought out an enormous growth in the volume of multimedia data. But, these data can be useful only when applications and databases provide an effective querying ability and let end users access what they need correctly in an efficient way. Therefore, new data models that can fulfill the requirements of multimedia objects are needed. Modeling multimedia data is the first, perhaps the most important step of developing a multimedia database management system.

The information inside raw multimedia data can be categorized into three groups; *low-level information*, *semantic and syntactic information*, and *textual information* [1]. For each type of information, there are studies concentrating on modeling that type of information. While studies [3, 14, 15] emphasis on low level information extracted from multimedia data, many others focus on semantic modeling since they represent information, perceived by humans, therefore more meaningful and important.

Semantic data models can be classified in two categories; *annotation-based* models and *rich semantic* models [16]. Annotation-based models, like OVID [17], VideoStar [18], CCM [19], use relatively simple structures, such as predefined keywords, some free text, or structured data, as annotations.

The study OVID [17] presents a video data model and offers a prototype database system that handles video objects in an object-oriented manner. All the information extracted from a video material should be an object or an attribute of the object. Although [17] supports only a few object types, they can be extended with new attributes whenever necessary. Description data can be shared by *interval-inclusion based inheritance*.

VideoStar [18] is another video data model presented at early 90's. It supports *StoredVideoSegment*, *VideoStream* and *VideoDocument* classes to cover physical video segments, logical video segments and mappings between them. Supporting frame sequences and structural components let this study to handle the most popular approach for temporal segmentation of multimedia data; *shot*, *scene* and *sequence*.

VideoText [20] introduces Information Retrieval (IR) in late 90's to retrieve contents encoded in free text annotations. In their latter works, a conceptual graph is utilized to represent knowledge in free text annotations and three layers of abstraction are used to achieve *physical video data independence*, *logical video data independence* and *user view independence.*

Although, simple annotation structures give *annotation-based* models a great flexibility, they limit the data expression power and querying capabilities. Since annotated information is associated with logical segments of multimedia materials, abstract and inferred information cannot be modeled with this approach. When annotation-based models turned out to be inadequate for satisfying user requests about semantic contents, *rich semantic* models that represent semantics in a real-world manner with more complex structures became more popular. Since they are able to express abstract entities and inferred concepts, represent relationships between entities, and provide better retrieval abilities, *rich semantic* models can be said to be richer than the *annotation-based* models [16]. VIMSYS [21], VIDEX [22], Extended $ExIFO_2$ [11], AVIS [23], VideoGraph [24], $CoPaV^2$ [1], BilVideo [25] and Ekin's model [10] are some studies proposing rich semantic models.

The leading study VIMSYS [21], a model for multimedia images, has defined domain event concept as spatio-temporal relations of domain objects using four levels of abstraction*; image presentation* level*, image object* level*, domain object* level and *domain event* level.

The QBIC [5], developed at IBM Research Center, focuses on extending and improving query methods for retrieving pictorial information from large image and video databases according to supplied color, shape and texture information.

In his study [10], Ekin et al. propose a new generic integrated semantic-syntactic video model by combining shot-based and object-based structural video models with entity-relationship (ER) and object-oriented models to model semantics in multimedia documents. The model, which is an extension of ER models, aims enabling structured video search and free browsing by combining textual and low-level descriptors of content. The main entities in the model are *events*, *objects* that participate in these events and *actor* entities that describe object instances which play roles in some events.

Another study that supports semantic objects and events, AVIS [23], presents activities as types of events by categorizing them. Handling feature – sub-feature relationship and replacing sub-features with features whenever required enable AVIS system to soften queries that return no result.

BilVideo [25], another video database management system, provides an integrated support for rule-based spatio-temporal modeling. Semantic and low-level feature queries are allowed using a SQL-like textual query language. The study offers using a knowledge base component for handling spatio-temporal queries and employs an object relational database for queries on low-level features or semantic contents. A semantic data model [26], supporting objects and events is utilized in BilVideo. To represent parts of events in model, an event - sub-event relationship is used.

The semantic meaning of content is subjective to user. Content extraction approaches, whether human driven or automatic, may found different semantic information from same source. There are a few studies that handle fuzziness and give support for uncertainty in their conceptual models for multimedia materials. Studies [11, 13, 27-29] try to handle this uncertainty by supporting fuzzy attributes, objects or relations in their model. [13, 29] uses an extended version of UML to represent fuzzy information in conceptual model. The study, adapts IFOOD architecture [30] into video database applications. IFOOD is an architecture which connects a fuzzy knowledge base with a fuzzy object-oriented database. Thus, [13] supports not only inferring fuzzy information using knowledge base, but also storing semantic content with their uncertain attributes as objects. Addition to attribute level uncertainty, the proposed model handles fuzziness that might occur in object itself or the relations between them. Even events might have fuzzy information.

[11] is based on $ExIFO_2$ data model, and being mapped to logical FOOD [31] model, which is a similarity-based fuzzy object-oriented model. The study supports representation of uncertainty at *attribute*, *object/class* and *class/subclass* level, for multimedia objects.

Study [27] proposes a data model that summarizes semantic content by examining features of video, audio and superimposed text. In the mentioned study, three levels of abstraction is used for representing semantics; a *context layer* for describing content independent features, a *concept layer* for abstraction of key concepts, and a *concept measure layer* for describing concepts using fuzzy approach to find semantic similarity.

Although there are many other researches on multimedia semantic content modeling, some leading ones are briefly explained above. Common considerations of the most approved studies are; using an object-oriented approach and taking intelligence into account as well as paying attention to uncertainty in multimedia objects.

2.2 Semantic Information Extraction

Most featured studies, offering a semantic data model, use the advantage of expressing semantic content with real world fundamental entities, such as concepts, objects, events and relations between them. Thus, they enable users to construct queries with concrete objects appearing in images or videos. However, this expressive approach brought out the necessity of detecting and extracting concepts, objects and events from raw multimedia content. In this section, some studies related with this significant subject are presented.

Semantic object extraction is mostly a matter of segmenting a piece of data and classifying or categorizing them. Many researchers execute this step in a brute-force manner, by checking all the possibilities and trying to find most possible entities.

Classifying raw data as a whole and finding the type of it may reduce the amount of time spent in entity classification step. Jadon and his friends, use video-structure based syntactic features (such as shot durations, editing style, camera work, etc.) to classify video sequences in their study [32]. They offer an evolutionary learning fuzzy rule-based system for categorizing video materials. Study [33] presents a video genre classification approach using analyses on the connection between video genres and the statistical characteristics of dominant events. In [34], a method for classifying talk and game shows, and segmenting these videos into host and guest shots by using video structures is proposed.

Detecting and subtracting background textures from images facilitate object and event extraction processes by reducing the amount of regions to be processed. Study [35], proposes a new background subtraction algorithm, which is not affected from the illumination factor. Since the study considers changes in illumination conditions, the proposed algorithm can be applied not only to indoor environments, but also to outdoors with natural illumination.

Basic semantic entities, objects and events have different behaviors and structures. Objects are mostly static entities, only attributes or roles of which change in a multimedia data. A single image or a few frames are adequate for extracting them. On the other hand, events have spatial and temporal characteristics, therefore, mostly a chronicle frame sequence are required for detection and annotation. Methods, used for extracting objects and events, differ from each other.

2.2.1 Object Annotation

Image is the smallest visual type among raw multimedia types. An ordered composition of which constitutes video objects. Since they are composed of collections of pixels, there is no semantic information that can be obtained directly in raw image data. Therefore, a powerful way of mapping regions of these pixels to semantic content is required.

Because of the reason that there is no straight way for mapping raw data into semantic information , it is very difficult to extract objects from raw multimedia data [3], which makes this area very challenging. Since object extraction is a complex and time consuming process, many research groups center their work on extracting specific objects. In [36], from digging text and timing information, to scene and face detection, a various information aspects are investigated to find a specific person in broadcast news video. Using the combination of background subtraction and edge detection methods, the study [37] is specialized for detecting vehicles in traffic surveillance and control systems.

Determining important frames, moreover finding the boundaries of possible objects, thus deciding whether a group of pixels should be classified as an object or background, may reduce the computational complexity in object extraction. To lessen the processed frame count, a Genetic Algorithm (GA) based method is proposed in [38], by extracting frames carrying more meaningful information using cross-correlations between all combinations of feature vectors. In another study, genetic algorithm is used for object localizations in complex scenes [39]. A more recent study [40], introduces fuzzy topology into object extraction area to distinguish objects from background.

Information, such as dominant color or region shape which can directly be extracted from the pixel information with simple procedures, is called as *low-level features*. Extracting these features is the first step in many studies focusing object extraction. MPEG-7 [41], formally named as Multimedia Content Description Interface, provides various simple low-level features and standardizes descriptions and the ways of structuring them. Besides, providing some Descriptors (Ds) to describe low level features, MPEG-7 also defines Description Schemes (DSs) to specify descriptors that can be used in a certain description and relations between DSs. Also, Description Definition Language (DDL) allows researchers to define their own Descriptors and Description Schemes.

In the first days of low-level feature extraction, many studies concentrate only on extraction of a specific feature. While some studies focusing only on color information, some studies just concentrated on shape information. A recent study

[42], focusing approaches and trends in content-based image retrieval, shows that using different features together with better calculations revealed better results.

A recent study [43], which tries to mimic human brain, proposes architecture for an association-based image retrieval system, using feature vectors to represent each image. Low level features are combined using weight matrices in the generalized bi-directional associative memory (GBAM) which keeps associations between feature vectors.

Another important point in object extraction, usually referred as semantic gap, is mapping extracted low-level features into high level semantics. In other words, after founding dominant color as yellow and the region shape as circular, obtaining the object *sun* by processing these low-level data.

The most popular method for handling semantic gap is assigning weights to the low-level features and using these values for mapping issues. Manual or automatic methods can be applied to find to weights of each feature.

The study [44] tries to fill the semantic gap by using a knowledge base. With the help of a domain expert, a rule-knowledge base is constructed during the learning phase, and this knowledge base is used in annotation step. Using fuzzy logic and rule mining techniques to approximate human-like reasoning resulted in more robust and accurate annotations. Study [45] proposes an ontology based object annotation system using semantic concept classifiers. They built their approach to learn rules automatically using Semantic Web Rules Language (SWRL) and combine this knowledge into the ontology. Relations of concepts' are utilized to get better performance from concept detectors.

The study [2] takes object extraction process as a categorization problem and uses GA for finding dominant features of representative images for object categories. In learning phase, Best Representative and Discriminative Features (BRDF) of objects are found, and the set of representative objects and their features are utilized by GA based classifier. Self improvement with genetic operations and supporting fuzzy decisions by making weighted multiple categorizations are discriminative properties of this study.

2.2.2 Event Annotation

Automatic event annotation is another challenging area in semantic information extraction from multimedia data. Although many earlier studies skip this step or utilize user driven techniques, rapidly increasing huge amount of multimedia data forces researchers to find ways of detecting events automatically. Approaches for event recognition are mostly focused on specific event types since researchers restrict their studies with particular domains [46].

Due to temporal characteristics, a group of ordered frames is required in event extraction. Therefore, in many studies [13, 18, 22, 23], a well-known method for representing temporal segmentation of video data is used; *shot-scene-sequence* hierarchy. Trying to classify a single or a few contiguous shots into events is the dominant approach in studies using this hierarchy.

In the study defining events as long-term temporal objects [47], a simple statistical distance measure is designed, based on the behavioral contents in video pieces and this measure is used for classifying events. Study [48] provides a solution for soccer domain and proposes an evolutionary event detection framework by utilizing neural networks with multimodal analyses. In [49], some audio-visual features indicating specific events are selected, and a framework, using these features for extracting events from sport videos, is proposed.

Study [50] proposes a novel approach to detect and classify key events in various sport videos, by using a hybrid method, which integrates statistics with logical rule-based models. Complementing deficiencies of using typical patterns of audio-visual features in sports domain with human observation and heuristic knowledge make this method a robust, less subjective and a generic approach for event detection.

A Gaussian Mixture Model (GMM) is built and presented to detect events using semantic features that can accurately reflect the event attributes, in study [33]. An event adjustment strategy is also proposed according to the analysis on the GMM structure and pre-definition of video events.

Different from other studies, heavily rely on audio/visual features, study [51] incorporates web-casting text into sports video analysis to extract events and detect event boundaries. Also the proposed study is capable of producing personalized summary from both general and specific point of view. Another study using web-casting texts [52], perform multimodal analyses for finding important events in a basketball game. The proposed framework contains four major parts: *web-casting test analyses*, *broadcast video analyses*, *text/video alignment* and *semantic annotation and indexing* for personalized retrieval.

Since, they restrict the types of events to be recognized in domains and used for inferring information from pre-extracted ones, some studies proposes usage of ontology for object and event extraction. Inyaem and his friends, propose a fuzzy-ontology based event extraction method for terrorism domain [53]. [9] presents a top level ontology using spatio-temporal relations of pre-defined semantic entities to extract events. The study [46] offers an ontology-based semantic content model for semantic content and event extraction. To reduce the computational cost, domain ontology is enriched with rule definitions.

2.3 High-dimensional Index Structures

Most of the current database management systems provide index structures for efficient accessing to data. For primitive types in database management systems, indexing and querying problem can be considered as efficiently solved. But indexing high-dimensional data is an emerging and problematic research domain. As dimension increases, performance of multi-dimensional index structures decreases, usually called the *curse of dimensionality*.

Algorithms for indexing high-dimensional data are mostly based on the principle of hierarchical clustering of data, in other words, partitioning search space into sets and pruning some of them in search time [54]. These indexing techniques can be classified into two groups; *data-partitioning index methods* that divide data space according to distribution of data; and *space-partitioning index methods* that partition search space in pre-defined sizes, regardless of data in these segments [55].

Most known data-partitioning methods derive from the well known index structure, R-Tree [56]. The method is originally designed to cope with only two-dimension data in Geographical Information Systems. Leaves of the tree refer to minimum bounding rectangle of the actual object. An extension of R-Tree, SS-Tree [57] is designed to handle multi-dimensionality and improve search performance by using minimum bounding spheres. Since bounding spheres usually need larger volumes than bounding rectangles, SR-Tree, [58] is proposed to handle this deficiency. SR-Tree uses an intersection of a bounding sphere and a bounding rectangle.

Another approach, TV-Tree [59] is based on the observation that some dimensions are more discriminative than others. Therefore, they classify dimensions as; ignorable dimensions, always used dimensions and dimensions used for fine tuning. Although this study is very successful in rather high dimensions, it requires a good knowledge of data distribution along each dimension.

Although some space partitioning index methods, grid-file [60], perfect kDB-Tree [61], Hybrid-Tree [62], SH-Tree [63], are proposed for handling multi-dimensional data in space granularity, these methods are inefficient for high-dimensions. Since this strategy partitions the search space rather than data itself, they mostly have problem of indexing large empty spaces. Besides, when retrieval of objects near cell boundaries is required, all the neighboring cells are to be examined, resulting in a huge growth in search path.

Some studies on multi-dimensional indexing, such as KPYR [64], a Pyramid-Technique [65], try to reduce dimension and map the information to a single-dimensional value. M-Tree [66] and Slim-Tree [67], uses distance calculation based algorithms. Relative distances of objects are calculated and the index structure is built using these values rather than the complex features of objects, with a cluster hierarchy on the top [54]. Since overlaps decrease the performance of index structures, Slim-Tree uses an algorithm, called as Slim-Down to reduce overlaps.

BitMatrix [54] tries to make an approximation and built a representative vector, significantly smaller than the original one. Since it is highly configurable, the tradeoff between precision and speed can be controlled with various parameters. Kernel VA file [68] tries to improve the performance of sequential scan by using two different data sets; one for real data and one for approximations of these data.

The study [69], proposes a Content-Based Video Retrieval (CBVR) system, which examines Slim-Tree and BitMatrix algorithms for efficiently executing low-level feature queries on video objects. In the process of combining features and defining distance function weights, Ordered Weighted Aggregation (OWA) operators are used. This research is specialized for indexing low-level features, what makes this study a good candidate for multimedia data indexing.

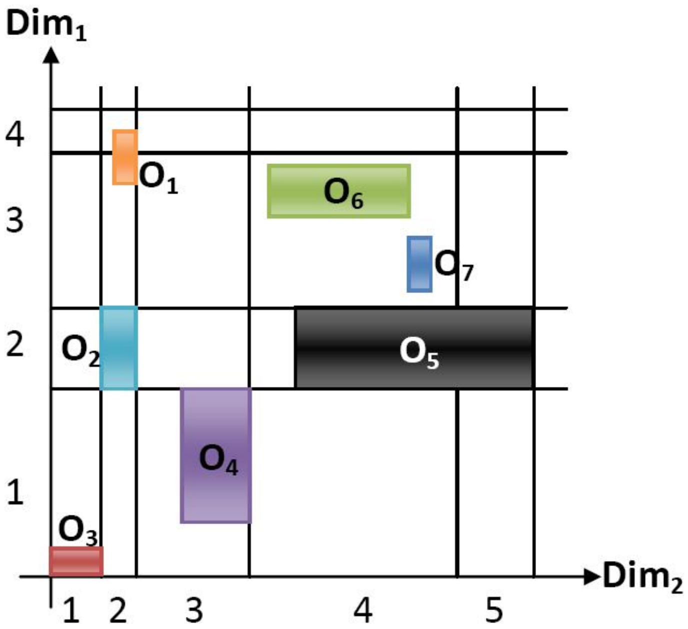


Figure 2-1 Clustering Sample for BitMatrix

Table 1 BitMatrix Representation

	Dim_1				**Dim_2**					
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**	**5**	
O_1	0	0	1	1	0	1	0	0	0	0011 01000
O_2	0	1	0	0	0	1	0	0	0	0100 01000
O_3	1	0	0	0	1	0	0	0	0	1000 10000
O_4	1	0	0	0	0	0	1	0	0	1000 00100
O_5	0	1	0	0	0	0	0	1	1	0100 00011
O_6	0	0	1	0	0	0	0	1	0	0010 00010
O_7	0	0	1	0	0	0	0	1	0	0010 00010

CHAPTER 3

BACKGROUND KNOWLEDGE

In the scope of this study, a multimedia database architecture is proposed, and a prototype database framework as well as a client application are developed. In design and implementation stages, some external tools and different technologies are utilized for the activities that are out of the scope of this research.

In this chapter, the fundamental concepts about the ideas that the study is built on and external tools used in this thesis are briefly explained. As our system is based on MPEG-7 concept, the first section is reserved for brief information about MPEG-7 and tools based on MPEG-7 technology (MPEG-7 reference software (eXperimental Model, XM) and IBM MPEG-7 Annotation Tool). DB4O database and JESS knowledge base, which are used in core database structure as storage and fuzzy inference engine, are explained in subsequent sections. Information about genetic algorithm and ontology, used in semantic information extraction components, JMF and JavaWS technologies, utilized in the prototype client application are provided in following sections.

3.1 MPEG7

3.1.1 Introduction

MPEG-7, a multimedia content description standard, is introduced as an ISO/IEC standard by MPEG (Moving Picture Experts Groups). It is formally called as *Multimedia Content Description Interface* and provides a rich set of standardized tools to describe multimedia content. The audiovisual (AV) information within the scope of MPEG-7 can be consumed both by human users and digital systems.

Being different from prior standards (MPEG–1, MPEG–2, MPEG–4), which focus on coding and presentation of audio-visual content, MPEG–7 focuses on multimedia content description that supports some degree of interpretation of information meaning, which can be processed by digital devices [41]. It was designed to standardize:

- *Descriptors (Ds),* which describes basic contents of multimedia, based on audio-visual information, such as color, texture, shape, frequency and motion characteristics that constitute the bases of CBIR systems,
- a set of *Description Schemes (DSs),* which specifies available descriptors that can be used in a given description and the relations of stated descriptors with the other DSs,
- a language to specify these schemes, called the *Description Definition Language (DDL)* and
- a scheme for coding the *Ds* and *DSs*.

3.1.2 Scope of MPEG-7

MPEG–7 focuses on the standardization of a common interface for describing multimedia data and representing information about the content, not the content itself. The scope is to define the representation of the features, related to AV content. Any application dependent issue is outside the scope. Therefore, neither feature extraction nor query and retrieval process is in the scope of MPEG–7. However, because of some interoperability issues, it also specifies extraction process at some degree. To summarize, main goal is to make audiovisual data as searchable as text.

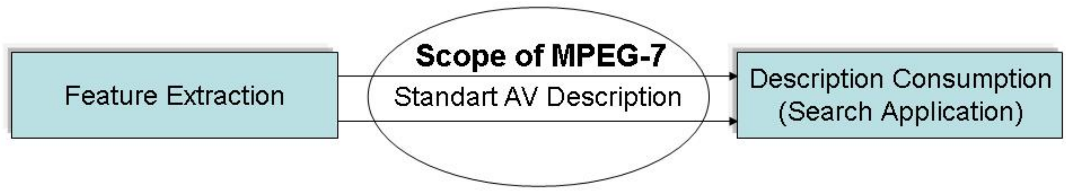


Figure 3-1 Scope of MPEG-7

3.1.3 MPEG–7 Parts and Descriptors

The MPEG-7 Standard consists of the following parts [41]:

- MPEG-7 Systems – the tools needed to prepare MPEG-7 descriptions for efficient transport and storage and the terminal architecture.
- MPEG-7 Description Definition Language - the language for defining the syntax of the MPEG-7 Description Tools and for defining new Description Schemes.
- MPEG-7 Visual – the Description Tools dealing with (only) Visual descriptions.
- MPEG-7 Audio – the Description Tools dealing with (only) Audio descriptions.
- MPEG-7 Multimedia Description Schemes - the Description Tools dealing with generic features and multimedia descriptions.
- MPEG-7 Reference Software - a software implementation of relevant parts of the MPEG-7 Standard with normative status.
- MPEG-7 Conformance Testing - guidelines and procedures for testing conformance of MPEG-7 implementations
- MPEG-7 Extraction and use of descriptions – informative material about the extraction and use of some of the Description Tools.

- MPEG-7 Profiles and levels - provides guidelines and standard profiles.
- MPEG-7 Schema Definition - specifies the schema using the Description Definition Language.

MPEG-7 provides tools and structures for describing both visual and audial content [Table 2]. MPEG-7 Standards Overview documentation [41] gives detailed information about all these descriptors. Although only a few of them are used in the implementation of the proposed architecture, any descriptor can easily be adapted into the system. In the following part, descriptors used in the implementation of prototype application are briefly explained.

- **Color Layout**: Among seven color descriptors, color layout represents the spatial distribution of colors of an image in the frequency domain in a very compact form. This compactness allows it to be used in index structures with small computational costs. Besides, it also provides high-speed image-to-image and sequence-to-sequence matching, which requires so many similarity calculations. Since it captures the layout information of colors, this descriptor allows very friendly user interface using hand-written sketch queries. No dependency on image/video format, resolution and bit-depths exists for this descriptor. It can be applied to whole image and even to any unconnected parts of an image with arbitrary shapes.
- **Dominant Color**: A small number of representative colors (up to 8) are enough to characterize the color information of an image or a specific region. Such compactness makes this descriptor a good candidate for index structures. Therefore, this descriptor is most suitable for representing color information of objects. To extract a few representative colors, color quantization is used and the percentage of each quantized color is calculated correspondingly. A spatial coherency on the entire descriptor is also defined, and is used in similarity retrieval.
- **Region Shape**: By capturing all the pixel distribution of a shape/region, this descriptor can be used in describing shapes. Not only simple ones, but also

complex shapes with multiple regions, possibly the ones with holes, can be described. Its small size, fast extraction time and low order of computational complexities for matching ability make this descriptor suitable for shape tracking in images and videos [70].

- **Edge Histogram**: This descriptor represents the spatial distribution of five types of edges in an image; four directional edges (vertical, horizontal, 45° diagonal, 135° diagonal), and one non-directional edge (isotropic). Since edges play a great role in object detection, this descriptor can be useful for image-to-image matching (by example or by sketch). When used with the conjunction of other descriptors, such as color and shape descriptors, it may significantly improve the retrieval performance. Due to low computational cost, it is suitable for CBIR or retrieval systems based on textures.

3.1.4 MPEG-7 Reference Software (eXperimental Model)

MPEG-7 reference software (eXperimentation Model, shortly XM) is the tool which has ability to extract low level information from video data, using MPEG-7 descriptors. It generates conformant MPEG-7 bit streams / DDL streams. Most of the Descriptors and Description Schemes are implemented in XM software. After loading data, the software extract low-level features and after encoding descriptions, it produces a file containing low-level information [41, 71].

XM software can also be used for distance calculations of similar data. After building a database containing extracted low-level information, the tool has the ability to calculate distance values between each data in the database and the given one.

MPEG-7 reference software is used both in annotation module and index mechanism in this study. While, annotation module utilizes this tool for obtaining low-level features and determining distance values at classification step, index mechanism uses extracted features for building index structure as well as calculating distance of objects.

Table 2 MPEG-7 Descriptor List

Type	Feature	Descriptors
Video	Color Descriptors	Color Space
		Color Quantization
		Dominant Color(s)
		Scalable Color
		Color Layout
		Color-Structure Descriptor
		GoF/GoP Color
	Texture Descriptors	Homogenous Texture Descriptors
		Texture Browsing
		Edge Histogram
	Shape Descriptors	Region Shape
		Contour Shape
		Shape 3D
	Motion Descriptors	Camera Motion
		Motion Trajectory
		Parametric Motion
		Motion Activity
	Localization	Region Locator
		Spatio Temporal Locator
	Others	Face Recognition
Audio	Silence	Silence
	Timbral Temporal	Log Attack Time
		Temporal Centroid
	Basic Spectral	Audio Spectrum Envelope
		Audio Spectrum Centroid
		Audio Spectrum Spread
		Audio Spectrum Flatness
	Basic	Audio Waveform
		Audio Power
	Signal Parameters	Audio Harmonicity
		Audio Fundamental Frequency
	Timbral Spectral	Harmonic Spectral Centroid
		Harmonic Spectral Deviation
		Harmonic Spectral Spread
		Harmonic Spectral Variation
		Spectral Centroid
	Spectral Basis	Audio Spectrum Basis
		Audio Spectrum Projection

3.1.5 IBM MPEG-7 Annotation Tool (VideoAnnEx)

The VideoAnnEx annotation tool provides annotation of video sequences with MPEG-7 metadata. It performs shot detection, when an MPEG video sequence and corresponding shot segmentation file are provided. The input video sequence is segmented into smaller units by detecting the scene cuts, dissolves and fades. Each shot in the video sequence can be annotated with static scene descriptions, key object descriptions, event descriptions and other lexicon sets. Descriptions are associated with each video shot and stored as MPEG-7 descriptions in an output XML file for later use. The tool also has the ability to process pre-executed annotations for loaded video sequences [72]. An example screenshot of the IBM MPEG-7 Annotation Tool is shown in Figure 3-2.

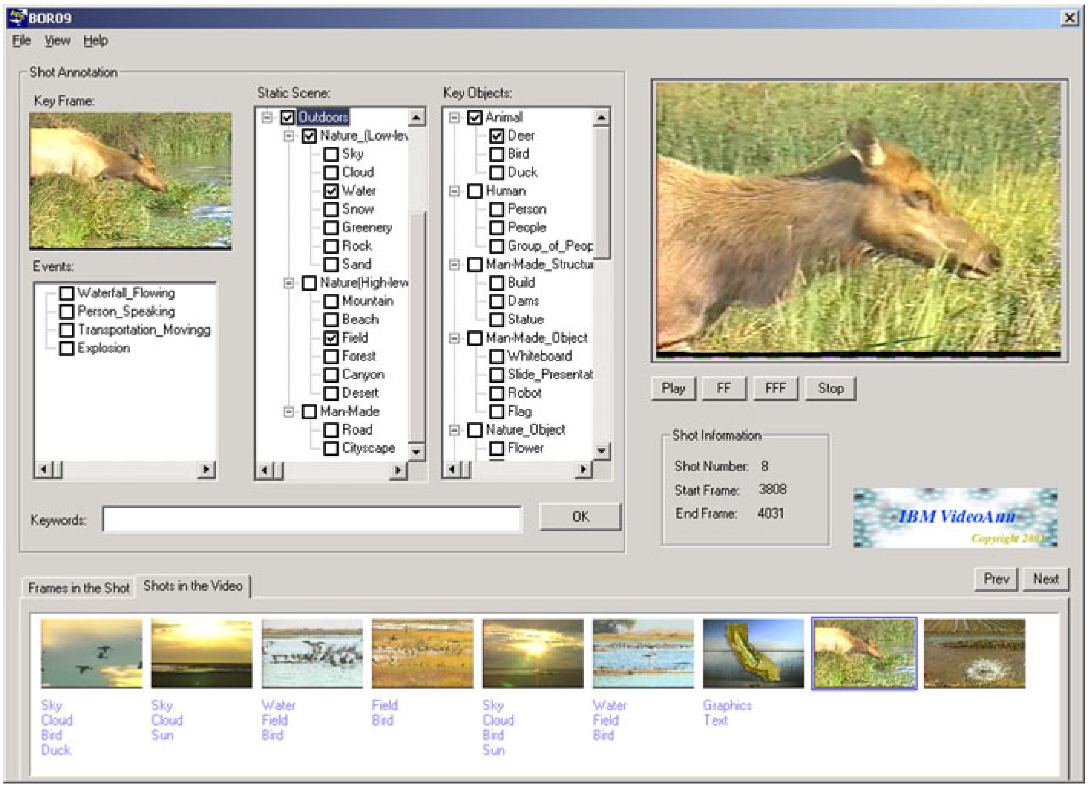


Figure 3-2 IBM Annotation Tool [72]

3.2 Java Web Start (JavaWS)

Java Web Start (also known as JavaWS), which is a framework developed by Sun Microsystems, allows users to start applications for Java Platform, directly from the Internet using a web browser. Unlike Java applets, JavaWS applications do not run inside the browser. Although it can be configured, the sandbox in which JavaWS applications run need not have many restrictions as Java applets [73].

Using JavaWS, a single Java application may be placed on a web server for deployment to a wide variety of platforms. Without going through complicated installation procedures, JavaWS provides the power to launch full-featured Java applications with a single click. From the user's point of view, it functions just as native applications. The launched applications are locally cached, for improved performance. Once installed and arranged, it enables offline usage of web based applications [74].

JavaWS allows developers to determine the security model used for applications. By default, JavaWS applications run in a restricted environment, known as a sandbox. You can also provide functionality that goes beyond what is allowed in the sandbox by signing application's JAR files. Even configuring security restrictions, interacting with the clipboard and working with the files on the computer are allowed. Automatically update application and JRE versions is another functionality that makes the JavaWS the best alternative to web based applications [74].

3.3 Genetic Algorithm

Genetic algorithm (GA) is an adaptive method used in finding exact or approximate solutions to search and optimization problems occurring in many fields; from bio-informatics, chemistry and physics to computational science, engineering, economics and lots of others. It is a powerful search technique and categorized in global search heuristics [75]. Using techniques inspired by evolutionary biology, such as inheritance, mutation, selection, and crossover (also called recombination), Genetic Algorithms hold the idea of natural selection (survival of the fittest) which suggests more successful solutions to live whereas others to die.

Evolution can be defined as slow changes or modifications on inherited attributes of species, whereas attributes are the expression of genes that are transferred from ancestors to offspring. In computer science, Genetic Algorithms generally follow evolution steps; reproduction, mutation, recombination and natural selection. A typical genetic algorithm requires:

- **a genetic abstract representation of the solution domain**, which encodes properties of individuals into lists of bits, integers, trees or pre-defined objects, each item of which represents a gene of the chromosome,
- **a fitness function to evaluate the solution domain**, which measures the quality of the represented solution.

Once genetic representation and the fitness function defined, GA proceeds to initialize a population of randomly generated individuals. It improves population through repetitive execution of mutation, crossover, inversion and selection operators. In each generation, the fitness of every individual in the population is evaluated, some of them are stochastically selected and they are randomly modified to form a new population, which is replaced with the worst-ranked population. Whenever a satisfactory fitness level has been reached or a limit of executions is exceeded, the algorithm terminates. A flowchart for genetic algorithm is given in Figure 3-3.

In this study, genetic algorithm is employed in the object extraction module, which defines the whole process as a classification problem and uses a Genetic Algorithm based classifier for classification.

3.4 Ontology

In computer and information science, ontology is defined as a formal representation of the knowledge by a set of concepts and the relationships between these concepts within a domain. It is used to reason about the properties of that domain, and even sometimes, it is utilized for describing the domain. Ontologies are used in artificial intelligence, semantic web, software engineering, bio-medical informatics, and information architecture as a form of knowledge representation about the world or some part of it [76].

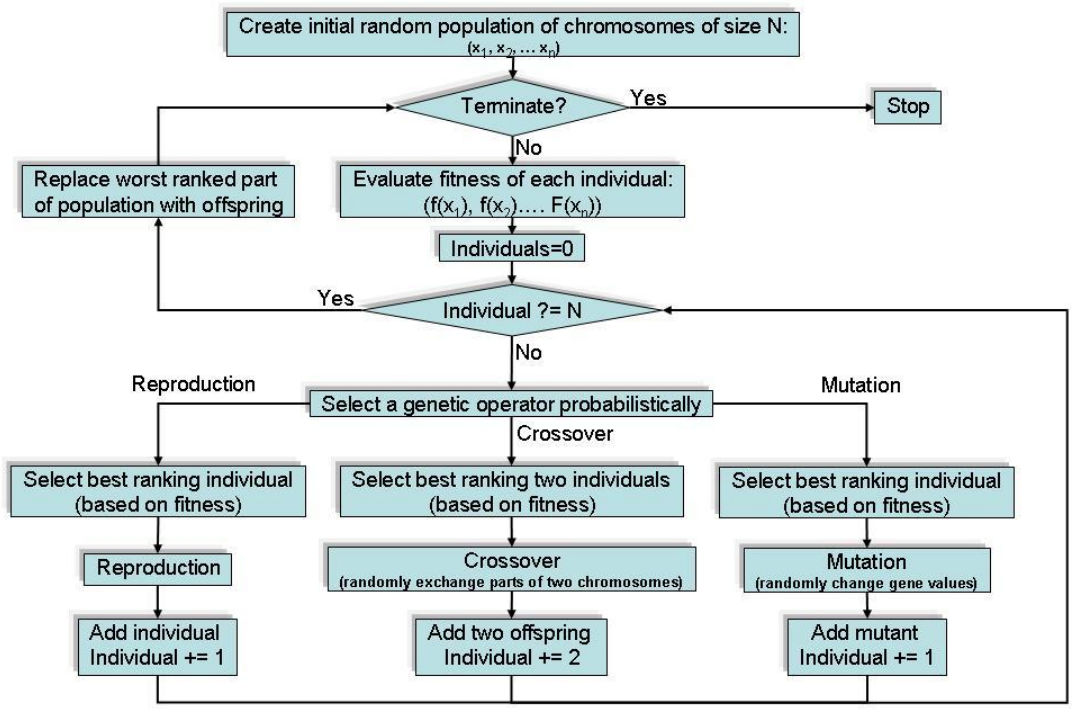


Figure 3-3 A Simple Flow Chart for Genetic Algorithm

The body of formally represented knowledge is based on conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them [77]. A conceptualization is an abstract, simplified view of the world that needs a formal representation for some purpose. Since it provides a shared vocabulary, which can be used to model a domain, ontology can be defined as formal specification of a shared conceptualization. It includes machine-interpretable definitions of basic concepts in the domain and relations among them. In other words, it defines what the data is and it's relation to everything else.

We develop ontology:

- To make users agree on the meaning of terms,
- To analyze domain and make sure the domain assumptions are explicit,
- To enable reuse of the domain knowledge and separate it from the operational knowledge.

In summary, ontology describes the logical structure of a domain; its concepts and the relations. Therefore, they should be constructed by a domain expert to guarantee consistency and accuracy. During the construction, it should be kept in mind that ontologies should be open, dynamic, scalable, interoperable, and easily maintained [76].

3.5 Fuzzy Logic

When we move towards the information era, human knowledge becomes increasingly important. For many practical systems, the origin of important information can be classified into two categories: the ones coming from human sources who describe their information in natural languages, and sensory measurements working according to physical laws. An important task is to combine these two types of information, which revealed the need for formulating human knowledge and finding a way to represent it with a machine readable structure. In other words, transform human knowledge into a mathematical model [78, 79].

The linguistic terms, like *cold, warm, old, young* or even some color terms, like *red* and *yellow*, cannot be easily mapped to precise values in digital world. One can never say that the boundary of *old* or *young* is a specific age. Such a sharp distinction may classify two people into two separate datasets; even they were born on only one or two adjacent but different days. Similarly, if you draw a crisp boundary for the term red, where as a color having value slightly under the limit is named as red, the other being slightly over is classified as another name, even human eyes cannot catch the difference. As a result, since the real world is too complicated for precise descriptions to be obtained, the information humans exposed and inferred are blurred, vague and confusing. Covering also uncertain and incomplete information, Dr.Lotfi Zadeh proposed the idea of fuzzy logic, using his earlier study over fuzzy sets [78]. In this study, to handle the partial truth and linguistic variables, an extended version of boolean logic is introduced, covering approximations as well as precise values.

Although the fuzzy systems theory operates on vague, uncompleted and blurred information, the theory itself is very precise. Fuzzy systems are precisely defined and completely deterministic, what makes them applicable to a wide variety of fields ranging from databases, signal processing and expert systems to computer software, pattern recognition, communication, etc. However, the most significant applications have concentrated on control problems.

The fuzzy theory covers a wide range of theories and techniques. The core technique is base on four basic concepts:

- **Fuzzy Sets:** sets with smooth boundaries, which are generalized version of classical set, allowing partial membership,
- **Linguistic Variables:** variables whose values are both qualitatively and quantitatively described by a fuzzy set,
- **Possibility Distributions:** constraints on the value of a linguistic variable imposed by assigning it a fuzzy set,
- **Fuzzy IF-THEN Rules:** a knowledge representation scheme for describing a functional mapping or a logic formula that generalizes an implication in two-valued logic.

Since, the semantic meaning of data is what you understand from it, thus very subjective, it is very difficult to express this knowledge with precise values. Therefore, to support linguistic terms in proposed model and handle fuzzy information in developed architecture, fuzzy processors are utilized in core database module.

3.6 Java Media Framework

Java Media Framework (JMF) is a java library that allows developers to embed time-based multimedia data controllers (audio, video, etc.) into cross-platform Java multimedia applications. JMF can capture, play, stream and transcode multimedia objects. It offers some optional performance packages for platform specific functions.

JMF supports lots of media types; such as AVI, MIDI, MPEG-1 Video, MPEG Layer II Audio, QuickTime, and Wave. It also has supports for many codec. Although they are limited, the support for RTSP and RTP enable JMF to stream multimedia data over internet. It works fully compatible with java applets and JavaWS applications, what makes it a common multimedia processor alternative for web based applications.

A JMF player been is developed in the scope of this study. Various multimedia control facilities (play a pre-determined portion of video, select a region or regions on a frame, display a single frame/image etc.) are accomplished using this bean.

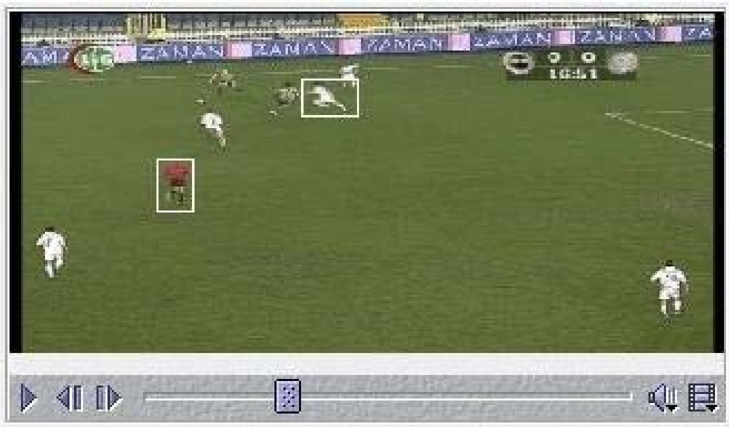
3.7 DB4O

3.7.1 Using Object-Oriented Database Management Systems

In object-oriented (OO) programming paradigm, usually, storing and accessing objects are the bottle neck of the system. While developing with an OO language, if you use relational database management systems (RDBMS), complex data relations, in other words multiple cross-references can result in complicated and difficult-to-maintain code [80]. Most of the time, you have to write object-to-relational mapping code for storing objects in RDBMS. Similarly, when an object is to be retrieved from relational database, since objects and their properties are usually stored in a normalized form, hence distributed in various fields, a group of time consuming retrieve and assemble functions are to be executed. Also, when dynamic class structures are used as in agile development environment, for each minor modification, you may have to change the schema and alter some queries to handle schema change.

(a) Message Viewing

(b) Loading Material

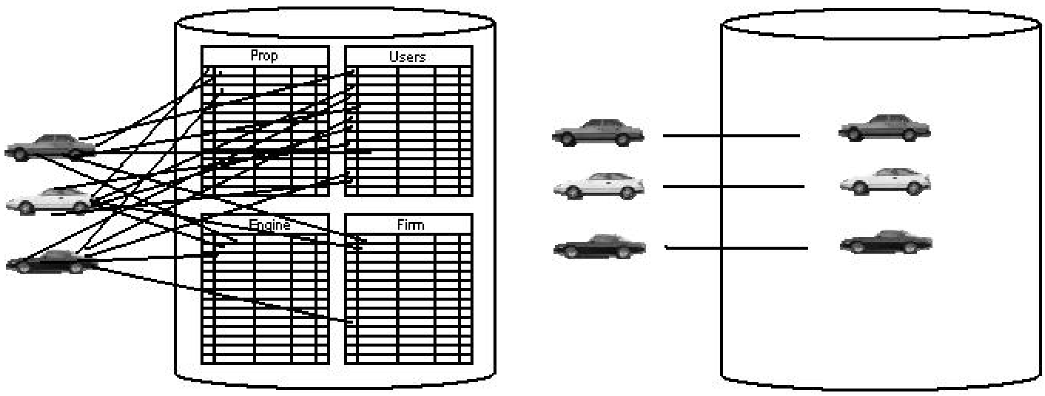


(c) Displaying and Region Selection

Figure 3-4 JMFPlayer Bean Interface

The aim of object-oriented database management systems (OODBMS) is to handle deficiencies of RDBMSs in object handling approach. Although, in earlier days, some object-relational database systems are offered for handling objects in relational approach, they could not reach the compactness and convenient usage of OODMSs. Since objects exist as whole entities in database, storing, retrieving and accessing to them, can be executed with single calls in OODBMSs, even they have compound structures or parent-child hierarchies.

"*Using tables to store objects is like driving your car home and then disassembling it to put it in the garage. It can be assembled again in the morning, but one eventually asks whether this is the most efficient way to park a car*." [81]



(a) (b)

Figure 3-5 Object storing behaviors: (a) RDBMS (b) OODBMS

3.7.2 What is DB4O?

DB4O is an open-source object-oriented database, providing a strong integration with object-oriented programming languages, like Java, .Net and MONU. As in OODBMSs, it eliminates the translation code which most OO developers should deal with. It provides high performance, cross platform, simple and easy-to-manage store and access environment. By using DB4O, there is no need to design an additional database schema since the class model becomes the database schema of application.

3.8 JESS

Jess [82], a library used for developing intelligent systems, called Expert Systems, is developed by Ernest Friedman-Hill at Sandia National Labs. It was first written in late 1995. It is a rule engine for Java platform that supports development of rule-based expert systems, which are set of rules that can be fired on a collection of facts/rules. It is inspired from another expert system shell, CLIPS [83], and entirely written in Java. Therefore, all java libraries and Java APIs are visible inside Jess, which makes it more dynamic and extensible than CLIPS.

Since being written in java, Jess can easily be embedded into the applications developed with java technology. If system logic is expressed with some kind of rules written using its own language or XML, Jess can process them like an interpreter. These rules can do anything that the Java programming language can do.

Expert systems require extensive pattern matching workforce during their execution; check first rule against the known facts in the knowledge base, fire that rule if necessary, move to the next rule and execute these steps until end of all rules, and loop back to first rule when finished. For even moderate sized rules and facts knowledge bases, this naive approach performs too slowly.

Jess uses an enhanced version of the Rete [84] algorithm to process rules. Rete, which was designed to speed up the pattern matching problem, is a very efficient algorithm for solving difficult many-to-many matching problem. Although, speed of JESS theoretically independent from the number of rules in the system, since it is designed to utilize memory for increased speed, in very big expert systems, this algorithm tends to run into memory consumption problems.

There is also a JESS extension, written with the fuzzy library FuzzyJ toolkit in Java, enables fuzzy rules to be used in JESS. A sample for a fuzzy rule can be:

If the weather is not so hot, operate air conditioner rarely.

Like any typical rule engine, Jess architecture consists of a rule/knowledge base, an inference engine and a working memory. *Agenda* component decides the checking order of rules in inference engine and *Pattern Matcher* component determine which rules to fire.

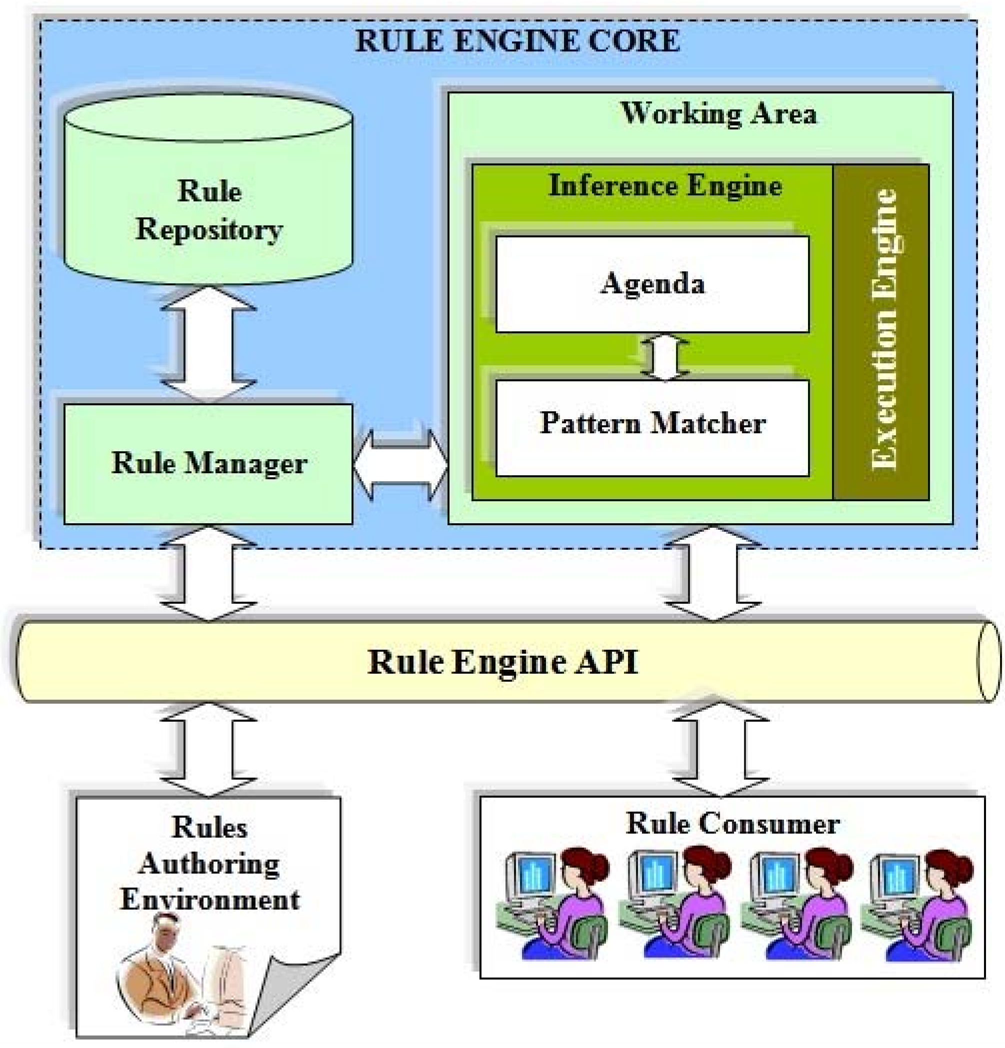


Figure 3-6 JESS Architecture

3.9 Web Services

Web service is a self contained, self describing, modular application that can be located, published, and invoked across the Web. From simple requests to very complicated business processes and transactions, web services can perform various functions. It is a software system, designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. [85]

Since all major operating system (OS) platforms can access Web using browsers, an interaction between these different platforms is inevitable. Web services are built around the XML technology and with HTTP standard that they can be used by any browser on any platform. They provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks.

Main elements of Web Service platform are:

- **SOAP**, Simple Object Access Protocol; an XML based communication protocol and language independent format for exchanging messages via internet,

- **UDDI**, Universal Description, Discovery and Integration; a directory of web service interfaces described by WSDL and used for storing information about web services

- **WSDL**, Web Services Description Language; an XML based language that describes and locates web services.

The architecture of web services is given in Figure 3-7.

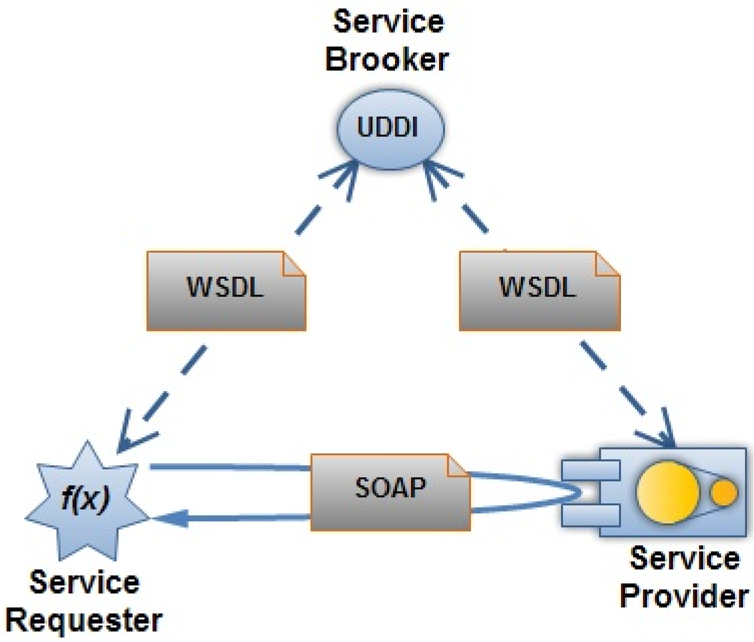


Figure 3-7 Web Service Architecture [85]

CHAPTER 4

INTELLIGENT FUZZY OBJECT-ORIENTED MULTIMEDIA DATABASE

There are database management systems that support multimedia objects by some add-ons or extensions. Although these systems are capable of storing and fetching multimedia materials according to their textual definitions, they are far away from satisfying user requests such as semantic querying and efficient retrieval of multimedia data. In this thesis, an intelligent fuzzy object-oriented multimedia database framework (IFooMMD) that is capable of performing complex and similarity based operations as well as basic input and retrieval functions, specialized for multimedia objects is proposed.

A complete multimedia database framework should be able to extract semantic information, in other words, annotate entities such as concepts, objects, actors and events; store these entities, and retrieve them whenever required in an efficient way. Since the focus of this study is to build a fuzzy object-oriented database architecture for handling multimedia materials, in this chapter, the proposed semantic data model and the conceptual framework architecture are briefly explained. Implementation details are presented in following chapter.

4.1 The IFooMMD Conceptual Data Model

Conceptual data model is one of the core concepts in multimedia database design. Functionalities of systems are determined according to the used conceptual data model. Existing data models are not qualified to support some required functionalities of multimedia materials; such as storing, indexing, retrieving and accessing these types of data. Therefore, a conceptual data model for multimedia objects is presented in this section.

As stated in previous chapters, multimedia objects contain huge amount of information that exist in complex and compound structures. Being in various forms and the diversity of semantic contents, make it difficult to model multimedia materials. Therefore, in this study, multimedia materials are categorized as: *visual* materials, *audial* materials and *textual* materials. Since their structures differ from each other, they should be analyzed separately.

Besides being huge and in compound form, distinctive features of multimedia data are; having various types of data in different formats, being lack of a standard structure, carrying spatial and temporal characteristics, and insufficiency of textual descriptions. All these features and the diversity of potentially available information make it difficult to express precise queries, because, the uncertainty is an inevitable and natural property of multimedia objects.

The most popular query types in conventional information retrieval systems are existence queries, that is, query definition is exact and all the results acquired exactly satisfy the query conditions. On the other hand, usually, we can only find the best match to the query predicates in multimedia information retrieval side; in most situations, even an exact match does not exist. In addition, due to the foggy ideas they have about the searched entities, it is not always easy for users to express their requests precisely. So, when developing a multimedia information system, the uncertainty must be considered in all steps of development.

In conceptual data models, entities and relations are completely defined. Therefore, in such a model concerning multimedia data, semantic entities; objects, events and the relations of entities should clearly be defined. Besides the hierarchical structure of these entities, since some multimedia data types contain time-specific components, temporal segmentation of such data gain importance in conceptual data modeling.

In many studies [9, 13, 18, 29], authors utilize a well-known temporal segmentation for visual materials having time information; *shot-scene-sequence* approach [Figure 4-1]. The smallest temporal segments, shots, are defined as the minimal group of adjacent frames, stating a continuous action and having images from the same area, therefore, contain some common low-level features. Related shots, happening in the same time and place constitutes scenes, and semantically closer neighboring scenes, which make up a continuous story when assembled, built sequences. Video objects are composed of these sequences.

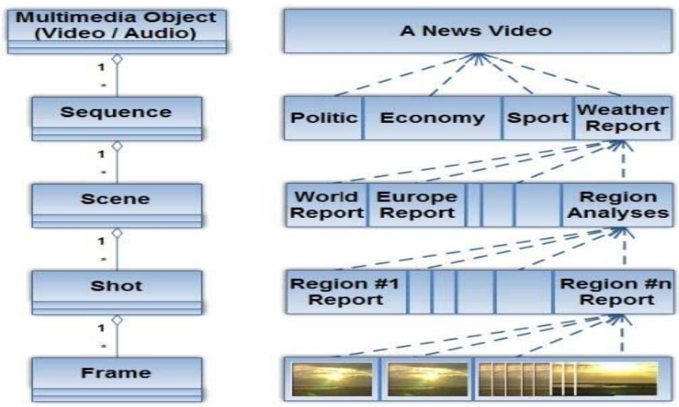
Temporal segmentation is important also for event definitions in videos. Although objects can be extracted from a single frame or image, since events contain temporal information, a group of continuous frames, at least shots are required for an event definition in visual materials. Events are bounded with a single or a few contiguous shots in many studies.

For image objects among visual types, a spatial segmentation may be applied for partitioning images into smaller granularities in space dimension. If a meaningful spatial segmentation is applied using some low-level features, as in study [2], some regions may directly be mapped into objects easily.

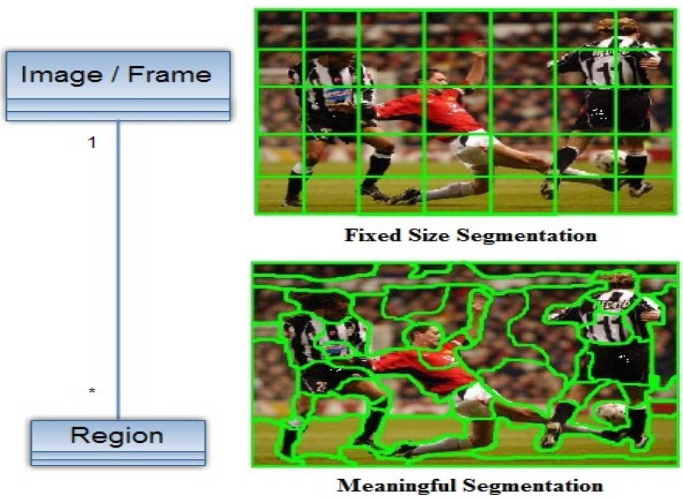
Only temporal segmentation is considered for audial objects, since they lack of visual components. Object extraction as well as event detection from audial objects requires a temporal segmentation, since minimum meaningful parts of audio objects have time-components. In many studies [86-88], a conceptual segmentation and classification is applied for partitioning audial materials. Audio objects are segmented into *audio segments* and *audio samples.* Also, these segments are usually classified into *silence*, *environmental sounds*, *harmonic sounds*, *speech* and their combinations. Since audio samples indicate a simple value for a specific time, audio segments are used for object and event boundaries.

A structural partitioning approach for textual materials is *paragraph-sentence-word* segmentation. Since time and spatial information is bound into textual materials, this type of division can be considered as temporal or spatial segmentation. Objects and events are extracted from textual materials using *words* or *word-groups*.

All these partitioning approaches, temporal and spatial segmentation, are useful for easy modeling and associating semantic contents with the physical portions of multimedia data.

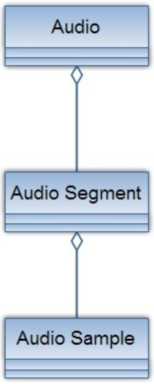


(a) Temporal Segmentation

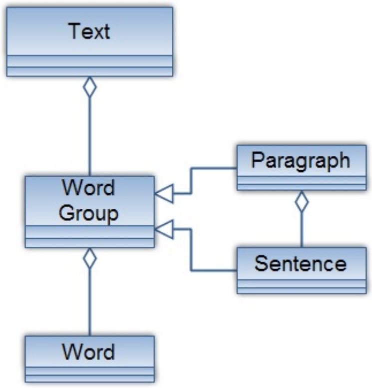


(b) Spatial Segmentation

Figure 4-1 Segmentation of Visual Multimedia Objects



(a) Audial Object Segmentation



(b) Textual Object Segmentation

Figure 4-2 Segmentation of Audial and Textual Multimedia Objects

There are semantic relations between multimedia entities that should be considered when modeling semantic information in multimedia materials. These relations and interactions, in which objects have different semantic roles, are called as events. An object may perform an active or passive role in events. In a scenario, containing several events, an object, being an actor in some events, can share the victim role, or be the affected one in some other events. In studies [10, 13, 29], objects, playing active role are named as *actors* of events. In addition to type, time and place information, the roles objects playing distinguish events from each other.

Beside object-event relationships, there may be object-object and event-event relations between these semantic entities. Usually, object-object relationships include spatial relations, and event-event relationships include temporal interactions. Event–sub-event relations, which represents a temporal hierarchy between events or a whole-part relation of events, can be handled with event-event relation definitions.

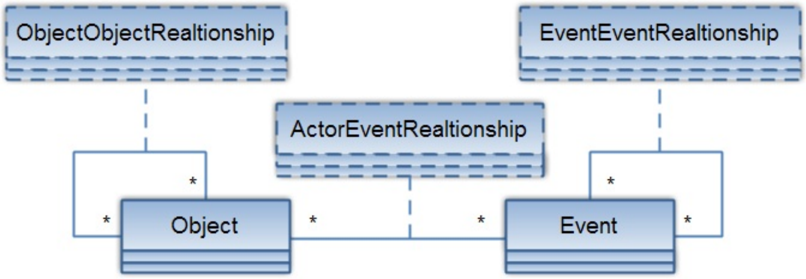


Figure 4-3 Object-Event Relations and Association Classes

Based on earlier studies [13, 29], semantic content relations with temporal segmentation hierarchy of multimedia entities are modeled in an object-oriented approach and a generic fuzzy conceptual model for multimedia data covering visual, audial and textual object types is given in Figure 4-4, using the extended UML [11, 13, 28]. This model does not restricted to any domain and domain specific entities can easily be derived by extending object and event classes. In the following empirical study chapter, *news* domain specific objects and events are created and briefly explained.

4.2 The IFooMMD Architecture

Today's almost all operating systems support exchanging information over Internet. Therefore, web based applications gain importance since they provide a standard means of interoperating between different software applications, running on different platforms.

When the variety of production and consumption environments of multimedia data is considered, the need for an adaptive infrastructure for manipulating these distributed data becomes clearer. Moreover, since researchers, usually focusing on a small sub-area of multimedia related issues, have their work operable on only some specific platforms due to some platform dependent external tools they used, the necessity of platform independent infrastructure for handling multimedia data is revealed. Therefore, a web based thin client multimedia database architecture is designed and proposed in the scope of this study.

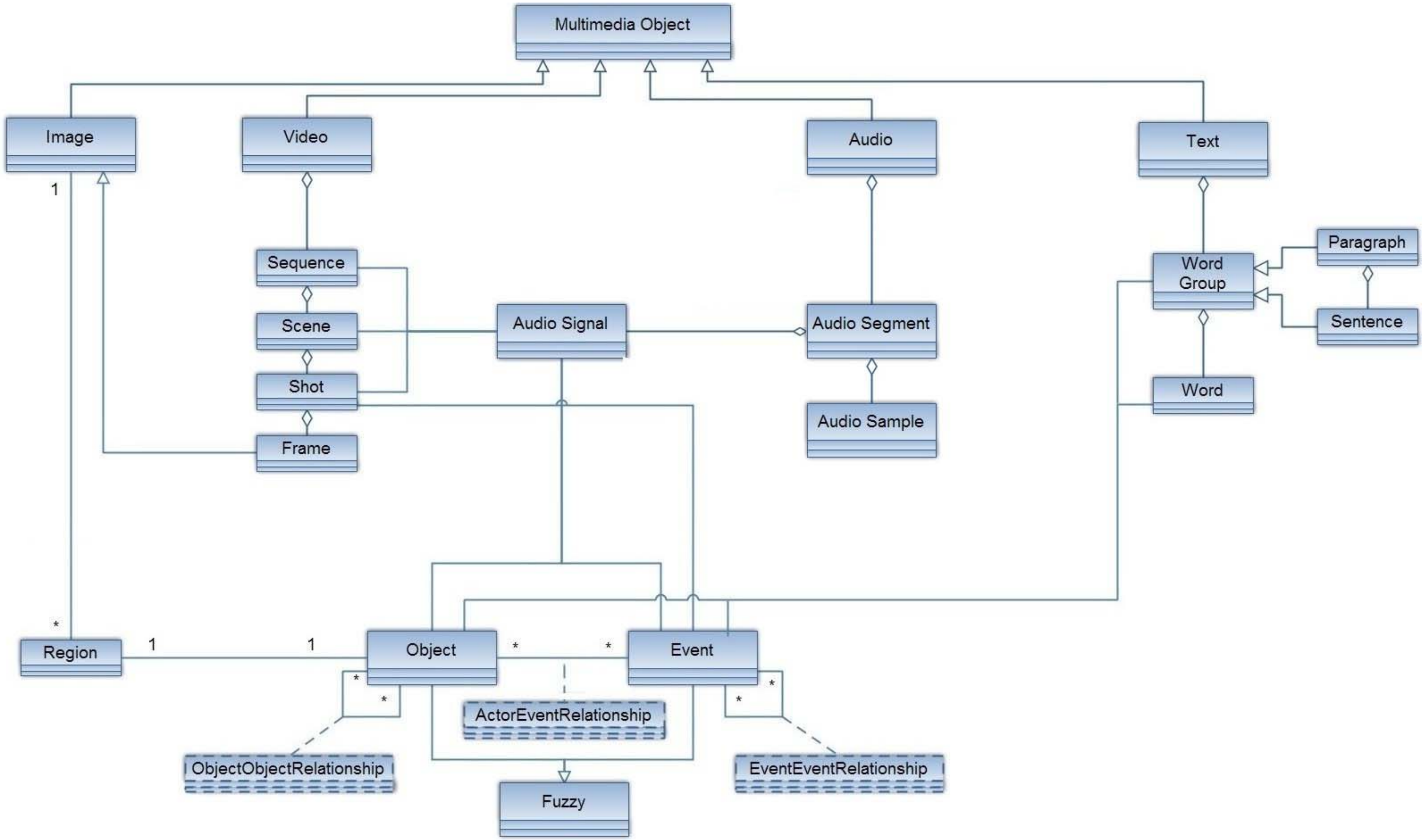


Figure 4-4 The Generic Conceptual Multimedia Data Model

The overall view of proposed system, requirements and detailed information about whose sub-modules are explained in subsequent sections, is given in Figure 4-5. A component-oriented approach is followed and a standart XML based communication method is recommended for interaction between the components of the architecture, as well as between client and server of the system.

In the IFooMMD architecture, a semantic information extractor, a high-dimensional index structure and an intelligent database components are tightly coupled through a coordinator structure that provides interoperability. Information is provided from semantic content extractor, stored in intelligent database component with uncertain properties, and effiently retrieved using the index structure specialized for multimedia data. The coordinater is responsible for providing the interoperability between these components to achive multimedia data management. Besides, since the interaction with the client side is established via corrdinator, it can be assumed as the interface of the system, to the outer environment.

In the client side of the architecture, a light-weight, thin client application structure is suggested, such as browsers or mobile applications. Since results of queries may contain audial and visual objects, it should be able to process audio-visual materials as well as textual results. Due to difficulties in constructing multimedia data queries, the client should have user-firiendly interfaces for building visual queries.

In following sections, the requirements and detailed explanations of each submodule, including coordinator structure is presented.

4.2.1 Semantic Information Extractor

In the conceptual data model proposed in previous sections, semantic entities that can be found in the multimedia data and their relations are provided. The semantic entities in the conceptual model; objects and events are abstract classes, and should be inherited by the domain specific entities while realizing the model. Thus, semantic contents in multimedia materials can be expressed with real world domain specific entities. This approach revealed the need for finding and extracting semantic entities from multimedia data, which is another challenging research area.

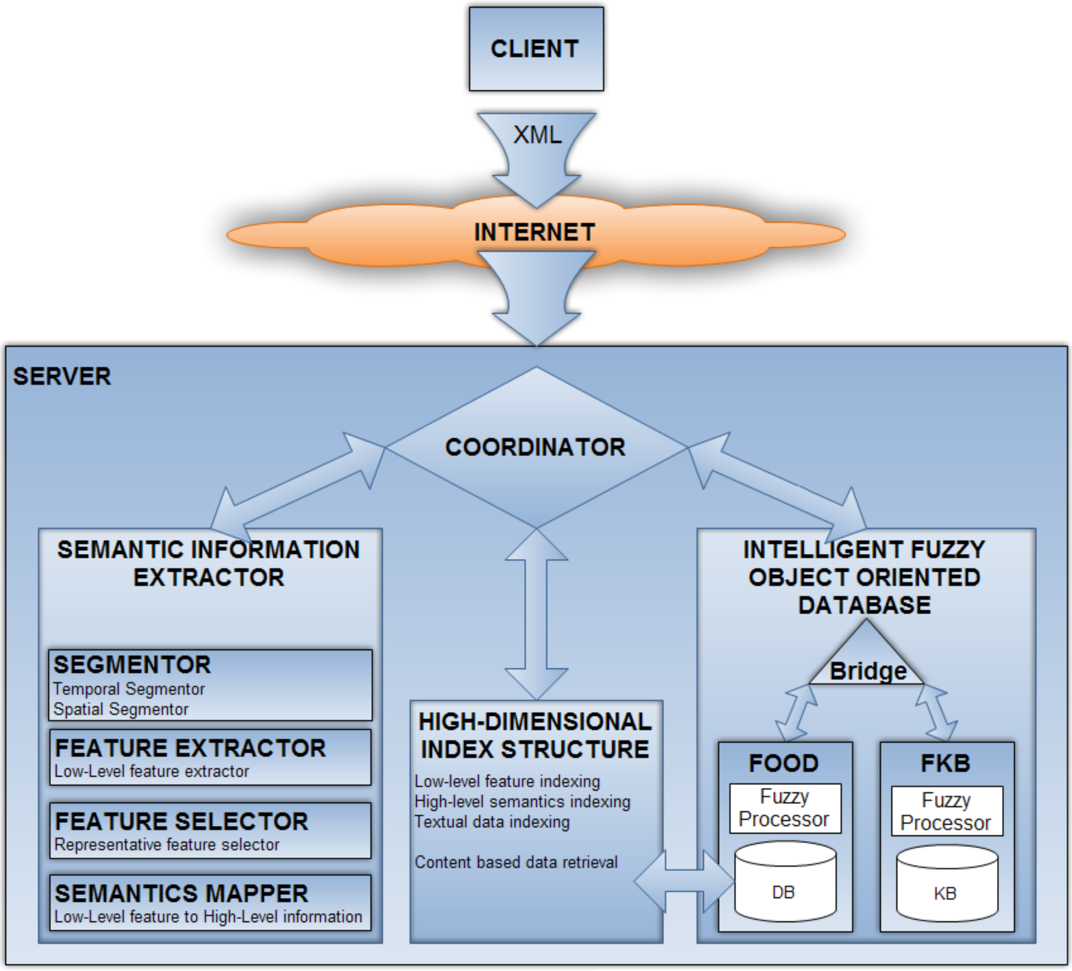


Figure 4-5 IFooMMD Architecture

Although there are studies offering information extraction methods from whole gobs of data, whatever the data type is, raw multimedia data should be segmented into small granularities to make them available for easy processing. Considering their complex and hybrid structure, it can be said that a partitioning approach in time and space dimension is inevitable for multimedia objects.

Temporal segmentation is required for data types containing time component, that is to say, video and audio data. Mostly used temporal segmentation approach, *shot-scene-sequence*, is offered in the proposed model. In literature, there exist various studies, offering algorithms and tools for dividing data in means of smaller time units. Among these studies, the most popular one, IBM MPEG-7 Annotation Tool

[72], is used in many studies for temporal segmentation of audio-visual materials, since it supports MPEG-7 specifications.

Although the smallest temporal segment is *shot* for audio data, *frame* is the smallest temporal granularity for video objects, which can also be represented with another type of multimedia data; *image*. In addition to one step further segmentation in time dimension, a spatial segmentation can be applied for images to small down the granularity in space dimension. Also for textual data types, the process of dividing text into meaningful units, such as words, word-groups, sentences or paragraphs can be considered as spatial segmentation, since time information usually does not exist for this type of materials.

There exist many spatial segmentation algorithms and tools, which segment images using a fixed width segmentation approach or partition images according to their color, shape and texture distribution. Since shape or color distributions are mostly variable in object types, algorithms usually over-segments images and produce some wrong or irrelevant piece of segments, resulting in the need for some further issues to eliminate or combine these parts [Figure 4-6]. Similarly, for textual materials, a group of words are needed to be combined to extract objects or events from them.

The next step in detecting semantic entities should be extraction of low-level features from images or segmented image chunks. MPEG-7 technology offers many descriptors for audio and video data. Although, feature extraction is out the scope of MPEG–7 specifications, due to some interoperability issues, it specifies extraction process at some degree. Any low-level feature extracted according to these specifications can be used in object detection. The MPEG Reference Software [71] (eXperimental Model (XM)), one of the featured low-level feature extraction tools, is utilized in many studies [2, 46, 69]. Besides extracting low-level features of multimedia materials, it is capable of calculating distance values for low-level features, which is usually used in similarity search approaches.

(a) **(b)** **(c)**

Figure 4-6 An Over-segmentation Sample, (a) Sample Image, (b) Segment Boundaries, (c) A Candidate Object, Obtained by Combining a Group of Over-segmented Parts

Low-level features should be mapped to high-level information in order to find semantic contents. Since this mapping cannot be defined as one-to-one relation and high-level entities usually have subjective and uncertain information where as low-level features are defined by crisp values, the semantic gap between them is another problem that should be solved in semantic information extraction step. Since no direct mapping is available, an intelligent and learning approach should be used.

Neural networks, learning-based algorithms, rule-based solutions and classification methods are proposed for bridging semantic gap between low-level features and high-level semantics. Independent from the computation type, usually, the assigned weights of low-level features are used in calculations. For each semantic entity, assigning some representative values to low-level features is required. A general solution, assigning fix-weights for all entities may tend to produce wrong mappings, since the representation degree of low-level features for different objects may differs. For instance, whereas, only the color information may be sufficient for extracting *sea* object, shape information is more important while finding a *ball*.

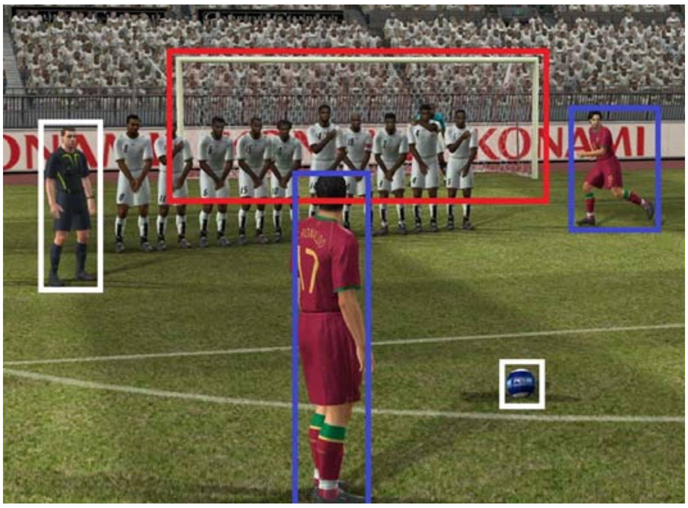
Both shape and color features gain importance when a *glowing sun* is in consideration.

Temporal and spatio-temporal relations of objects, samples of which are given in Figure 4-7, constitute another type of semantic content; *events*, which should also be detected in semantic information extraction step. Since this type of contents have time components, granularity in any level of temporal segmentation can be used in detection scheme. Usually, the boundaries of events are defined using the borders of shots or a group of shots. In certain cases, even scenes are used in defining the duration of an event.

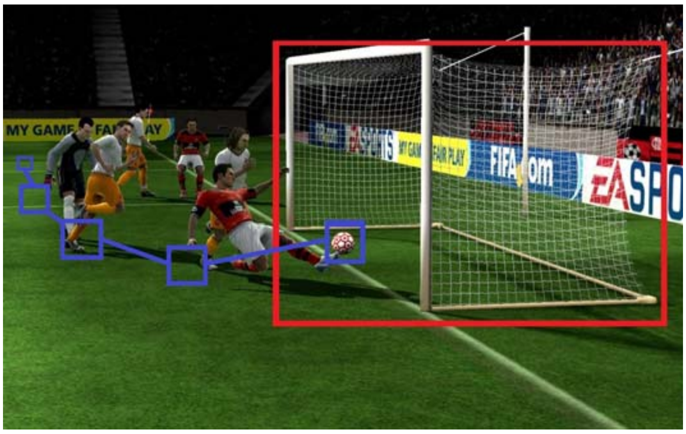
Being not valid for audial objects, event extraction from visual material is mostly based on pre-existing objects. The spatial relations of objects and the positional changes (trajectory information) in time dimension, namely temporal characteristics, provide information used in defining events. With the detection of these behaviors, rule based approaches, classification methods, ontology or any deduction or inference methodologies can be used in event detection. Therefore, event extraction is a complementary operation in semantic information extraction process, which should be performed after object annotation step.

Obtaining semantic entities from audio data is different from acquiring these contents from visual materials, in that, object entities are inferred using events in audial materials, whereas, usually events are detected according to movements or positional relations of objects in video data. Therefore, events should be annotated before deducting objects from audial materials.

After temporal segmentation of audio data into *audioshots*, usually some filters are applied for removing or reducing noise signals. Particularly, smoothing filters truncate momentarily peaks or environmental background sounds, which may cause faulty decisions.



(a) Spatial distribution of objects in a free-kick event



(b) A ball trajectory in a goal event

Figure 4-7 Sample Spatial and Temporal Relations for Event Detection

Low-level features of clean audio signal should be extracted to be processed in event classifiers. The MPEG Reference Software (XM) may be utilized for obtaining these features also from audial objects as from visual data. For obtaining events from audio signals, the obtained low-level features are classified using feature characteristics of events. Several rule based or model-based approaches can be applied in classification steps; Gaussian Mixture Model (GMM), Hidden Markov Model (HMM), Support Vector Machines (SVM) as well as Neural Networks (NN) or fixed thresholds.

Some specific information as well as crisp object instances can be obtained from textual materials. Using a structural segmentation, textual data is partitioned into *words* and *word-groups*. Objects can be detected by examining these textual segments using special dictionaries. Also, rule based methods can be applied for obtaining some detailed information in this step; such as time, location information and particular properties of objects. To detect events and concepts, usually, frequencies of *words* and *word-groups* are calculated and some statistical or rule based methods are utilized.

4.2.2 Database Architecture

4.2.2.1 Multimedia Data and Databases

Multimedia data and semantic entities inside are different from the primitive data types, like textual or numerical types, in the way that they require a large amount of memory and disk storage. In traditional databases, this kind of information, in other words, compound objects having high-dimensional information are mostly stored in binary large objects (BLOBs) as a whole (Figure 4-8) and the content of these objects are usually left out of the database concern. In certain cases, some cataloging information for this kind of objects, such as authors, subjects, and rarely, textual definitions of contents are defined in other fields to allow querying this data, in a manner restricted to those fields.

BLOB fields in traditional databases contain large amount of data, therefore, long and nested transactions in terms of access and retrieval time is inevitable. Although some commercial systems enable indexing and accessing the content of these data, using some linear scanning methods, this field is not specialized for such operations. Even when the contents inside these fields are extracted, with relational approach, a time consuming object-relational mapping is required at storing or retrieving step, due to some normalization issues. In order to cope with all the above considerations, using an object-oriented database system is the best alternative to traditional relational approach for managing multimedia data.

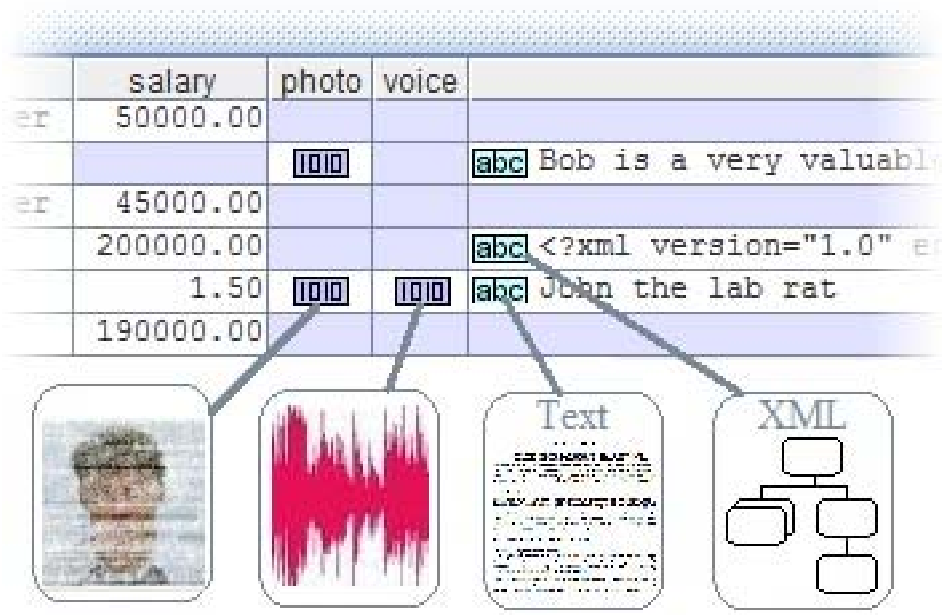


Figure 4-8 A BLOB Field Representation in Traditional Databases

Since multimedia data contain great amount of complex information that exist in compound and hybrid form, automated semantic information extraction systems are mostly not able to extract all the entities and the relations between them, posing a risk for loss of some possible important information. But, manual methods for extracting information from multimedia data are limited and subjective to user. Therefore, obtaining information from pre-existing data, that is to say, using deduction and inference methods for information extraction is indispensable for multimedia databases.

Intelligent systems require an interoperable use of database and knowledge base systems [30]. Rule-based approaches or knowledge-base mechanisms can be employed to enrich and improve extracted information, and add a level of intelligence to multimedia database systems. In addition to obtain additional semantic information, these inference methods may be utilized to fulfill the semantic gap between low-level features and high-level concepts. The inference system, to be used in multimedia databases, should be able to process fuzzy information besides accurate ones and should also be able to infer both certain and uncertain results whenever possible.

IFOOD [30], an intelligent information management system, offers an efficient interaction between database and knowledge base technologies. By integrating a fuzzy object-oriented database and a fuzzy knowledge base, IFOOD is able to handle fuzzy object management with the assist of knowledge base inference capabilities. Therefore, in the proposed infrastructure, an IFOOD based database architecture is suggested.

4.2.2.2 Intelligent Fuzzy Object-Oriented Database

IFOOD [30] is an intelligent information management system architecture, based on coupling a fuzzy object-oriented database (FOOD) with a fuzzy knowledge base (FKB).

In the IFOOD system, object and storage management is handled by FOOD module. The logical model, used in this module, is firstly proposed in [89] and able to represent imprecise information that might occur in complex objects. The uncertainty is handled at tree levels: *attribute level*, *object/class* and *class/superclass levels*. Domain objects, relations and membership functions are stored in this module.

Domain knowledge is represented by rules and a rule engine checks each rule for a set of facts in order to decide whether to fire the rule or not. In the IFOOD system, knowledge management is administrated by FKB, which has fuzzy inference capabilities to handle fuzzy rules.

Both FOOD and FKB modules use fuzzy processors for uncertainty. All the fuzzy membership functions, object/class and class/superclass membership degrees are calculated using fuzzy processors.

A bridge is utilized for managing communication and interaction between the FOOD system and FKB. In addition to provide interoperability, it is used as an abstraction layer and entry point for user requests. Bridge is the organizer while processing user requests, in evaluation time.

As this architecture enables users to define entities and relations, create fuzzy rules, and query the system with uncertain predicates, it is a candidate database architecture for multimedia frameworks. The proposed architecture of IFOOD is given in Figure 4-9.

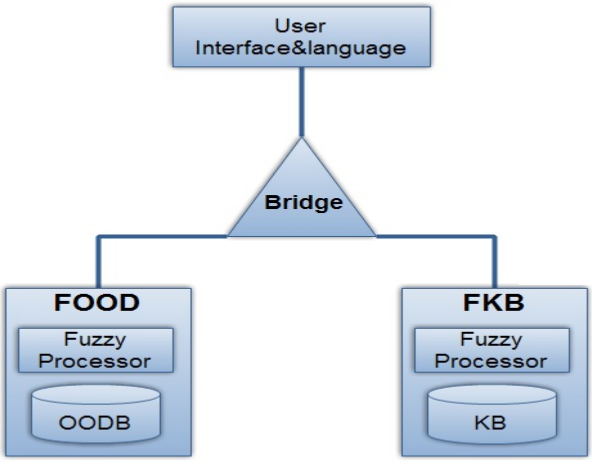


Figure 4-9 IFOOD Architecture

4.2.3 High-Dimensional Index Structure

In previous sections, a conceptual data model for multimedia materials is proposed and methodologies for detecting and extracting low-level features are presented. Also, approaches for analyzing low-level features and methods for mapping low-level information into high level semantics are briefly explained. In addition, storage requirements, the importance of employing knowledge-bases and the necessity of using an object-oriented database structure with fuzzy additions are presented. In other words, objects and events are detected and extracted from raw multimedia data, and stored so far.

The next important advance in developing a database for multimedia material is finding an efficient way of indexing, querying, and accessing to them. Since the choice of index is correlated with the contents and query types frequently executed, besides high-dimensional characteristics, multimedia data types and the properties of semantic information that can be obtained from these materials should be in consideration.

In multimedia databases, since the information is not atomic and has an unformatted, hybrid structure, it is mostly difficult for users to express their queries in formal query forms. Blurred ideas about the searched content and difficulties in expressing fuzzy linguistic terms in structured query languages brought the query-by-example type of content based queries in consideration. Since exact match is not possible in most situations, closeness or similarity searches gain importance in such queries.

Content based retrieval, particularly query-by-example type search, is a difficult task to be accomplished in multimedia databases. Although the semantic content of stored multimedia data has already been extracted before, there is no clue about the information inside the given example image. Also, since some level of information is truncated as an inevitable result of mapping low level features into high level concepts, it is usually impractical to use high-level information in similarity searches. Therefore, a way of comparing low-level features of the searched example image with the ones stored in database and deciding the similarity degrees for each object is needed. Since a lossless way of remapping high-level concepts down to low-level features is not exists, all levels of information, obtained in the extraction step of semantic entities should be preserved for latter use in similarity searches.

Multi-dimensionality is the natural feature of multimedia data and semantic entities extracted from them. Indeed, high-dimensionality should be mentioned for multimedia materials since they have hundreds of features and properties of features that can be used to express them. Even just for low-level features, there are many descriptors defined in MPEG-7 specifications, most of which have more than a half-hundred dimension. Unfortunately, when the dimension of the data increases, performance of multi-dimensional index structures decreases, usually called the *curse of dimensionality*.

Where high-dimensionality is in consideration, most index structures become extremely inefficient because the number of nodes increases exponentially and a simple linear scan of all objects is typically faster than using an index structure. Also, most high-dimensional index algorithms face the problem of indexing large volumes of empty space [55]. Therefore, storing and efficiently retrieving multimedia data requires an index structure, capable of handling such huge dimensions of multimedia objects, as well as their low-level features.

To overcome the drawback mentioned above, an index structure, specialized for multimedia data, even supporting MPEG-7 specifications and capable of handling low-level features is required. Since high-dimensional indexing methods are based on the principle of hierarchical clustering of the data or search space, it should be able to build a cluster hierarchy on the top of data nodes. Due to its bad effects on processing queries, the index structure should minimize overlaps between directory and leaf nodes, since they increase the number of paths to be traversed for accessing data nodes in tree based approaches. To be used for multimedia materials, the index structure must be appropriate for similarity searches, which is the most meaningful query type used for audio-visual objects.

4.2.4 Coordinator Component

The coordinator structure is responsible for setting up interoperability between components of the architecture. Flow of the communication is arranged with this component. Also, as an interface to outer environment, the interaction between client and server components is established via coordinator component.

When the variety of researches on different platforms about multimedia is considered, to support different types of components developed in different environments and platforms, the coordinator should be flexible and extendible, what makes the IFooMMD architecture more scalable. The realized structure of the coordinator should be easily manageable, since minor modifications are usually required for integration of components.

CHAPTER 5

IMPLEMENTATION OF THE PROPOSED SYSTEM

In previous chapter, a conceptual data model and a database architecture for managing multimedia data are proposed. The empirical study on the whole framework and implementation details of integration are presented in this chapter.

In the scope of this thesis, a server framework which tightly couples some fuzzy multimedia database modules and a prototype client application for test purposes are implemented. Server framework and the prototype are developed using Java programming language and *news* domain is selected for the proof of concept. Also, because of the reason they comprised lots of the multimedia object types (such as image, audio, text, temporal and spatial events), in this study, most of the experiments are executed over video objects.

In the first section of this chapter, details, related with overall implementation are explained. The next section gives brief information about the integration process. For ease of understandability of the whole architecture, following section is reserved for clarification of workflows for possible multimedia database operations in our architecture, such as object creation, insertion, querying and retrieval. The development steps of event ontology and operations of web services, implemented for interoperability with other systems are explained in last section.

5.1 Overall Implementation Details

There are three main components in IFooMMD architecture; Core Database Component, Semantic Information Extractor Component and an Index Component, information about each of which is given in previous chapter. Since each component, and even some inner sub-components of IFooMMD architecture is a research area itself, in the implementation phase, some pre-existing researches are integrated over the skeletons of coordinator structure of the architecture.

One of the coupled study, [13], proposes a conceptual data model for video materials, as well as a video database framework, based on the IFOOD architecture [30]. In addition, an index structure specialized for handling low-level features of multimedia data is presented in [69]. Using a native XML database to store low-level features of video shots, it utilizes an approximation methodology and implements BitMatrix index structure. A Genetic Algorithm (GA) based object extraction methodology [2] and a domain independent ontology-based event detection studies are the other modules, coupled in realization stage of the proposed architecture. These studies are adapted to the IFooMMD architecture and gathered into an interoperable form using the coordinator servlet interfaces.

In the implementation steps, all the development is carried out in a web based manner on Java environment using the interface layers of each module, implemented as java servlets for this study, in a component-oriented approach. In each step, an extendible, platform independent and rather simple system is aimed.

In the IFooMMD architecture, MPEG-7 technology and usage of low-level features described in MPEG-7 specifications are recommended. Although any of low-level features can be adapted to the system, only four of MPEG-7 features are utilized in this implementation; Color Layout (CL), Dominant Color (DC), Edge Histogram (EH) and Region Shape (RS), detailed information about which is given in background knowledge chapter. These features are extracted by Semantic Information Extractor Component using MPEG-7 reference software (eXperimentation Model, XM), and both object annotator and index modules benefit from resulting information.

During the realization of the system, in addition to the server framework, a prototype client application is implemented. Query interfaces of client application are formed at runtime according to an XML document, defining domain and semantic entity information. Therefore, domain modifications or changes in server framework require very few modifications for prototype client application.

Since the proposed IFooMMD architecture recommends using XML based communication methods, interaction between server and client instances is established using both XML and HTTP messaging standards. Therefore, any other client application can easily connect to the server framework and use the implemented sample multimedia database framework so long as they satisfy the requirements of messaging protocol. Due to minimized communication messages between client and server framework, the implemented system can also be used in mobile systems. For interoperability with other systems, some web services are also developed and presented.

Although the offered IFooMMD architecture is platform independent, some external tools, utilized in the implementation of object annotation module, force semantic information extractor component of server framework to be in Windows environment. Platform independence may be gained by replacing these Windows-Sticky external tools with platform independent equivalents.

In the following sections, brief information about coupled modules is presented in the order of integration.

5.1.1 Core Database Module

In study [13], a conceptual data model for video data and a video database system, based on the IFOOD architecture, is proposed. Since this study has common properties with the database component of IFooMMD architecture, employ a temporal segmentation approach, use similar semantic entities and handle uncertainty in all data related levels, it is adapted and utilized in the database structure component of proposed IFooMMD study.

In the proposed data model, in order to support uncertain data and represent video specific properties, UML is utilized and extended by introducing some special structures. The extended UML is capable of representing uncertainty at the attribute, object/class and class/subclass levels as defined in FOOD [31] model.

For handling attribute level uncertainty, tree new types are defined; *UT_nu* for null data type, *UT_in* for incomplete data type and *UT_fy* for fuzzy data type. For uncertain attributes;

<attribute_name> : <type> <range> <relevance>

syntax is used.

For object/class level uncertainty, which refers to partial memberships of objects to their class(es), a *U* notation is used for fuzzy objects and an *objectMShip* variable is introduced for representing the inclusion values of objects to their class(es). Uncertainty at class/subclass level, which defines the membership degrees of classes to their superclass(es), is handled by introducing a fuzzy relationship constructor using an *F* notation, and a *classSClassMShip* variable. Similarly, fuzzy aggregation and composition relationships are represented with an *F* capital on upper side of their symbols.

Besides handling uncertainty, similar semantic entities and relationships are modeled in conceptual model. Also, a partitioning approach is followed and IBM Annotation Tool [72] is utilized for temporal segmentation, used for stating boundaries of events. Spatial segmentation and entity extraction are declared as out of the scope of research and in prototype application of this study, manual techniques are employed.

A video database system, based on the IFOOD architecture is also presented in study [13], the conceptual model of which is given in Figure 5-1. Using IFOOD architecture, modeling complex and rich semantic content of video data is provided.

DB4O [81] is used as object-oriented database and a rule engine, JESS [82] is integrated and some fuzzy inference rules are developed and included in the fuzzy processor of FKB architecture.

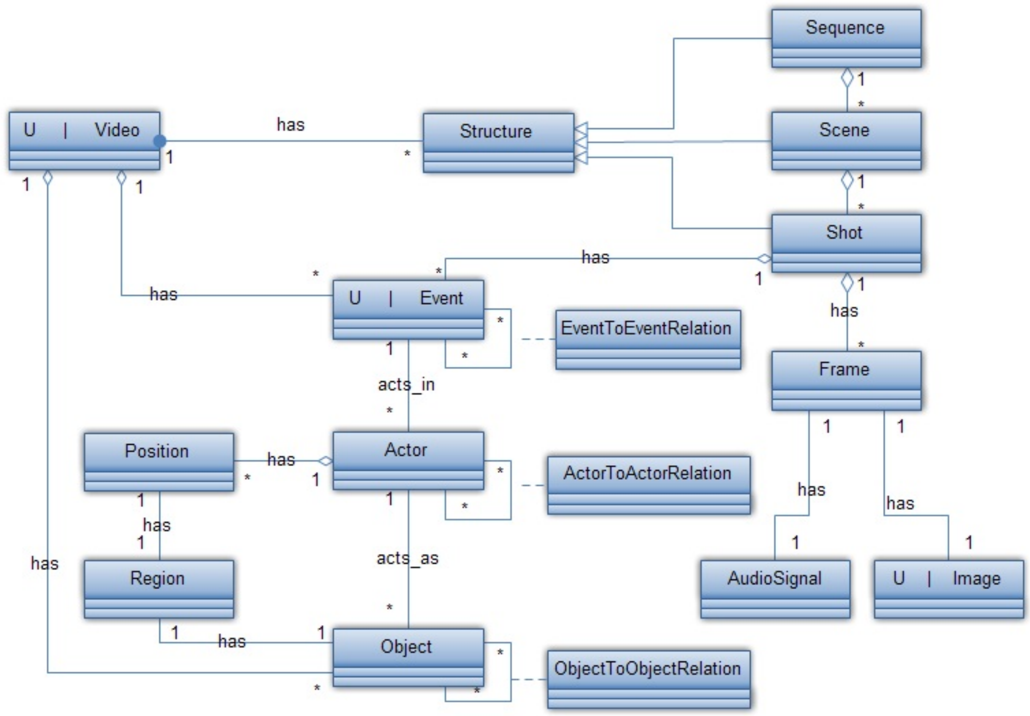


Figure 5-1 Conceptual Model of [13]

5.1.2 Object Annotator Module

A Genetic Algorithm (GA) based object extraction methodology is presented in [2]. In the study, the object extraction process is handled as a categorization problem and the proposed GA based classifier is used for classification of candidate objects in image/frame segments. Since it has full support for MPEG-7 specifications and uses some MPEG-7 descriptors, classifies objects using frame/image segments and defines objects with the Best Representative and Discriminative Feature (BRDF) model, this study is utilized in order to meet the object extraction requirements of the Semantic Information Extractor component in IFooMMD system.

The architecture of object annotator has four components; key frame extraction module, segmentation module, feature extraction module, and classification module [Figure 5-2].

IBM MPEG Annotation Tool [72] is utilized as the key frame extraction component and important frames of videos are extracted automatically. Prior to feature extraction and classification steps, N-cut [90] segmentation library is employed in segmentation module and extracted key frames of video are partitioned spatially into meaningful granularities. Low-level features of these segments are extracted using MPEG Reference Software (eXperimental Model, XM) [71] in feature extraction module and a developed GA-based object classifier is employed in classification module for making decisions about possible objects. The classification is performed iteratively, and in each iteration, some neighboring segments are combined and the new segment is added to the classification queue.

In training phase, to find the best representative object instances for categories, instead of using the average feature values of training samples, a random set of feature values is stored, then for each training sample, the relevance of sample object to object categories is calculated and a genetic algorithm is used to find the best ranked representative set. This representative set is used in calculating similarity distances of query objects to the object categories. The system supports uncertainty and gives fuzzy decisions on the objects, since it is able to make multiple-categorization with some similarity degrees.

5.1.3 Event/Concept Annotator Module

In [46], a general purpose domain independent ontology based VIdeo Semantic COntent Model (VISCOM) is introduced. In this model, object definitions, spatial and temporal relations in event and concept definitions are defined. Various relation types are declared to describe fuzzy spatio-temporal relations between ontology classes in order to construct domain ontologies. Besides, domain ontologies are enriched with rule definitions to lower spatial relation computations and to be able to define complex events more efficiently.

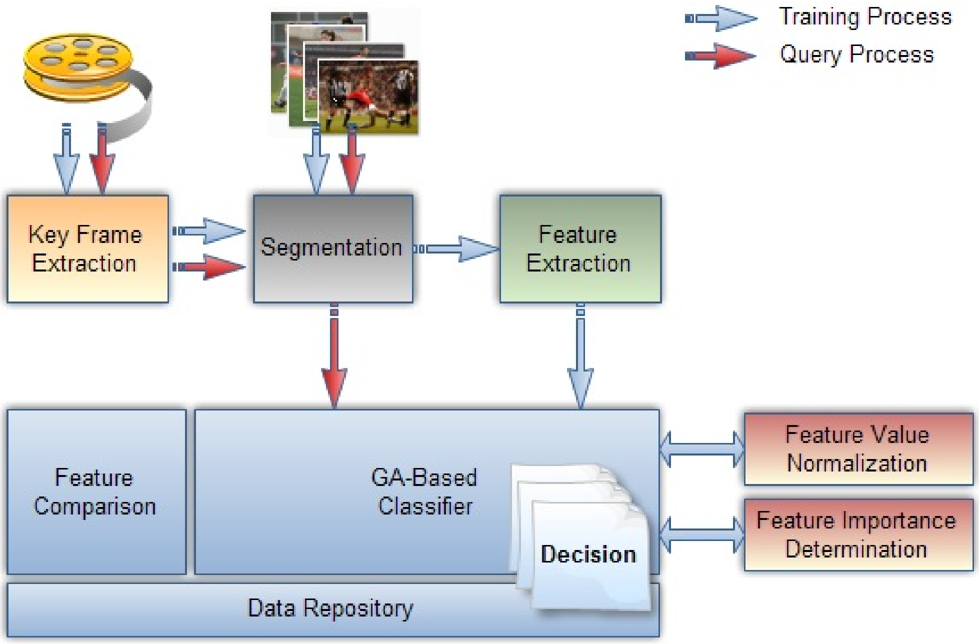


Figure 5-2 Object Annotator Components

After determining object, event and concept individuals, spatial relations between objects and temporal relations between events are decided and individual classes for each relation are defined. Finally, similarity and role definitions were included and creating domain ontology is completed by adding a number of domain specific rules. The classes and relations of VISCOM are given in Figure 5-3.

If used domain ontology allows defining inclusion values for ontology individuals (objects and all other spatial and temporal individuals) for representing membership degrees to events, the system may support uncertainty by inferring partial memberships to defined events.

5.1.4 High-dimensional Index module

Study [69] proposes an index approach for efficient retrieval of videos, based on low-level features of objects. Using a native XML database to store low-level features, it utilizes BitMatrix index mechanism. An Ordered Weighted Averaging (OWA) operator in BitMatrix aggregates features to find final similarity between any two objects. Since it supports MPEG-7 Descriptors and capable of handling such huge dimensions of low-level features, the study is utilized as the Index Component of server implementation.

Index module utilizes IBM Annotation Tool [72] for shot detection and key frame extraction, MPEG Reference Software (eXperimental Model, XM) [71] for feature extraction and a native XML database, Berkeley DB [91] for index and object storage.

5.2 Coupling Sub-Modules

In this study, a sample implementation of the IFooMMD architecture is presented by coupling some modules explained in previous section. For integration purpose, all modules are revised, adapted to IFooMMD architecture and converted into external libraries to be used as components of the system. Some interface layers are implemented for each and communication between these modules and the coordinator is established on these layers.

The first component, integrated into the system is the database module. In this module, objects are manually created and inserted into database. It supports executing textual semantic queries over inserted semantic contents. With the integration of object annotator module, automatic object detection and semi-automatic object creation abilities are added to the system. With the implementation of an event ontology and integration of event annotator module, the requirements of Semantic Information Extractor component of IFooMMD architecture is fully covered and the capabilities of the system is extended to detect and create all the semantic entities in multimedia materials. Having index module integrated into the system, efficient data indexing is achieved and retrieval of semantic contents is accomplished in a human friendlier way using similarity based semantic content queries as well as traditional text based searches.

Below, integration procedures of each module and solutions to encountered problems are briefly explained.

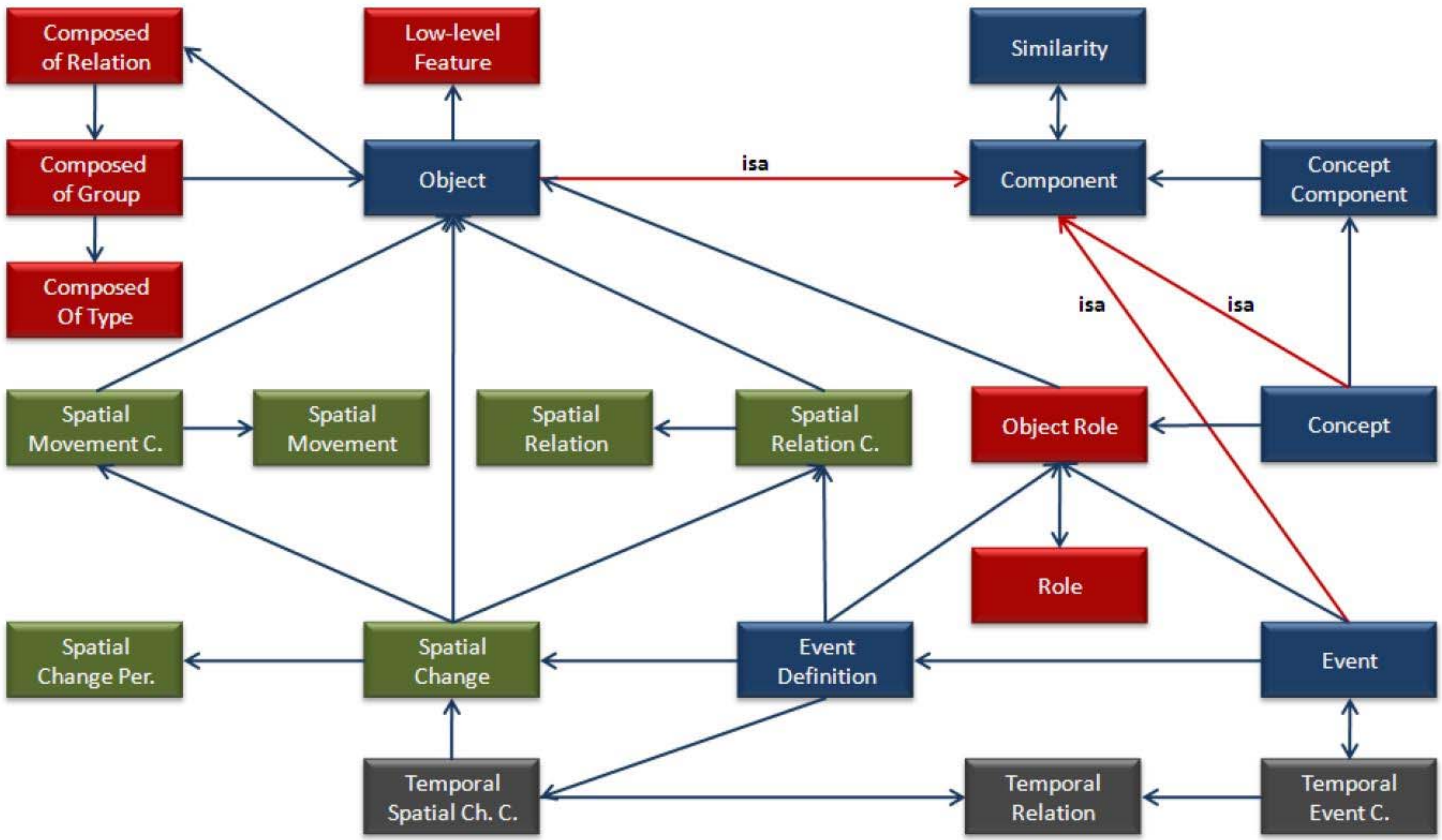


Figure 5-3 VISCOM meta-model

5.2.1 Intelligent Database Component Integration

Core database module study was implemented as an application and manual selection of objects and events from video frames, creation of corresponding real entities and providing attributes and properties for these semantic contents are accomplished using some application interfaces. Since IFooMMD architecture uses thin client technology, this module needs some modifications to work in web environment.

In IFooMMD architecture, as a requirement of thin client technology, all the operational functions are recommended to be executed in server side components and the supplementary functions, which mostly requires user interactions, are suggested to be performed on client side applications. To ensure this requirement, in database module, manual selection and entity creation functions are separated from each other, modified to work as separate libraries and only the creation of real semantic entities is performed in server side components, where as manual selection of object regions and event boundaries, and providing attributes and properties of these semantic contents are left as client side operations.

In this step, *InsertAllNewsDomainObject* and *InsertAllNewsDomainPerson* servlets, which are responsible for creation of entities derived from object and person types, are implemented to be used as an interface of server side object create and insertion operations. Similarly, *InsertAllNewsDomainEvent* servlet is deployed to get information of events from client side, and to store them in the database. Also, the infrastructure of XML based communication channel is constructed and information messages about objects and events are transferred using this channel.

Although most functionalities of the integrated database module are adapted to the IFooMMD architecture, it still lacks an automatic detection schema for semantic entities. The abilities of core database module are enriched by the automatic semantic entity detection and extraction capabilities of Semantic Information Extractor component after integration of object and event annotator modules.

Coupled database module is developed for football domain. The implementation of IFooMMD architecture in the scope of this thesis is aimed to support *news* domain. However, *news* is a wide domain to be used as proof of concept, and so, some restrictions are required. The domain is narrowed into four sub-domains:

- Accident Domain
- Sports Domain
    - Football Domain
    - Tennis Domain
    - Basketball Domain
- Weather Report Domain
- News Report Domain.

The proposed conceptual model is extended to support *news* domain entities using the extended UML [13] and all new class definitions and module modifications are applied to support these new sub-domains. A sample class diagram for a sub-domain, sports/football domain, is provided at [Figure 5-4]. While extending the conceptual model, all previous definitions are preserved and beside inherited properties, new attributes are added at this step. Also the frame-shot-scene-sequence approach of proposed model is protected.

5.2.2 Semantic Information Extractor Component Integration

Integrated database module is capable of handling manual entity detection, creation and insertion functionalities as well as querying these contents using textual methods. With the integration of Semantic Information Extractor component, the manual detection capability of the system is improved, and an automatic semantic entity detection and a semi automatic object and event creation methodology is assembled into the system.

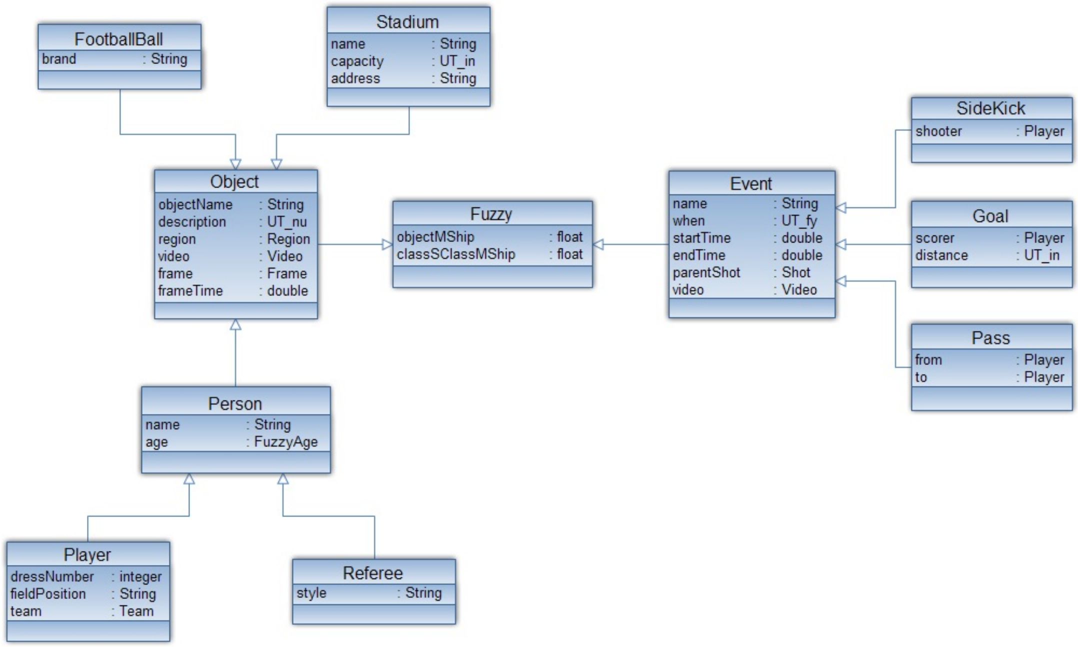


Figure 5-4 UML Class Diagram for Sports / Football Domain Semantic Entities

Object and event detection methods differ from each other, in that, raw frame data is examined for detection of objects, where as pre-existing objects are used for finding event entities in multimedia materials. Therefore, an object annotator and an event detector module are integrated to achieve capabilities of proposed Semantic Information Extractor component. In this association, objects are extracted using segmented portions of images with object annotator module, and then event annotator module, collaborated with a domain ontology, detects events according to the temporal changes in spatial positions and relations of objects. Since event detection depends on the extracted objects, object annotator module is the first one to integrate into the system.

Object annotator module has four components; key frame extraction module, segmentation module, feature extraction module, and classification module. Since even processing of a short video requires a huge amount of time, and some components of the object annotator requires user interaction, this module is modified to work online on web.

The IFooMMD is a web based architecture. Since thin client technology is recommended, a lightweight client is considered in implementation. Because key frame extraction and segmentation functions require user interaction, these functions of object annotator module are moved to the client side, and feature extraction and classification operations are left to be executed in the server.

For executing key frame extraction and manual segmentation on client side, a JMF Player bean is developed for this study. Various multimedia control facilities (load a material from web, play a pre-determined portion of video, select a region or regions on a frame, display and select a single frame/image etc.) are accomplished using this bean.

In object extraction process, the communication between client and server is established using *ObjectAnnotatorServlet*, which is implemented in the coordinator structure and responsible for controlling the flow and format of back-and-forth messages between client and object annotator modules.

MPEG Reference Software (XM) is utilized in object annotator module for extracting low-level features of segmented portions in audio-visual multimedia objects. In classification step, the module uses these features as well as the segmented piece of images. Since index structure needs these features in rebuilding phase, annotator module is adjusted to keep all the information, extracted and used in the classification step, with the annotated objects.

The role of object annotator is extended with the integration of event annotator module and index component. With the completion of the system, it is used at object creation, event detection and query execution operations. In object create and insertion operation, it is used for object classification and extraction of low-level information, which is also consumed by index mechanism. In similarity based query retrieval task, the module is used for pre-processing and extracting future information of given image or frame region. Since event detection is based on previously annotated objects, the object annotator is also responsible for providing object information as well as their spatial positions to event annotator.

Object annotation is not sufficient for multimedia applications. Detecting positional and dimensional changes of objects, following trajectories of them and deducting semantic relations between these objects are necessary. Namely, event detection is also required for covering Semantic Information Extraction requirements of IFooMMD architecture.

A domain independent, ontology based general event detector is integrated and adjusted to support requirements of IFooMMD infrastructure. Since the study proposes only a generic meta-model for extracting event and concepts, a domain ontology is required for detection of domain specific events.

In the scope of this thesis, a sport/football domain event ontology is implemented to be used in conjunction with event annotator. In the event ontology, some common events are defined for detection, such as sidekick, goal and pass events. Detailed information about developed event ontology is given in latter sections in this chapter.

5.2.3 Index Component Integration

The last component, integrated into IFooMMD implementation is the index module. MPEG-7 low-level features, which are extracted by the annotator module and kept with the objects as parts of them, are used in index structure. Although both index and object annotation module supports any low-level feature, only four of MPEG-7 descriptors are used in this implementation; Color Layout (CL), Dominant Color (DC), Edge Histogram (EH) and Region Shape (RS). Therefore, object annotator is revised to extract only these descriptors and index structure is built using them.

In the original study, *Berkeley XMLDB* [91] was employed for storing low-level features in index structure. Since DB4O [81] is utilized as fuzzy object-oriented database in sample IFooMMD implementation, index module is modified to work with DB4O and adapted to read low-level information inside each object.

A clustering approach is utilized as majority of multi-dimensional index models do. Since individual insert operations corrupt clustering scheme, an index rebuild is required after fragmentation of clusters. Also, since low level information of each semantic entity is stored with their objects instead of keeping index structure in database, a build operation is needed at database startups.

ConstructIndex servlet is developed for triggering an index rebuild event and called at database startup automatically. Index rebuild operation accesses all objects in the database, gets desired low-level features, calculates relative distance of objects to each other and constructs index structure according to these information. Although it is the most resource consuming operation on databases, insertion mechanism triggers the index rebuild operation by calling this servlet for each entity insertion, since index module is not designed to be managed individually and since this bulk operation lasts in acceptable times.

The main importance of index module arises when executing similarity queries. Evaluation of similarity search begins at the object annotator module and after extracting low-level features of the given portion of frame and detecting type of the possible objects, *queryByExample* servlet redirects the serialized features to index module. The adapted version of index module is able to attain query string as serialized low-level features. After processing given low level information, and found cluster or clusters that similar object may reside, index module calculates closeness of stored objects to the given features and returns the candidate objects in the order of similarity degree. After fusing results, coming from index module with the ones inferred according to the classification information of object annotator, they are presented to the client.

5.2.4 Implementation of the Coordinator Component

Considering the recommendations of IFooMMD architecture that the coordinator component should be implemented in a flexible and extensible structure, it is developed in a web based manner, using servlet instances. Even the interaction of inter-components is based on web based communications, allowing remote components in other environments to be integrated easily and tightly.

Since the coordinator component is the interface of the architecture to the outer world, some web services are also implemented for enabling other systems benefit from the IFooMMD system. A list of servlet interfaces, implemented in the scope of this study is given in Table 3. Detailed information about web services is given in subsequent sections in this chapter.

Table 3 Coordinator Structure Sub-Components

	Domain		Servlet Name
Insert Servlets	All Domains		InsertAllNewsVideoDomainObject
			InsertAllNewsVideoDomainPerson
			InsertAllNewsVideoDomainEvent
Query Servlets	News Report		SearchNewsReportObject
			SearchNewsReportPerson
			SearchNewsReportEvent
	Accident		SearchAccidentObject
			SearchAccidentPerson
			SearchAccidentEvent
	Weather Report		SearchWeatherReportObject
			SearchWeatherReportPerson
			SearchWeatherReportEvent
	Spor	Football	SearchFootballObject
			SearchFootballPerson
			SearchFootballEvent
		Basketball	SearchBasketballObject
			SearchBasketballPerson
			SearchBasketballEvent
		Tennis	SearchTennisObject
			SearchTennisPerson
			SearchTennisEvent
Helper Function Servlets	All Domains		ConstructorServlet
			ObjectAnnotatorServlet
			ImageFeatureQueryServlet
			InsertVideo

.

5.2.5 A View from Whole Architecture

A view from composed architecture, after integration processes, is given in Figure 5-5. The architecture supports extracting semantic information automatically, enables storing semantic content with their fuzzy attributes and allows query-by-example type queries using an efficient high-dimensional index structure.

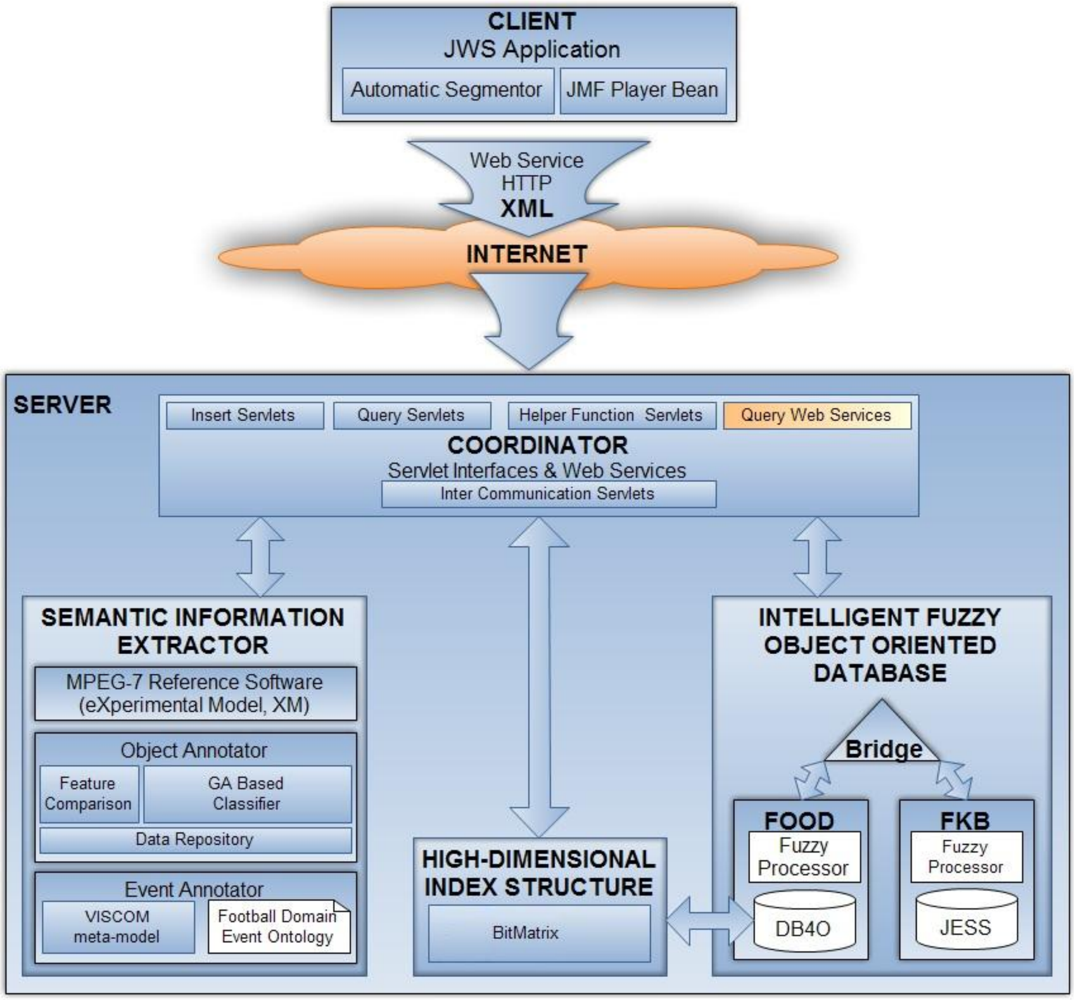


Figure 5-5 A View from Implemented IFooMMD Architecture

5.3 Client Application

The integration process is realized in a component-oriented approach. Testing was necessary during all steps of integration. Since the proposed IFooMMD architecture recommends usage of thin client technology, a prototype client application is necessary for test and development purposes.

Development of client application is started with the integration of first module. With each coupled module, client application is adapted for testing the functionality and the interoperability of new component with other modules. In order to provide platform independency, the main flow of the client application is developed using java language and Java Web Start (JavaWS) technology. Using JavaWS keeps users away from complex installation steps and enables version control procedures, thus, provides a rapid and easy web development environment for the prototype application.

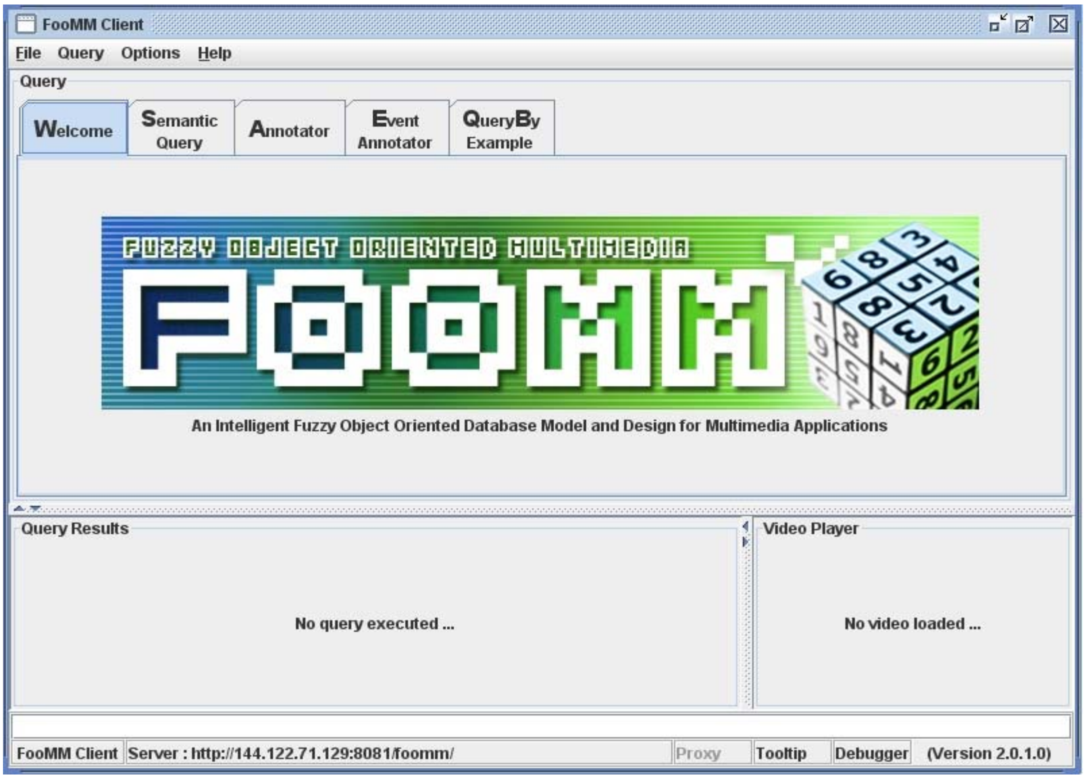


Figure 5-6 A Sample Screenshot from Client Application

In order to provide a single access point and to hide the underlying details, there are user interfaces both for semantic entity creation and query facilities in the prototype client application. The main view of the client application is given in Figure 5-6.

There are three main parts in client application; *query builder*, *result fetcher* and *video player*.

The main functionality of the application is controlled from *query builder*, the part that resides at the upper side of client application. Textual and similarity based queries are built using the interfaces on these part. Also, object and event annotation functions are invoked from relevant interfaces in this part.

Result fetcher is responsible for sending the constructed query to the server and receiving the results coming from coordinator of the architecture. The status of the query, error messages and resulting semantic objects are displayed in this part [Figure 5-7].

JMFPlayer bean, developed for this study for displaying visual materials, is also used in the *video player* part of client application. Details and capabilities of the bean are previously explained with some figures in Chapter 3. Whenever possible, result of the query is displayed in this part.

Since multimedia data usually exists in huge sizes, transferring resulting objects from server to client may cause problems in slow connections. Wherever lesser quality data provides hi-speed transfer rates, but worse object classification performance, higher quality visual data slows-down the system, but ensures high accuracy in object classification. Therefore an ability to select network speed for image quality is added to the client application. Also for debugging and testing purposes, a logger and debugger is implemented for the client application.

(a) A query is being executed

(b) Results of a query

Figure 5-7 Query Results Section of Client Application

Client application provides following four capabilities; executing text based retrieval, extracting and inserting semantic objects, detecting events and performing query-by-example type queries. Detailed information about each capability and the workflows of operations are given in the next section.

5.4 Workflows of Basic Operations

Server framework consists of previously explained sub-modules (database module, annotator modules and index module) and a coordinator structure to provide interaction between them. From logical point of view, important tasks of server framework can be divided into two categories; insertion operations and database queries. To clarify them, detailed information about each operation, is presented in following sub-sections. Workflows of operations are also provided for a well understanding of interactions of modules.

5.4.1 Insertion Operation

What makes database a database is the data inside. Before evaluations of queries, data must be collected and populated. Since IFooMMD is an object-oriented multimedia database architecture, multimedia semantic entities should be detected, created and stored in database before executing queries. As mentioned in previous sections, in the sample implementation of IFooMMD architecture, DB4O is used as storage/database mechanism.

Objects and events can be inserted into the implemented system. Since these are abstract classes, their realized forms, such as ball and player entities, derived from object class, and goal and foul entities, derived from event class, should be created and stored in database.

Due to the difference in detection routines, operations performed in extraction and insertion processes of objects and events are different from each other. Besides, the employed modules and inter communications of these components differ in object and event annotation tasks.

5.4.1.1 Object Annotation and Insertion

In IFooMMD architecture, semantic information extraction processes and database operations are handled in an isolated way. Therefore, extracting objects from multimedia materials and inserting into database are two different tasks to be executed, with the help of distinct modules.

In the implemented IFooMMD system, object annotation is based on finding and classifying visual objects in multimedia materials. In annotation step, detecting boundaries of entities, finding object types and extracting low-level information of objects are performed. Following object annotation, properties and attributes of objects are collected; real object is created and inserted into the database.

For object insertion task, two servlets are developed; *objectAnnotationServlet* and *objectInsertionServlet*, both of which are located in the coordinator component of server.

Object annotation task starts with selection of a region of an object from an image or a key frame of video [Figure 5-8]. The utilized JMFPlayer bean allows users to select desired frame and mark a region for annotation. The selected region coordinates, with video and frame information are sent to *objectAnnotationServlet* for processing.

objectAnnotationServlet works as an adapter for object annotator module; gets parameters from client, converts this information into a structure that object annotator module understands, invokes annotate function of the module to obtain low-level features and decisions about the possible object types in selected regions. Possible object types, as well as an assigned id and possibility degrees of these types are returned to user, according to information gathered from annotator module. In this step, low-level feature information of the annotated object is stored in database and not returned to user due to lessen communication size.

Decisions, returned from *objectAnnotationServlet* gives information about the type of the object. Because of the fuzzy decisions object annotator made, usually more than one possible object is detected and at client side, only the most possible one, the one with the highest possibility percentage, is activated. For correcting possible wrong decisions, prototype client application allows users to change the activated object and select a different type.

Although some attributes of objects, like color and shape information, can automatically be detected, since some properties of objects cannot be automatically extracted, such as power of the car, talent of the player, automated object detection should be complemented with manual methods. Also, some properties having fuzzy information and the ones, which should be expressed with linguistic terms, like age, cannot be found automatically. In this step, some detailed information about the detected object is gathered from user and *objectInsertionServlet* is invoked for storing objects into the database.

Following the transmission of collected information about object and its properties to server side, the real object is ready to be created and inserted into the database. Although the low level features, extracted by annotator, are stored in database, they are not associated with an object yet. Responsibilities of *objectInsertionServlet* are to get attributes of object from client environment, create real object with properties got from the client, attach low-level information extracted and stored in annotation step and insert the real object to the database. The status of insertion operations is returned to the user. For unsuccessful attempts, an error code is also provided.

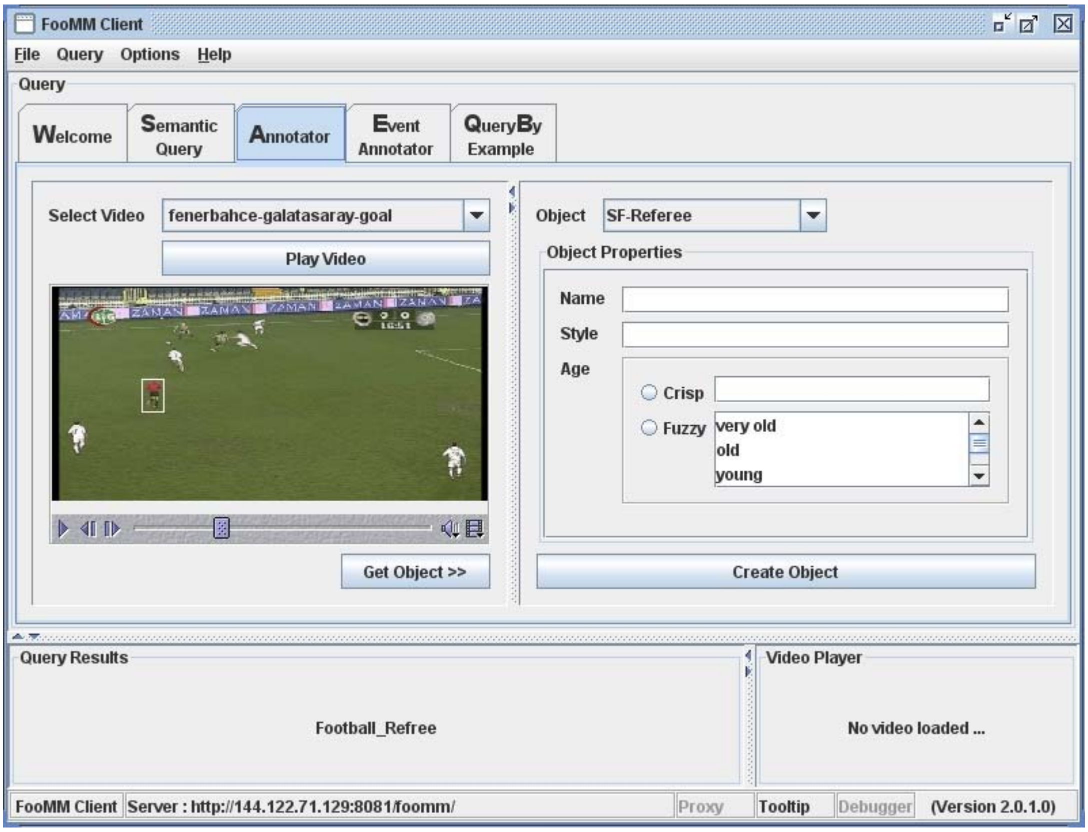


Figure 5-8 A Screenshot from Object Detection and Insertion Interface

Index structure is constructed at database startup. Since the new object may not exist at startup time, index component should be informed about the new object and low-level feature information should be sent to the index structure, which is another responsibility of *objectInsertionServlet*. The serialized low-level feature, obtained from the object annotator module and attached to the object, is redirected to index structure after successful completions of insertion operations.

Since object annotation and creations are separated, if a user cancels object insertion process after finishing annotation step, the low level features, extracted by object annotator and stored in database became garbage. This kind of waste should be cleaned regularly.

The UML sequence diagram for object annotation is given in Figure 5-9.

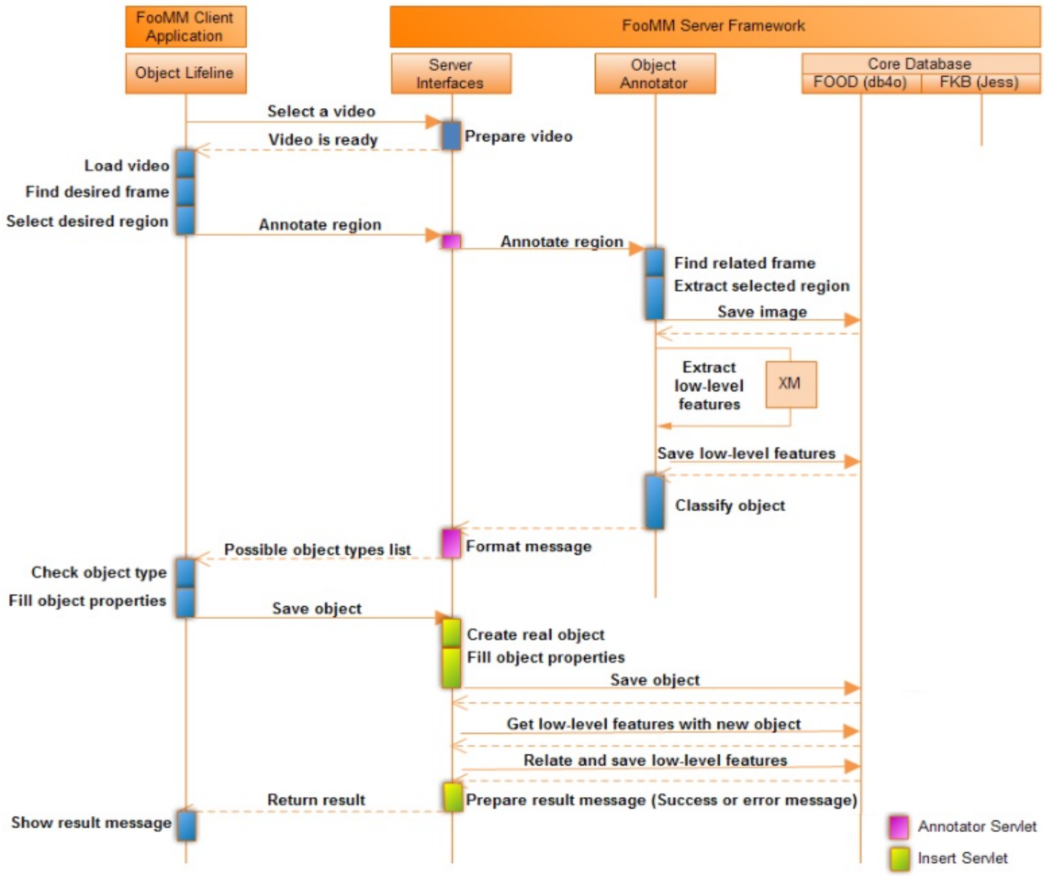


Figure 5-9 UML Sequence Diagram for Object Annotation

5.4.1.2 Event Annotation and Insertion

Events are another type of semantic content that should be extracted from multimedia materials and inserted into database. While low-level information is used for object extraction, some high-level semantics, objects and their spatial relations are required for event detection. Therefore, event annotation starts with object extraction.

Objects, which participate in the event, should be extracted before event detection is started. Since we just need object types, and spatial information of objects in a group of frames, full automatic or semi-automatic methods can be used for object extraction. Object annotation and insertion task, given in previous sub-section is a semi-automatic method since human intervention is required. Also, a full automatic object annotation, detailed information about which is given in the following sub-section, can be used in event detection task. No matter which method is used, *objectAnnotationServlet* is employed for object extraction. Whenever a group of objects is extracted and ready to be annotated for events, *eventAnnotationServlet* is invoked.

In the scope of this thesis, an event ontology is developed for sample implementation of IFooMMD architecture. Detailed information about event ontology is presented in latter sections.

Following adjustments on position and dimension information of objects, *eventAnnotationServlet* invokes event annotator module with calculated spatial information and desired event ontology definitions. Also, video meta-model information is provided for processing event ontology.

Figure 5-10 A Screenshot from Event Detection and Insertion Interface

Considering temporal changes in spatial information of objects, and utilizing the video meta-model as well as the proposed domain events ontology, event annotator module makes some decisions about the possible events in the selected portion of video. In the evaluation step, objects and their relations are extracted and possible spatial change definitions are examined in temporal dimension. After checking relations between spatial change individuals and events, an inference is made about possible events and concepts, with some degrees of similarity and partial memberships.

Possible events, returned from the event annotator are formatted and ordered according to membership degrees by *eventAnnotationServlet*, and returned to the user to gather some properties that cannot be detected automatically.

In client side, after receiving the possible event types and their inclusion values, the most possible event type is selected. Although a full automatic event detection method is used, some user driven tasks are needed to obtain attributes of events after correcting the event type, in case the inference is wrong. Using the detected and user-provided event information, *eventInsertionServlet* is called.

The event annotation is completed after real event entity is created and inserted into database by *eventInsertionServlet.*

5.4.2 Automatic Object Annotation

Object insertion is the slowest function of the proposed study. The biggest portion of object insertion time is consumed by manual operations that cannot be executed automatically. Although, object types and positions can be detected automatically, some properties that cannot be extracted from the visual objects and fuzzy linguistic terms should be provided manually. Due to user intervention, a complexity for insertion function cannot be calculated and the average completion time cannot be estimated.

Type and spatial information of objects are adequate for detecting possible events in event annotation process. The back-number of a player or age information has nothing to do with event annotator module. Therefore, to be used in event annotator module, an automatic approach for finding objects and their spatial information in a group of frames is added to this multimedia framework.

Integrated object annotator module is capable of detecting objects in a region of a frame. If temporal and spatial segmentation of frames can be executed automatically, the object annotator may detect the objects inside them and thus, objects in a video can automatically be annotated.

Using a linear approach, an automatic temporal and spatial segmentation ability is added to the system to make system capable of automatically detect a group of objects in a video. Considering the given parameters, a number of frames are extracted and using a fixed-size partitioning method, these frames are segmented automatically. All segments are evaluated by the object annotator module and all possible objects in a given piece of video are automatically detected and extracted.

In automatic object annotation, segmentation parameters should be selected carefully. To detect an object in a segment, low-level features of that object should be dominant in that region, since classifications of objects are based on them. Therefore, fixed segment size should be small. But, since segments are not combined and evaluated as groups, the size of the segment needed to be at least as big as the biggest object in frames.

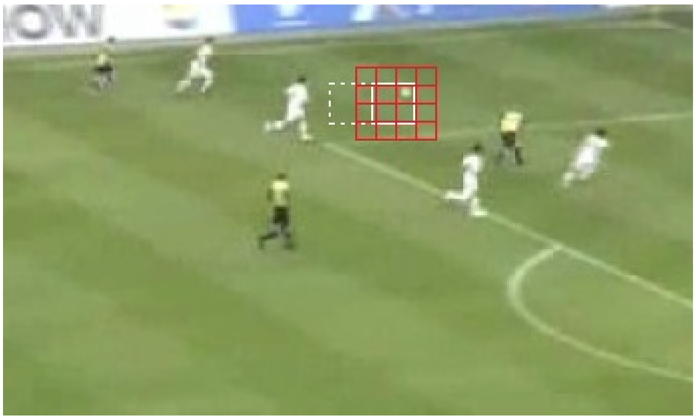
Also, a refinement parameter can be assigned to sharpen the boundaries of objects. It is a tradeoff between the speed and the success of automatic annotation. When an object is detected in a region, if desired, the automatic annotate operation may partition the region into smaller segments and a refined progressive search is executed in these smaller granularity pieces. A sample representation of automatic object annotation and refinement process is given in Figure 5-11.

5.4.3 Search Operations

The implemented IFooMMD system allows two types of semantic queries to be executed in database; *text based* and *similarity based (query-by-example)* semantic queries. In order to execute queries, some web services and servlets interfaces are implemented in coordinator structure of the server. Also, in the prototype client application, user interfaces are developed for constructing queries in a human friendlier way.

Text based semantic querying ability is inherited from core database module, adapted for IFooMMD architecture and extended for news domain. Objects can be searched and retrieved by providing desired textual attributes, including uncertain ones. Also fuzzy linguistic terms can be used in this type of searches.

(a) Fixed Size Segmentation and Object Annotation



(b) Progressive Refinement Process

Figure 5-11 Automatic Object Annotation

Query-by-example, which is a special kind of similarity based semantic querying, is evaluated with the collaboration of object annotator module and index structure. Database is scanned with the help of index structure in order to find similar objects to the given one.

In all types of searches, whenever applicable, the JMFPlayer bean displays visual objects.

5.4.3.1 Text Based Semantic Querying

As in traditional databases, the prototype IFooMMD system is capable of executing text based queries on previously extracted semantic contents. The properties and attributes of semantic objects are used in evaluating such queries. Beside crisp predicates, uncertain values and query conditions are acceptable in the system. Inference mechanism is activated whenever required.

SearchWeatherReportPerson, *SearchWeatherReportObject* and *SearchWeather-ReportEvent* servlets are implemented for searching actor, object and event entities for weather report domain. Similar servlets for each domain are developed for executing text based queries. Also some web services are implemented to enable other systems benefit from this system. Detailed information and implementation steps of developed web services are given in the last section of this chapter.

In client side, for constructing text based queries in a visual manner, a user interface is created at runtime using an XML file that defines the domain and its entities. Current domain and entities supported in prototype IFooMMD system are given in Table 4.

Text based query mechanism starts when user selects an object type from textual query section of query builder [Figure 5-12]. The user fills the desired properties of searched object using some crisp or uncertain values. For example, for a player whose age is not crisply known, user may provide a linguistic fuzzy term for age information; infant, young, old or even some combination of them.

When evaluating a query, all relevant objects, filtered by crisp query predicates, are brought to the bridge from database, DB4O. If any rule is needed to be fired, objects in the bridge are shifted to the working memory of knowledge base. After checking all rules and firing needed ones, knowledge base send resulting objects back to bridge and the objects, meeting all predicates of the query, are serialized into XML messages and returned to the client application.

Table 4 Domains, Sub-Domains and Semantic Entities

Domain		Object Type	Actor Type	Event Type
Accident		Car	Doctor	Accident
		BrokenCar	DeadPerson	Threatment
		Ambulance	Fireman	
		FireEngine	Nurse	
		PoliceCar	Policeman	
		Road	WoundedPerson	
NewsReport		Camera	Anchorman	Interview
		Computer	Audience	Meeting
		Microphone	CameraMan	NewsPresentation
		Studio	Reporter	PressMeeting
			Speaker	Reading
			Visitor	Speaking
WeatherReport		Map	WeatherReporter	Marking
		MarkingStick		Speaking
		WMicrophone		
Sport Domain	Basketball	BasketBallBall	BasketballPlayer	Block
		BasketballHop	BasketballReferee	Steal
		SportArena	BasketballSpectato	Rebound
			Coach	Foul
				Point
	Footbal	FootballBall	Player	SideKick
		Stadium	Referee	Shot(Goal)
				Pass
	Tennis	Field	TennisPlayer	Ace
		Net	TennisSpectator	Fault
		Racket		Point
		TennisBall		

If query predicates contain uncertain information, fuzzy processors are activated in which component requires making fuzzy decisions: database or knowledge base. Simple fuzzy attributes are evaluated in fuzzy processor of FOOD structure in the system, whereas in the knowledge base component, fuzzy processors are used when making similarity based fuzzy decisions.

Since index mechanism is not activated for textual queries, capabilities and efficiency of this type of searches are restricted with the abilities of FKB and FOOD, in our system JESS and DB4O with fuzzy processors.

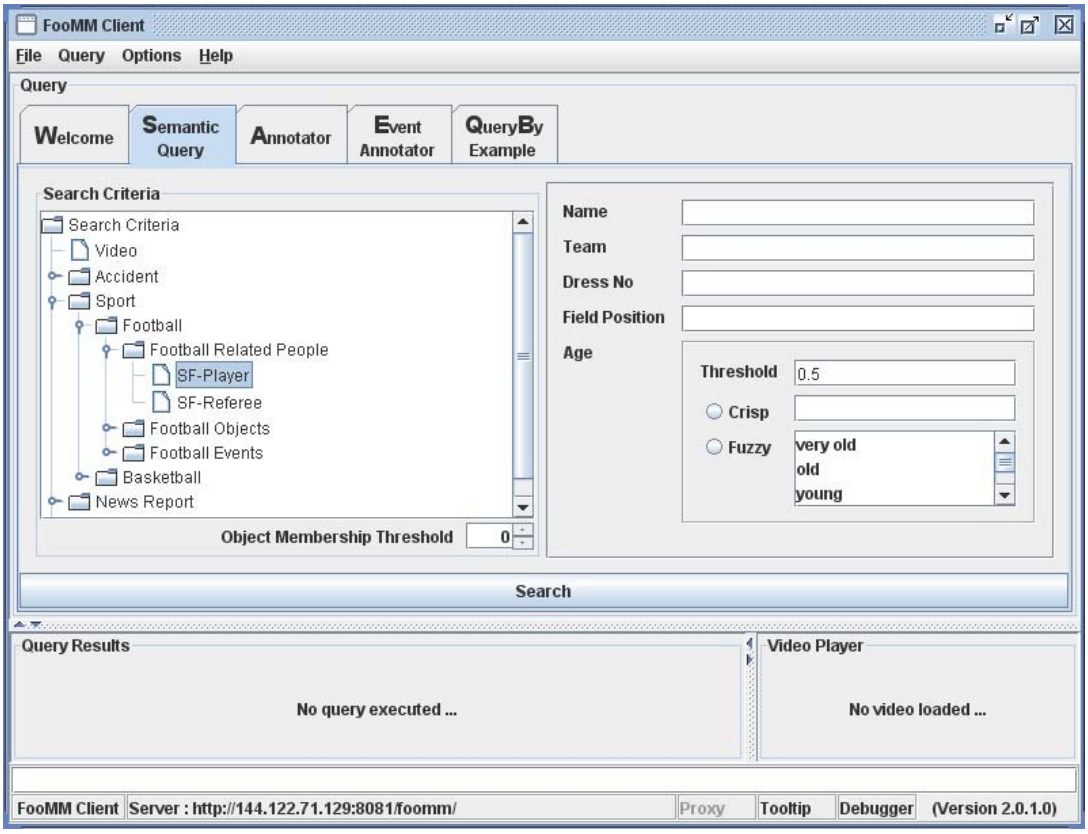


Figure 5-12 A Screenshot from Text Based Semantic Query Interface

5.4.3.2 Similarity Based Semantic Querying (Query-By-Example)

In multimedia databases, similarity based queries are more meaningful than text based searches. A user may not know the name or singer of a piece of music, although the melody is in mind. Or having a picture of a girl or a castle, about which there is no textual information (such as name or location), the user may want to find other information or visual objects relevant to them. In order to meet these considerations, a kind of similarity search mechanism for visual objects is utilized in IFooMMD system. In this type of search, a piece of image is given as query predicate and some similar images, having common visual properties with the given one are returned in the order of similarity percentage.

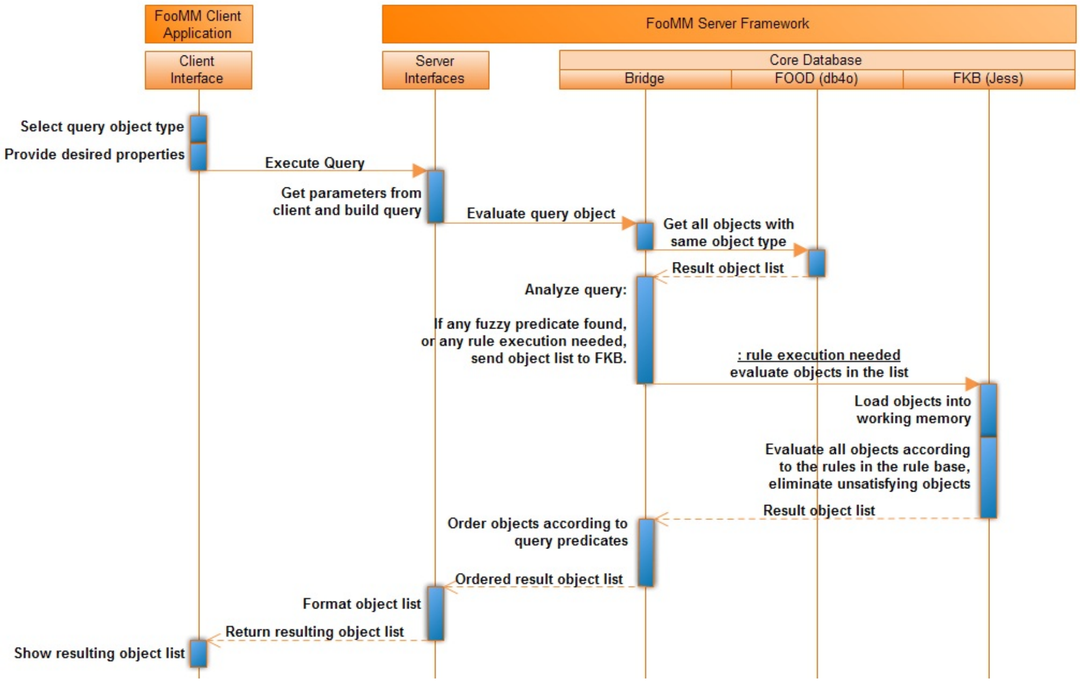


Figure 5-13 Workflow of Text Based Semantic Querying

The execution of query-by-example type queries starts with the selection of a region on the desired frame of loaded video [Figure 5-14]. This selection is the predicate of the query and the features of this region are used in calculation of distance to other objects in database.

Section information about selected region is sent to the *queryByExample* servlet interface. Besides establishing communication between server and client for similarity search, this servlet provides interaction of inter-communicating components: annotator module and index structure, which are the required modules in similarity based query evaluations.

After gathering the information from client, *query-by-example* redirects this information to object annotator module for obtaining low-level features and classification information. Possible object type information is utilized in the last step of query evaluation for filtering and ordering results coming from index structure. The extracted low-level features are serialized and sent to index structure for further processing.

Using the serialized low-level features, index structure finds the clusters that possible similar objects locate. For each object in these clusters, relative distance of features to given object features is calculated and the most similar k objects are returned to the *query-by-example* servlet.

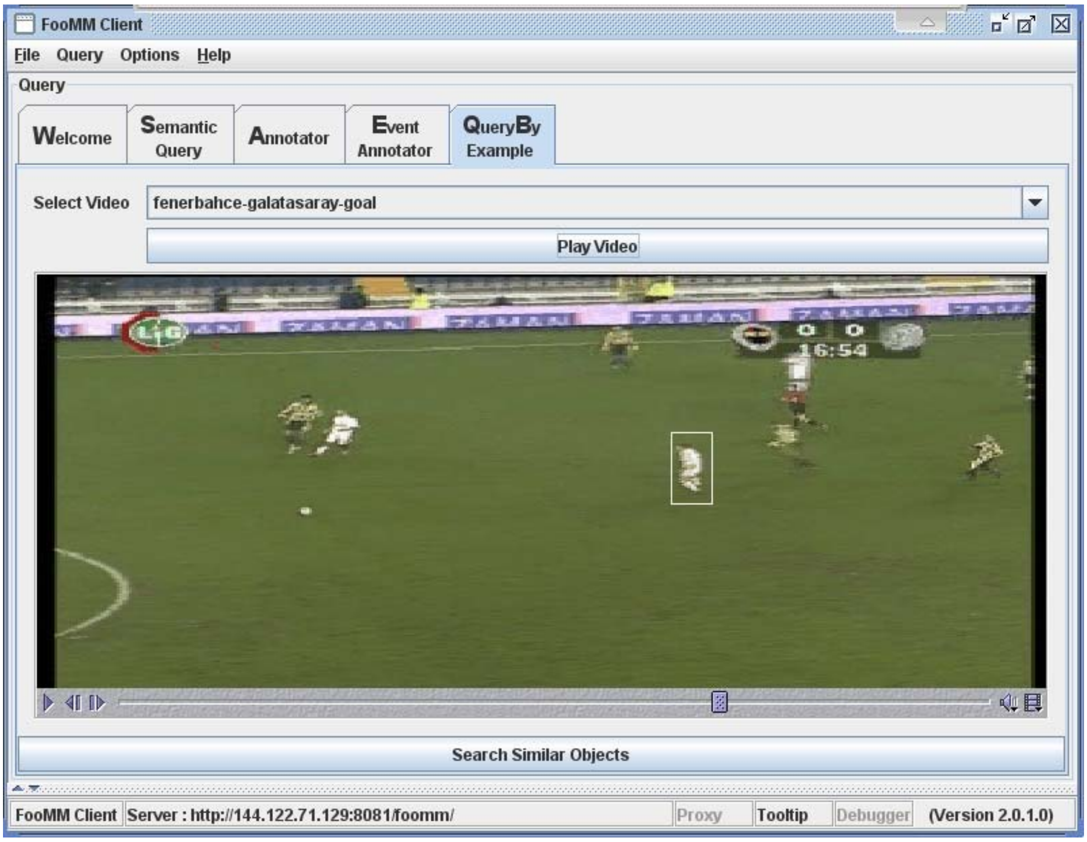


Figure 5-14 A Screenshot from Similarity Based Semantic Query Interface, *QueryByExample*

The resulting objects, coming from index structure, are filtered by the possible object categories, obtained from annotator module in low-level extraction step, and ordered using similarity degrees. After being formatted, the results are sent to the client as an XML message and listed in the query result region of interface. The workflow of similarity based querying is given in Figure 5-15.

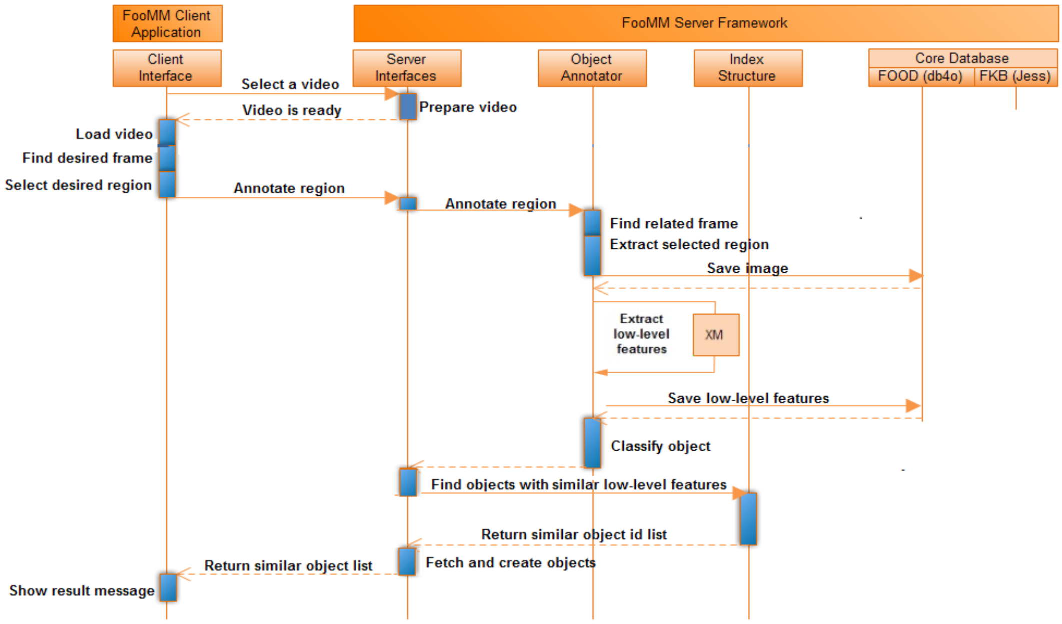


Figure 5-15 Workflow of Similarity Based Querying

5.5 Event Ontology

In the scope of this thesis, an event ontology for football domain is developed. The generated event ontology is compatible with the meta-model proposed in study [46], which is integrated as event annotator module in implemented system. Semantic entities, their possible temporal or spatial relations, individuals indicating temporal changes, are clearly defined in the developed ontology.

Collaborating with event annotator module, the event ontology is used in event extraction process. Topological relations, distance relations and positional relations of defined objects are extracted by event annotator module. Using the definitions of events and rules in ontology, temporal changes in spatial information of object are detected and decisions about some possible events are given using the inference mechanism in event annotator. In the created event ontology for football domain, sidekick, goal and pass events are defined.

A sample definition for goal event is given in Figure 5-16. In the given goal definition, three object types are used; *player*, *ball* and *fortress*. Using the topological and positional relations, four spatial relation components are defined; *ball with player*, *ball away from player*, *ball away from fortress* and *ball inside fortress*. The order of change in spatial relation components defines spatial changes; *send ball* and *ball going to fortress*. Spatial changes in time dimension is used for detecting *goal* event.

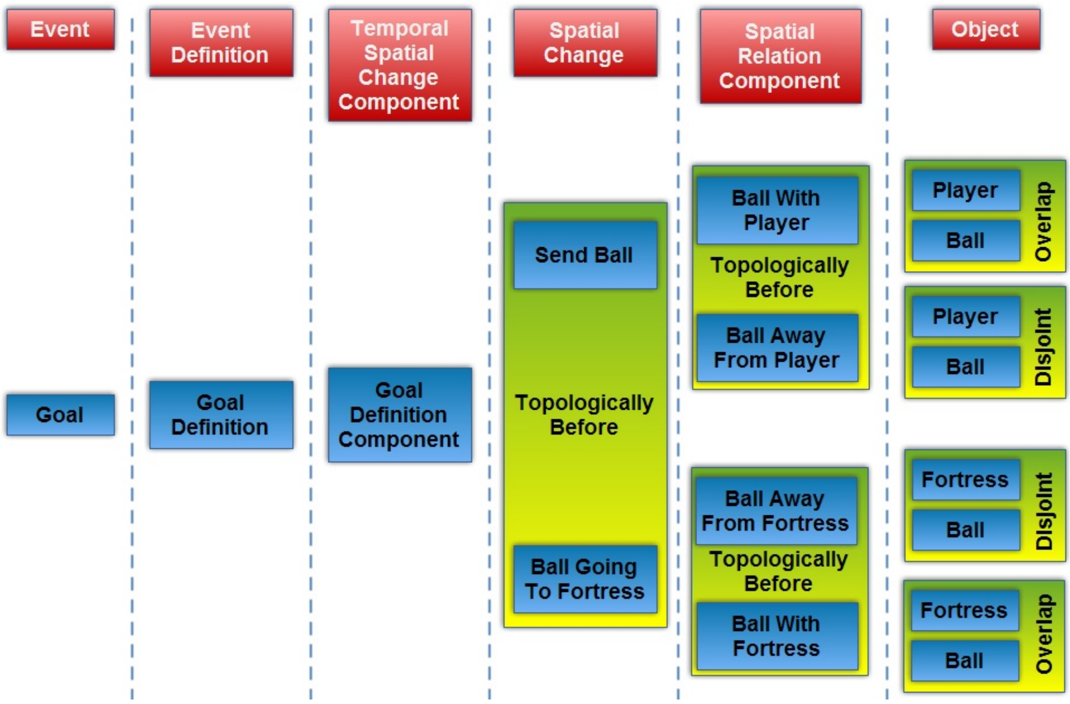
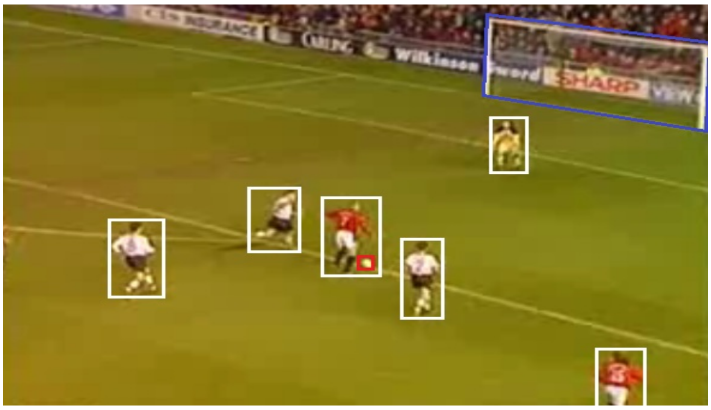


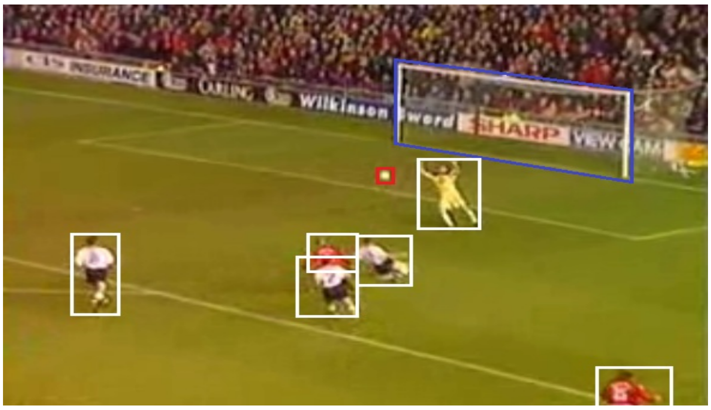
Figure 5-16 Ontology Definition for Goal Event

Using the goal definitions, a sample goal detection frame sequence is given in Figure 5-17. The steps for inferring a goal event are:

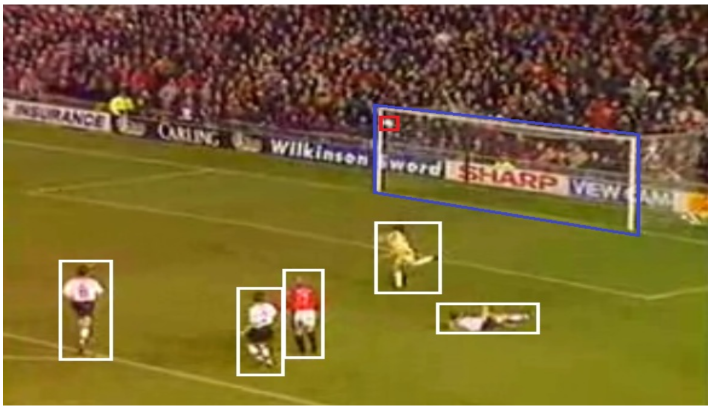
1. In each frame player, ball and fortress objects are extracted. In Figure 5-17, player, ball and fortress objects are marked with white, red and blue shapes, respectively.
2. For each frame, spatial relations of objects are examined.
    a. In the first frame, since there is a ball, topologically *overlap* with a player, there is *ballWithPlayer* spatial relation.
    b. In the second frame, since ball and player objects are *disjoint*, a *ballAwayFromPlayer* spatial relation exists. Besides, since ball and fortress object are also *disjoint*, there is one more spatial relation: *ballAwayFromFortress*.
    c. In the last frame, ball object and fortress are topologically *overlap*, resulting in a *ballWithFortress* relation.
3. In this step, the spatial relation components are mapped to spatial change components.
    d. Since there is a *ballWithPlayer* relation before *ballAwayFromPlayer* relation, according to the goal definition in Figure 5-16, there is a *sendBall* spatial change component.
    e. In former frames, there is a *ballAwayFromFortress* spatial relation and in last frame, there is a *ballWithFortress* spatial relation. Therefore, there is a *ballGoingToFortress* spatial change in latter frames.
4. In this step, using the spatial change components, temporal spatial change components are inferred.
    f. There are two spatial changes in the given frame sequence; *sendBall* and *ballGoingToFortress*. According to the temporal relation between these spatial changes, in other words, since *ballGoingToFortress* happens later than *sendBall*, the *shootDefinitionComponent* is detected.
5. Since *shootDefinition* contains only *shootDefinitionComponent*, the shoot or goal event is inferred.



(a) First Frame



(b) Second Frame



(c) Third Frame

Figure 5-17 Event Detection Example Frames

In the sample evaluation of a goal event, for clarification issues, only the related ontology inferences are given. For instance in step 2-a, there is also, *ballAwayFromFortress* spatial relation exists. But since this relation is not used in inference of the goal event in this scenario, it is not given in step 2-a.

The system is also capable of making fuzzy decisions. For example, in the first frame, the ball and player objects are topologically *overlap*. According to the created event ontology, this relation is resulted as a *ballWithPlayer* spatial relation with full inclusion value, 1. Instead of a full overlap, a *partial overlap* relation would also result as a *ballWithPlayer* spatial relation but with inclusion value 0,7. These inclusion values enable sample IFooMMD system to make fuzzy inferences.

5.6 Query Web Services

Web services enable sharing functionality among different applications even if they are running on different platforms. In the scope of this thesis, a group of web services are developed for querying the implemented IFooMMD system using different client applications.

During the service generation, the following steps are followed;

1. For each service, a service endpoint interface is implemented. A service endpoint interface (SEI) is the starting point for developing a web service. It is a java interface, in which all the methods that can be invoked by clients are declared. In this step, all classes and configuration files, which are compatible with server-framework objects, are developed. A sample SEI structure is given in Figure 5-18.
2. All classes, configuration files and SEI is compiled.
3. Description language files (WSDL) for all web services are created using Java2WSDL tool for configured platform. In this step, the built-in bean-serializers are used for serializing objects (such as wsFootballPlayer, wsBasketballBall) and return types.
4. Deployment and undeployment files are created. In deployment files, all types, functions and serializer beans are declared.
5. The AdminClient tool is used for deploying created Web Services.

```
@WebService()
public class SearchSportPerson {
    @WebMethod
    public wsFootballPlayer[] searchFootballPlayer(String name, int dressNumber,
                                                   String crispAge, String fuzzyAge,
                                                   float ageInclusion,String fieldPosition,
                                                   String teamName,float meanThreshold) {
        Player template = new Player();
        ArrayList<UncertainAttribute> uncertainAttrList = new ArrayList<~>();
        :
        ArrayList queryResult = Bridge.query(template, uncertainAttrList, meanThreshold);
        :
        for (int cnt = 0; cnt < queryResult.size(); cnt++)
            retVal[cnt] = new wsFootballPlayer((Player) queryResult.get(cnt));

        return retVal;
    }

    @WebMethod
    public wsFootballReferee[] searchFootballReferee(String name,
                                                     String crispAge, String fuzzyAge,
                                                     String style, float meanThreshold) {
```

Figure 5-18 Picture from a SEI Class, SearchSportPerson

Table 5 The List of Web Services

Web Service	Web Methods
SearchVideoObject	searchVideo
SearchSportObject	searchFootballBall
	searchFootballStadium
	searchBasketballBall
	searchBasketballHop
	searchBasketballArena
SearchSportPerson	searchFootballPlayer
	searchFootballReferee
	searchBasketballPlayer
	searchBasketballReferee
	searchBasketballSpectator
	searchBasketballCoach

CHAPTER 6

TEST AND EVALUATION

6.1 Querying Capabilities

Text based queries can be categorized into two groups; object based queries and event based queries. Since actors of events are derived from object class, queries related with actors are classified as object based queries. Object based queries may contain fuzzy and crisp predicates:

- Retrieve the name of the referees in “Beşiktaş”-“Bursaspor” match (crisp predicate).
- Retrieve very popular weather reporters (fuzzy predicate).
- Retrieve the Ambulance objects of hospital “Yaşam” (crisp predicate).
- Retrieve old defense players in team “Trabzospor” (crisp and fuzzy predicates).

In the last query, both crisp and fuzzy linguistic terms are used. The *old* word, defining the age of the player, is a fuzzy predicate, and *defense* and the team name *Trabzonspor* are crisp predicates. In evaluation time, the football players, whose *fieldposition* attribute is defense and the name of the *team* attribute is Trabzonspor, are found and inclusion degrees for ages of all these players to fuzzy set *old* is computed. In this computation, the similarity matrix is employed for players with fuzzy ages, or membership functions are utilized for crisp ages. Candidate objects with inclusion degrees higher or equal to a given threshold value are returned.

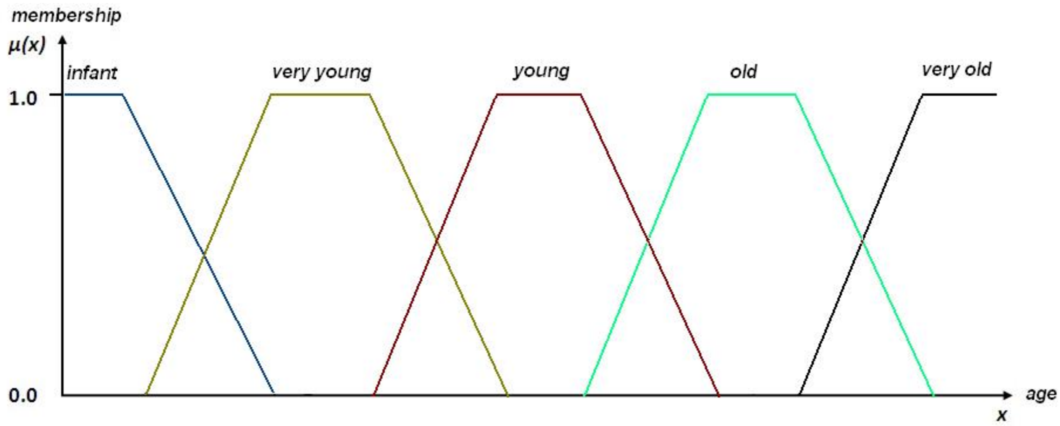


Figure 6-1 Membership Function Graphs for *infant*, *very young*, *young*, *old* and *very old*

Table 6 Sample Similarity Matrix for Fuzzy Attribute *age*

	infant	very young	young	old	very old
infant	1.0	0.4	0.1	0.0	0.0
very young	0.4	1.0	0.7	0.0	0.0
young	0.1	0.7	1.0	0.2	0.0
old	0.0	0.0	0.2	1.0	0.7
very old	0.0	0.0	0.0	0.7	1.0

Although being represented with crisp words, all event queries includes uncertainty due to the detection method. Some sample text based event queries are:

- Retrieve goals in the first half of semifinal match.
- Retrieve shots, kicked from 25m or farther.
- Retrieve harsh fouls, resulted with a red card.
- Retrieve side-kicks, the shooter is "Mehmet".

Also, answers for more complex queries may be found executing several queries and combining them:

- Retrieve goal events in final match, scored by “Hakan” and assisted by “Okan”.

In the evaluation of this query, *goal* events in final match should be found. Also pass events in that match should be detected and using the temporal correlation between goal and pass events, the query should be evaluated.

- Retrieve goals, kicked from middle field by “Hagi”.
- Retrieve side-kicks, resulted with a goal.

Beside object and event queries, text based cataloging queries about multimedia objects are allowed:

- Retrieve videos, named as “2010 Semifinal Match “

6.2 Implementation and Test Environments

Details of implementation and test environment are as follows:

- The implementation is developed on Windows XP, Intel Core 2 Duo 2.13 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 1.6.0_03 on IntelliJ IDEA Community Edition 9.0.1. The user interfaces and main flow of client is developed on NetBeans 6.8.
- The Java implementation contains nearly 18000 lines of code in 70 files for server component and 4500 lines of code in 20 files for client component (Integrated modules’ information is excluded).
- As the web server of the application, Apache Tomcat 6.0.24 is used. To deploy web services, Axis 1_4 SOAP engine is used.
- Some important Java libraries used are:
    - Java Media Framework library, for handling multimedia operations.
    - db4o-6.3-java5, for DB4O database operations.
    - Jess, for knowledge base operations.

6.3 Dataset

Two types of tests were executed to evaluate the performance of the implemented IFooMMD system; speed tests and accuracy tests.

Time measurements were performed on a video data set containing 5 football and 2 basketball video sequences with nearly 10 thousand frames. For semantic information extraction and similarity search tests, randomly selected video sequences and their frames were used. While evaluating exact match query performance, the semantic entities, extracted in semantic information extraction tests were used.

Two tests were conducted in accuracy tests. In the first test, in which automatic segmentation was utilized, five 10 second football video shots were used. For each second of video data, two frames were extracted and each frame was automatically segmented into two hundred pieces, each of which was annotated automatically. In the second test of accuracy evaluation, football videos, used in time measurement tests, were used.

6.4 Tests, Results and Evaluation

6.4.1 Time Measurements

Average execution times of several processes are given in Table 7. The execution time of each process was measured as the difference of start and completion time of process. Since a thin-client architecture was employed, in order not to be affected from network issues, all tests were evaluated on the server.

The speed evaluation of the system was realized in three steps:

- Evaluating semantic information extraction process,
- Testing exact match query abilities,
- Testing similarity search ability, query-by-example.

In semantic entity detection step, since it is not possible to find some fuzzy and hidden entity specific information automatically, such as age of a player and harshness of a foul event, some user intervention is required. Object insertion can only be executed after these entity attributes are obtained. Therefore, while evaluating semantic information extraction capability, detection of semantic content and insertion of found entities were examined separately.

The processes, which include extraction of low-level features like object detection and query-by-example type searches, consumes longer times than other processes, such as exact match querying. A great portion of time, spent for these processes are consumed by removal of image segments from whole image and extraction of low-level features of these segments by MPEG-7 reference software, XM.

The execution times of object and event detection processes are slightly change according to the size of the image segments. Also, since an object-oriented database is used, the size of the objects effect the insertion and retrieval time. The number of predicates, whether crisp or uncertain, has no impact on query execution times. But the execution times of queries are proportional to the object amount in database.

Table 7 Average Execution Times for Information Extraction Retrieval Tests

Test Type	Detailed Test Information	Time (ms)
Semantic Information Extraction	Object detection in a selected region	4925
	Object insertion	225
	Event detection using an object set	1386
	Event insertion	228
Exact Match Query	Search for all objects of a specific type	125
	Using single crisp predicate	141
	Using multiple crisp predicates	141
	Using single fuzzy predicate	143
	Using multiple fuzzy predicates	152
	Using both crisp and fuzzy predicates	147
	No rows returned	120
	Single row returned	120
	Multiple rows returned	135
Similarity Search	Query-By-Example	4800

6.4.2 Accuracy Tests

As well as the speed of the system, another important criterion in evaluating the success of the implemented architecture is the accuracy of the system. Two tests were executed to evaluate the success of the implemented architecture. In both, object and event extraction processes were executed automatically and all the semantic information extraction processes, explained in this study were used. The difference between tests was the method of segmentation process.

Since fixed size segmentation is used in automatic object annotation module, an object is accepted as correctly extracted, if the examined segment includes the object or part of it. Similarly, if the time boundaries of an automatically extracted event instance intersect with the boundaries of the real event, it is accepted as correctly extracted.

The precision, recall and f-measure values, which are metrics, commonly used for accuracy evaluation of retrieval systems, are evaluated according to given formulas:

$$Precision = \frac{number\ of\ Relevant\ Retrieved}{number\ of\ All\ Retrieved} \tag{6.1}$$

$$Recall = \frac{number\ of\ Relevant\ Retrieved}{number\ of\ All\ Relevants} \tag{6.2}$$

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{6.3}$$

As implied in the formulas, precision shows the ratio of the correctness of the query results, as well as recall gives the ratio of the completeness of the result set. F-measure is the average best point in the precision-recall graph.

Before accuracy evaluation, to refine event definitions in ontology, a pre-test procedure was executed. In this routine, all object coordinates and types were manually detected and event annotator module was activated with this set of entities.

When results of this pre-test routine were examined, it was seen that some event instances were missed or misclassified due to some wrong definitions in sub-components of event definitions. For example, the *ballWithPlayer* spatial relation, used in *pass* event, was defined using o*verlap* operator between *ball* and *player* object. But in some frames, although a player has the ball, due to position of the camera or due to the player style, the ball and player may be seen as disjointed, but near; causing an undetected *ballWithPlayer* spatial relation, therefore, missed pass event.

Using decisions of the event annotator module in this step, the event definitions were tuned and the manual segmentation and object extraction procedure was repeated until all events were extracted correctly. The results of this pre-test routine with former and subsequent values are given in Table 8 and Figure 6-2. After completing event ontology refinement process, the system was ready for the accuracy tests.

Table 8 Event Definition Refinement Results

	Event Name	Manual	Correct	False	Missed	Prec (%)	Rec (%)
Before	pass	8	5	-	3	100.00	62.50
	sidekick	1	1	-	-	100.00	100.00
	shot(goal)	2	1	-	1	100.00	50.00
		11	7	-	4	100.00	63,63
After	pass	8	8	-	-	100.00	100.00
	sidekick	1	1	-	-	100.00	100.00
	shot(goal)	2	2	-	-	100.00	100.00
		11	11	-	-	100,00	100,00

(a) Precision Values **(b) Recall Values**

Figure 6-2 Event Refinement Metrics

In the first test, automatic object annotator module was activated and using the linear segmentation approach mentioned in previous chapter, automatic image segmentation, object and event extraction were executed. The event annotator module was fed by object annotator module and in the event extraction steps, automatically segmented and annotated objects were used. Object and event counts with precision, recall and f-measure scores of first test according to different threshold values are given in Table 9 and Table 10. Also, corresponding metric curves; precision, recall and f-measure graphs are presented in Figure 6-3 and Figure 6-4.

Since the object annotator module is forced to make a classification for each segment, the detected object count is calculated very high. But, even for high threshold values, the ratio of correctly detected object count to all detected objects is not satisfying and resulting in low recall and f-measure values. In order to understand the effect of the fixed width segmentation, a second test was executed. The automated segmenting was disabled and images were segmented manually, but object and event annotations were executed automatically. The results and metric scores are given in Table 11 and

Table 12, with corresponding curves in Figure 6-5 and Figure 6-6.

Table 9 Automatic Segmentation: Object Extraction Test Results

	Threshold Values								
	0,50	0,55	0,60	0,65	0,70	0,75	0,80	0,85	0,90
Real Object Count	864	864	864	864	864	864	864	864	864
Detected Object Count	6602	5227	4598	4014	2986	1919	989	431	36
Correctly Detected Count	263	262	254	244	223	191	144	70	8
Precision	0,04	0,05	0,06	0,06	0,07	0,10	0,15	0,16	0,22
Recall	0,30	0,30	0,29	0,28	0,26	0,22	0,17	0,08	0,01
F-Measure	0,07	0,09	0,10	0,10	0,11	0,14	0,16	0,11	0,02

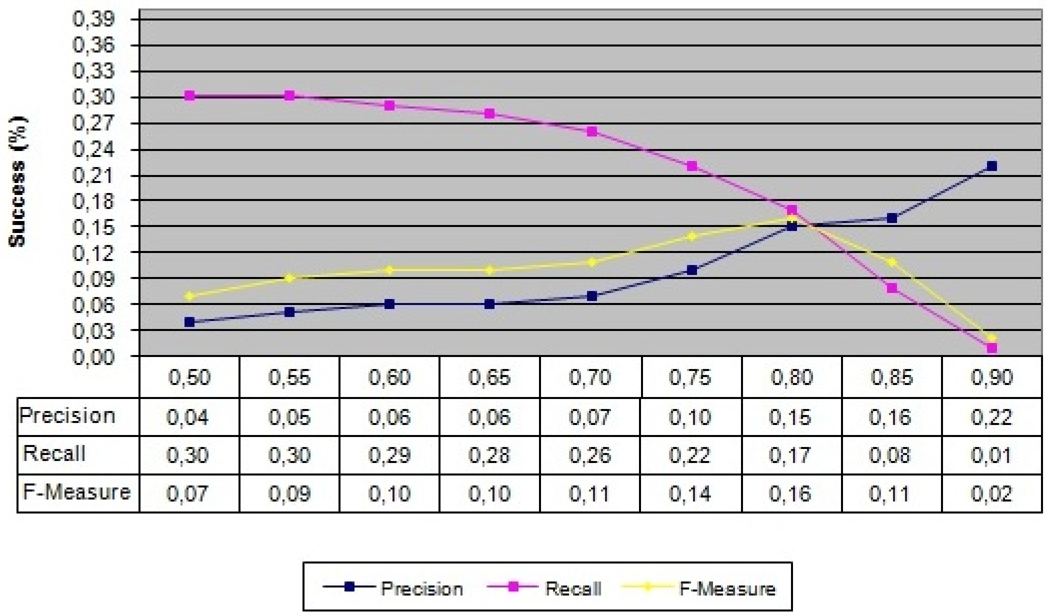


Figure 6-3 Automatic Segmentation: Precision, Recall and F-Measure Curves for Object Extraction Test

Table 10 Automatic Segmentation: Event Extraction Test Results

	Threshold Values							
	0,50	0,55	0,60	0,65	0,70	0,75	0,80	0,85
Real Event Count	9	9	9	9	9	9	9	9
Detected Event Count	11	10	9	8	8	8	5	1
Correctly Detected Count	1	1	1	1	1	1	1	0
Precision	0,09	0,10	0,11	0,13	0,13	0,13	0,20	0
Recall	0,11	0, 11	0, 11	0, 11	0, 11	0, 11	0,11	0
F-Measure	0,10	0,10	0,11	0,12	0,12	0,12	0,14	0

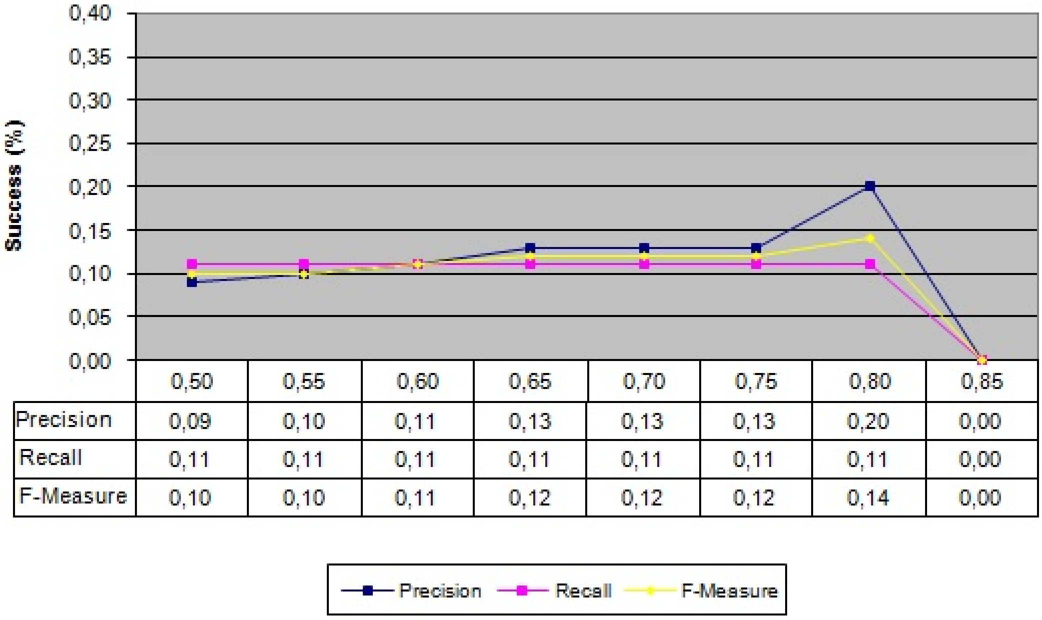


Figure 6-4 Automatic Segmentation: Precision, Recall and F-Measure Curves for Event Extraction Test

Table 11 Manual Segmentation: Object Extraction Test Results

	Threshold Values								
	0,50	0,55	0,60	0,65	0,70	0,75	0,80	0,85	0,90
Real Object Count	143	143	143	143	143	143	143	143	143
Detected Object Count	138	135	134	130	120	110	88	65	19
Correctly Detected Count	90	90	90	89	87	87	70	54	17
Precision	0,65	0,67	0,67	0,68	0,73	0,79	0,80	0,83	0,89
Recall	0,63	0,63	0,63	0,62	0,61	0,61	0,49	0,38	0,12
F-Measure	0,64	0,65	0,65	0,65	0,66	0,69	0,61	0,52	0,21

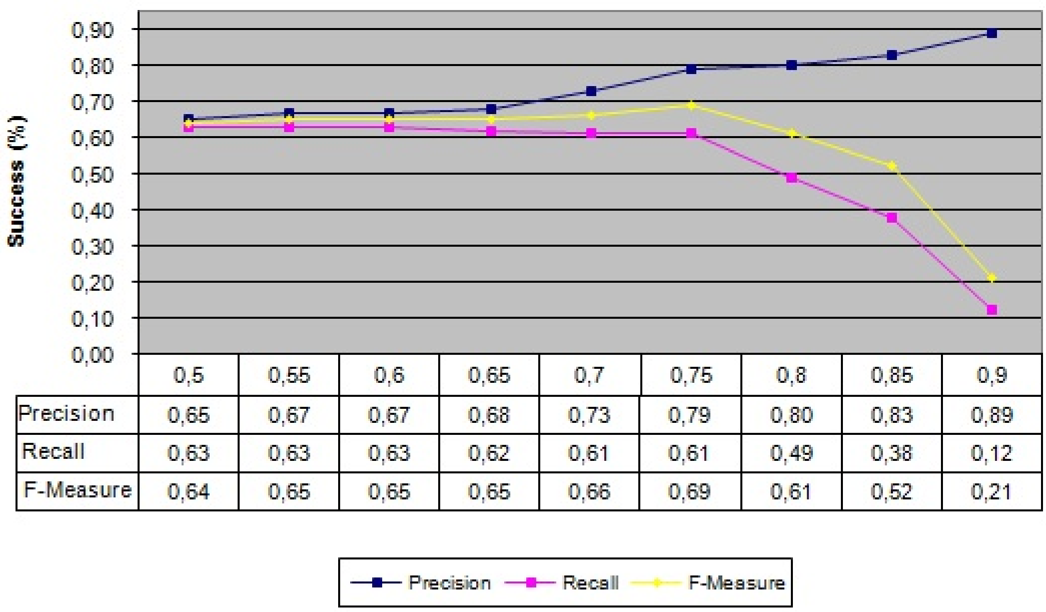
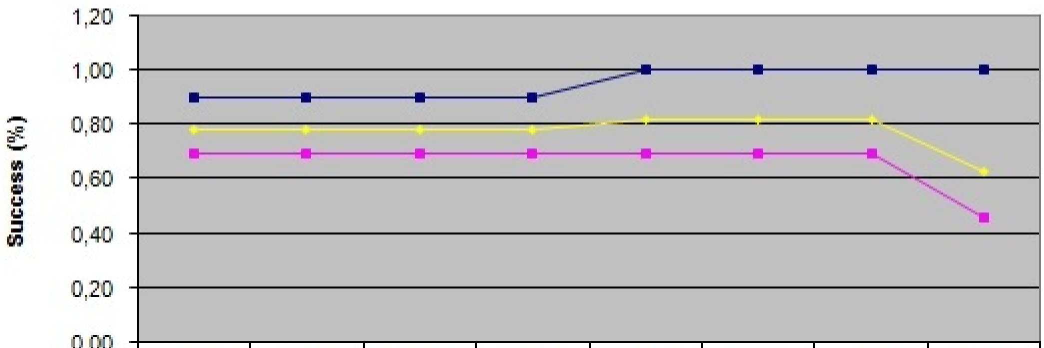


Figure 6-5 Manual Segmentation: Precision, Recall and F-Measure Curves for Object Extraction Test

Table 12 Manual Segmentation: Event Extraction Test Results

	Threshold Values							
	0,50	0,55	0,60	0,65	0,70	0,75	0,80	0,85
Real Event Count	13	13	13	13	13	13	13	13
Detected Event Count	10	10	10	10	9	9	9	6
Correctly Detected Count	9	9	9	9	9	9	9	6
Precision	0,90	0,90	0,90	0,90	1,00	1,00	1,00	1,00
Recall	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,46
F-Measure	0,78	0,78	0,78	0,78	0,82	0,82	0,82	0,63



	0,50	0,55	0,60	0,65	0,70	0,75	0,80	0,85
Precision	0,90	0,90	0,90	0,90	1,00	1,00	1,00	1,00
Recall	0,69	0,69	0,69	0,69	0,69	0,69	0,69	0,46
F-Measure	0,78	0,78	0,78	0,78	0,82	0,82	0,82	0,63

Figure 6-6 Manual Segmentation: Precision, Recall and F-Measure Curves for Event Extraction Test

Since sizes of objects, such as a ball and a player, are not close, and images are not segmented according to semantic content, the selected segmentation methodology decreases the performance. A good segmentation or object boundary detection algorithm may result in better solutions. In the original study [2] of integrated object extraction module, n-cut segmentation algorithm is utilized and test solutions, similar to our manual segmentation test results are obtained. A comparison table, showing precision and recall values for object extraction with different segmentation methods is given in Table 13.

Table 13 Segmentation Method Comparison

	Best threshold	Precision	Recall	F-Measure
Fixed Size Segmentation in our study	0,80	0,15	0,17	0,16
Manual Segmentation in our study	0,75	0,79	0,61	0,69
N-Cut Segmentation in study [2]	0,81	0,70	0,57	0,63

The implemented IFooMMD system and all these performance tests show that, the architecture is successful and applicable for multimedia databases. Since the success of each module depends on other modules and the bottlenecks of each module decrease the overall performance, for a successful implementation, all integrated modules should be selected carefully and compatible with each other.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

A multimedia database, which enable content based retrieval of materials in an easy, less formalized and human-friendlier way, gain more importance lately due to recent improvements in multimedia technology and increasing popularity of multimedia devices.

In this thesis, a conceptual model for multimedia database and a prototype application, which is based on MPEG-7 descriptors and is support uncertainty at all levels of abstraction, is developed. Attribute level, object/class level and class/superclass level fuzziness is handled in both object insertion and query construction steps. To deal with uncertain information, the system benefits from a fuzzy knowledge base with the conjunction of a fuzzy object-oriented database.

As well as supporting fuzzy information, the system is capable of extracting semantic information from multimedia data. Although detection of objects and events can be executed full-automatically, since some information about them cannot be obtained without human interaction, such as shoe number of a player or harshness of a foul event, a semi automatic way of object and event insertion is employed.

An index structure, specialized for handling low-level features is also utilized in this study to enable similarity searches, which is an indispensable property of content based retrieval systems.

The generic conceptual data model and proposed architecture, explained in Chapter 4, are designed to support all kind of multimedia information. For proof of concept, the developed sample framework is limited to handle only visual materials due to the coupled modules. A future work is to extend this sample framework to support handling audial and textual multimedia materials and enable extraction of semantic information from them.

Another future direction is to employ a web-crawling facility to gather more information about detected objects and events. Back-number of a player with the team information which can be detected using visual annotators, when searched in web, may result in name and age information, as well as other undetectable knowledge.

Considering the automatic linear segmentation method, a better algorithm may be employed for more meaningful segmentations. Since spatial information of objects is used in detecting event entities, such an improvement may result in better annotations.

In the index structure, since it disrupts clustering information, the individual insertion of semantic content is resulted in rebuilding index structure. This approach is very time consuming and causes some latency in insertion procedure. An improvement for this deficiency can be detection of fragmentation in clustering schema and rebuilding index structure when the fragmentation exceeds a pre-defined limit.

The developed application is able to display the results of queries, whenever the result is a video object. Java Media Framework is used in playing visual materials. Although it is the most current version of JMF, it does not support all video formats and codec. Moreover, the RTP and RTSP support of JMF is very limited, resulting in an unusable streaming capability. A future improvement is needed to overcome the limitations of this tool.

This study uses some visual MPEG-7 descriptors in annotation and building index structure processes. Although the system does not have any limit, only four common descriptors are employed. Using more MPEG-7 descriptors may result in better annotations and improvements in similarity searches. For a future direction, the number of used descriptors can be increased and moreover, the system can be modified to benefit from all types of descriptors including audio and text caption features in multimedia materials.

REFERENCES

[1] M.-S. d. Hacid, C. Decleir, and J. Kouloumdjian, "A Database Approach for Modeling and Querying Video Data," *IEEE Transactions on Knowledge and Data Engineering,* vol. 12, pp. 729-750, 2000.

[2] T. Yilmaz, "Object Extraction from Images/Videos Using a Genetic Algorithm Based Approach," *Master thesis, Computer Engineering Department, METU, Ankara, Turkey,* January 2008.

[3] 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* l'Aquila, Italy: Scuola Superiore G. Reiss Romoli, 2000.

[4] O. Marques and B. Furht, "MUSE: A Content-Based Image Search and Retrieval System Using Relevance Feedback," *Multimedia Tools Appl.,* vol. 17, pp. 21-50, 2002.

[5] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by Image and Video Content: The QBIC System," *Computer,* vol. 28, pp. 23-32, 1995.

[6] R. Datta, J. Li, and J. Z. Wang, "Content-based image retrieval: approaches and trends of the new age," in *In Proceedings ACM International Workshop on Multimedia Information Retrieval*: ACM Press, 2005, pp. 253-262.

[7] J. Fan, A. K. Elmagarmid, S. Member, X. Zhu, W. G. Aref, and L. Wu, "ClassView: Hierarchical Video Shot Classification, Indexing, and Accessing," in *IEEE Trans. on Multimedia*. vol. 6, 2004, pp. 70-86.

[8] I. Kompatsiaris, Y. Avrithis, P. Hobson, and M. G. Strintzis, "Integrating knowledge, semantics and content for user-centred intelligent media services: the acemedia project," *in Proceedings of Workshop on Image Analysis for Multimedia Interactive Services,* pp. 21-23, 2004.

[9] P. Natarajan and R. Nevatia, "EDF: A framework for Semantic Annotation of Video," in *In Tenth IEEE International Conference on Computer Vision Workshops (ICCVW'05)*: IEEE Computer Society, 2005, p. 1876.

[10] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Integrated Semantic-Syntactic Video Modeling for Search and Browsing," in *in IEEE Transactions on Multimedia*. vol. 6, 2004, p. 839.

[11] R. S. Aygün and A. Yazici, "Modeling and Management of Fuzzy Information in Multimedia Database Applications," *Multimedia Tools Appl.,* vol. 24, pp. 29-56, 2004.

[12] H. Öztarak and A. Yazıcı, "Structural and Event Based Multimodal Video Data Modeling," *ADVIS 2006,* vol. 4243, pp. 264-273, 2006.

[13] N. B. Özgür, "An Intelligent Fuzyy Object Oriented Database Framework for Video Database Applications," *Master thesis, Computer Engineering Department, METU, Ankara, Turkey,* September 2007.

[14] K. Aas and L. Eikvil, "A Survey on: Content-based Access to Image and Video Databases," 1997.

[15] F. I. Bashir and A. A. Khokhar, "Video Content Modeling Techniques: An Overview; 2002," 2002.

[16] Y. Wang, C. Xing, and L. Zhou, "Video Semantic Models : Survey and Evaluation," *International Journal of Computer Science and Network Security,* vol. 6, pp. 10-20, 2006.

[17] E. Oomoto and K. Tanaka, "OVID: Design and Implementation of a Video-Object Database System," *IEEE Trans. on Knowl. and Data Eng.,* vol. 5, pp. 629-643, 1993.

[18] R. Hjelsvold and R. Midtstraum, "Modelling and Querying Video Data," in *In Proceedings of the 20th VLDB Conference*, 1994, pp. 686-694.

[19] Q. Li and L. S. Huang, "A dynamic data model for a video database management system," *ACM Comput. Surv.,* vol. 27, pp. 602-606, 1995.

[20] D. Montesi and H. Elmagarmid, "VideoText Database Systems," in *ICMCS '97: Proceedings of the 1997 International Conference on Multimedia Computing and Systems* Washington, DC, USA: IEEE Computer Society, 1997, p. 344.

[21] A. Gupta, T. E. Weymouth, and R. Jain, "Semantic Queries with Pictures: The VIMSYS Model," in *VLDB '91: Proceedings of the 17th International Conference on Very Large Data Bases* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991, pp. 69-79.

[22] H. Kosch, "VIDEX: An Integrated Generic Video Indexing Approach," in *In Proceedings of the ACM Multimedia Conference*: ACM Press, 2000, pp. 448-451.

[23] S. Adali, K. S. Candan, K. S. C, S.-s. Chen, K. Erol, and V. S. Subrahmanian, "Advanced Video Information System: Data Structures and Query Processing," in *Multimedia Systems*. vol. 4, 1996, pp. 172-186.

[24] D. A. Tran, K. A. Hua, and K. Vu, "VideoGraph: A Graphical Object-based Model for Representing and Querying Video Data," in *IN THE PROC. OF ACM INT'L CONFERENCE ON CONCEPTUAL MODELING (ER 2000*, 2000, pp. 9-12.

[25] M. E. Donderler, E. Saykol, U. Arslan, and O. Ulusoy, "BilVideo: Design and Implementation of a Video Database Management System," *Multimedia Tools and Applications,* vol. 27, pp. 79-104, September 2005.

[26] U. Arslan, "A Semantic Data Model and Query Language for Video Databases," *Master thesis, Department of Computer Engineering, Bilkent University, Ankara, Turkey,* January 2002.

[27] A. Sunitha and D. P. Govindarajulu, "Multilayer Semantic Data Model for Sports Video," 2007.
[28] R. S. Aygun, A. Yazici, and N. Arica, "Conceptual Data Modeling of Multimedia Database Applications," *Proceedings of the 4th International Workshop on Multimedia Database Management Systems,* pp. 182-189, August 1998.
[29] D. Küçük, N. B. Özgür, A. Yazıcı, and M. Koyuncu, "A Fuzzy Conceptual Model for Multimedia Data with a Text-Based Automatic Annotation Scheme," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 17, pp. 135-152, August 2009.
[30] M. Koyuncu and A. Yazici, "IFOOD: An Intelligent Fuzzy Object-Oriented Database Architecture," *IEEE Transactions on Knowledge and Data Engineering,* vol. 15, pp. 1137-1154, 2003.
[31] A. Yazici, R. George, and D. Aksoy, "Design and implementation issues in the fuzzy object-oriented data model," *Inf. Sci. Inf. Comput. Sci.,* vol. 108, pp. 241-260, 1998.
[32] R. S. Jadon, S. Chaudhury, and K. K. Biswas, "Generic Video Classification: An Evolutionary Learning based Fuzzy Theoretic Approach," 2010.
[33] J. You, G. Liu, and A. Perkis, "A semantic framework for video genre classification and event analysis," *Image Commun.,* vol. 25, pp. 287-302, 2010.
[34] O. Javed, Z. Rasheed, and M. Shah, "A Framework for Segmentation of Talk & Game Shows," *Computer Vision, IEEE International Conference on,* vol. 2, p. 532, 2001.
[35] Y. Guo, B. Yang, Y. Ming, and A. Men, "An Effective Background Subtraction under the Mixture of Multiple Varying Illuminations," *Computer Modeling and Simulation, International Conference on,* vol. 1, pp. 202-206, 2010.
[36] J. Yang, M. Y. Chen, and A. Hauptmann, "Finding person x: Correlating names with visual appearances," in *Dublin City University Ireland*: Springer, 2004, pp. 270-278.
[37] X. Pan, Y. Guo, and A. Men, "Traffic Surveillance System for Vehicle Flow Detection," *Computer Modeling and Simulation, International Conference on,* vol. 1, pp. 314-318, 2010.
[38] N. D. Doulamis, Y. S. Avrithis, N. D. Doulamis, and S. D. Kollias, "A Genetic Algorithm for Efficient Video Content Representation," 1998.
[39] D. Swets, B. Punch, and J. Weng, "Genetic Algorithms For Object Recognition In A Complex Scene," in *in Proceedings, International Conference on Image Processing*, 1995, pp. 595-598.
[40] W. Shi, K. Liu, and C. Huang, *A Fuzzy-Topology-Based Area Object Extraction Method* vol. 48. New York, NY, ETATS-UNIS: Institute of Electrical and Electronics Engineers, 2010.
[41] J. M. Martínez, "Overview of the MPEG-7 Standard (ver. 10). ISO/IEC JTC1/SC29/WG11 N6828," http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, Last date accessed: April, 2010.

[42] R. Datta, J. Li, and J. Z. Wang, "Content-based image retrieval: approaches and trends of the new age," in *MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval* Hilton, Singapore: ACM, 2005, pp. 253-262.
[43] A. Kulkarni, H. Gunturu, and S. Datla, "Association-based image retrieval," *WSEAS Trans. Sig. Proc.,* vol. 4, pp. 183-189, 2008.
[44] A. Dorado, J. Calic, E. Izquierdo, and S. Member, "A rule-based video annotation system," in *IEEE Transactions on Circuits and Systems for Video Technology*. vol. 14, 2004, p. 2004.
[45] L. Ballan, M. Bertini, G. Serra, and A. D. Bimbo, "Video Annotation and Retrieval Using Ontologies and Rule Learning," *IEEE MultiMedia,* vol. 99, 2010.
[46] Y. Yıldırım, "Automatic Semantic Content Extraction In Videos Using A Spatio-Temporal Ontology Model," *PhD. thesis, Computer Engineering Department, METU, Ankara, Turkey,* March 2009.
[47] L. Zelnik-Manor and M. Irani, "Event-Based Analysis of Video," 2001.
[48] M. Chen, C. Zhang, and S. C. Chen, "Semantic Event Extraction Using Neural Network Ensembles," in *Proceedings of the International Conference on Semantic Computing*: IEEE Computer Society, 2007, pp. 575-580.
[49] D. A. Sadlier and N. E. O'Connor, "Event Detection in Field Sports Video Using Audio-Visual Features and a Support Vector Machine," *CirSysVideo,* vol. 15, pp. 1225-1233, October 2005.
[50] D. W. Tjondronegoro and Y. P. P. Chen, "Knowledge-Discounted Event Detection in Sports Video " *IEEE Transactions on Systems Man & Cybernetics, Part A.* , 2010.
[51] C. Xu, J. Wang, H. Lu, and Y. Zhang, *A Novel Framework for Semantic Annotation and Personalized Retrieval of Sports Video* vol. 10. New York, NY, ETATS-UNIS: Institute of Electrical and Electronic Engineers, 2008.
[52] Y. Zhang, C. Xu, Y. Rui, J. Wang, and H. Lu, "Semantic Event Extraction from Basketball Games using Multi-Modal Analysis," *IEEE International Conference on Multimedia and Expo,* 2007.
[53] U. Inyaem, P. Meesad, C. Haruechaiyasak, and D. Tran, "Construction of Fuzzy Ontology-Based Terrorism Event Extraction," *International Workshop on Knowledge Discovery and Data Mining,* vol. 0, pp. 391-394, 2010.
[54] C. Calistru, C. Riberio, and G. David, "Multidimensional Descriptor Indexing: Exploring the BitMatrix," *CIVR'06,* pp. 401-410, 2006.
[55] L. Amsaleg and P. Gros, *Content-based retrieval using local descriptors: Problems and issues from a database perspective* vol. 4. Heidelberg, ALLEMAGNE: Springer, 2001.
[56] A. Guttman, "R-trees: a dynamic index structure for spatial searching," in *Readings in database systems*: Morgan Kaufmann Publishers Inc., 1988, pp. 599-609.
[57] D. A. White and R. Jain, "Similarity Indexing with the SS-tree," in *Proceedings of the Twelfth International Conference on Data Engineering*: IEEE Computer Society, 1996.

[58] N. Katayama and S. i. Satoh, "The SR-tree: an index structure for high-dimensional nearest neighbor queries," in *Proceedings of the 1997 ACM SIGMOD international conference on Management of data* Tucson, Arizona, United States: ACM, 1997.
[59] K. I. Lin, H. V. Jagadish, and C. Faloutsos, "The TV-tree: an index structure for high-dimensional data," University of Maryland at College Park1994.
[60] J. Nievergelt, H. Hinterberger, and K. C. Sevcik, "The grid file: an adaptable, symmetric multikey file structure," in *Readings in database systems*: Morgan Kaufmann Publishers Inc., 1988, pp. 582-598.
[61] H. Y. Lin, "Perfect KDB-Tree: A Compact KDB-Tree Structure for Indexing Multidimensional Data," in *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2 - Volume 02*: IEEE Computer Society, 2005.
[62] "The Hybrid Tree: An Index Structure for High Dimensional Feature Spaces," in *Proceedings of the 15th International Conference on Data Engineering*: IEEE Computer Society, 1999.
[63] T. K. Dang, J. Kueng, and R. Wagner, "The SH-tree: A Super Hybrid Index Structure for Multidimensional Data," in *Proceedings of the 12th International Conference on Database and Expert Systems Applications*: Springer-Verlag, 2001.
[64] T. Urruty, F. Belkouch, and C. Djeraba, "KPYR: An Efficient Indexing Method," *Multimedia and Expo, IEEE International Conference on,* vol. 0, pp. 1448-1451, 2005.
[65] S. Berchtold, C. Böhm, and H.-P. Kriegal, "The pyramid-technique: towards breaking the curse of dimensionality," in *Proceedings of the 1998 ACM SIGMOD international conference on Management of data* Seattle, Washington, United States: ACM, 1998.
[66] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An Efficient Access Method for Similarity Search in Metric Spaces," 1997, pp. 426-435.
[67] C. Traina, A. Traina, B. Seeger, and C. Faloutsos, "Slim-trees: High performance metric trees minimizing overlap between nodes," in *In 7th International Conference on Extending Database Technology (EDBT*: Springer-Verlag, 2000, pp. 51-65.
[68] D. R. Heisterkamp and J. Peng, "Kernel VA-files for relevance feedback retrieva," in *Proceedings of the 1st ACM international workshop on Multimedia databases* New Orleans, LA, USA: ACM, 2003.
[69] E. Açar, "Efficient Index Structures for Video Databases " *Master thesis, Computer Engineering Department, METU, Ankara, Turkey,* January 2008.
[70] M. Bober, "MPEG-7 visual shape descriptors," in *IEEE Trans. Circuits Systems*, 2001.
[71] MPEG, "Overview of mpeg-7 part-6 reference sw: Experimentation model (xm). ISG ISO/IEC JTC1/SC29/WG11 N7544, International Organisation For Standardisation," http://mpeg.chiariglione.org/technologies/mpeg-7/mp07-rsw/index.htm, Last date accessed: April, 2010.
[72] "IBM MPEG-7 Annotation Tool," http://www.research.ibm.com/VideoAnnEx, Last date accessed: June, 2010.

[73] "Java Web Start (JavaWS)," http://en.wikipedia.org/wiki/Java_Web_Start, Last date accessed: July, 2010.
[74] "Java Web Start Witepaper ": http://java.sun.com/javase/technologies/desktop/webstart/, Last date accessed: July, 2010.
[75] "Genetic Algorithm ": http://en.wikipedia.org/wiki/Genetic_algorithm, Last date accessed: April, 2010.
[76] "Ontology," http://en.wikipedia.org/wiki/Ontology, Last date accessed: April, 2010.
[77] M. R. Genesereth and N. J. Nilsson, *Logical foundations of artificial intelligence*: Morgan Kaufmann Publishers Inc., 1987.
[78] M. M. Gupta, "Book review: Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems (Selected Papers by Professor Lotfi A. Zadeh)," *J. Intell. Fuzzy Syst.*, vol. 6, pp. 497-498, 1998.
[79] "Fuzzy Logic," http://en.wikipedia.org/wiki/Fuzzy_logic, Last date accessed: January, 2010.
[80] "Object Database ": http://en.wikipedia.org/wiki/Object_database, Last date accessed: April, 2010.
[81] "Db4o," http://www.db4o.com/about/productinformation/db4o, Last date accessed: March, 2010.
[82] "Java Expert Shell System (JESS)," http://www.jessrules.com, Last date accessed: January, 2010.
[83] "C Language Integrated Production System (CLIPS)," http://en.wikipedia.org/wiki/CLIPS, Last date accessed: January, 2010.
[84] C. L. Forgy, "Rete: a fast algorithm for the many pattern/many object pattern match problem," in *Expert systems: a software methodology for modern applications*: IEEE Computer Society Press, 1990, pp. 324-341.
[85] "Web Services," http://www.w3.org/TR/ws-arch/, Last date accessed: April, 2010.
[86] L. Lu, H. Jiang, and H. Zhang, "A robust audio classification and segmentation method," *ACM Multimedia*, pp. 203-211, 2001.
[87] E. Doğan, M. Sert, and A. Yazıcı, "Content-based classification and segmentation of mixed-type audio by using mpeg-7 features," *Int. Conf. on Advances in Multimedia (MMEDIA), IEEE*, July 2009.
[88] T. Zhang and C. Jay Kuo, "Hierarchical system for content-based audio classification and retrieval," *Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 398-409.
[89] R. George, B. P. Buckles, and F. E. Petry, "Modelling class hierarchies in the fuzzy object-oriented data model," *Fuzzy Sets Syst.*, vol. 60, pp. 259-272, 1993.
[90] J. Shi and J. Malik, "Normalized Cut and Image Segmentation," University of California at Berkeley1997.
[91] "Oracle Berkeley DB," http://www.oracle.com/technology/products/berkeley-db/index.html, Last date accessed: April, 2010.