A Fuzzy Conceptual Model for Multimedia Data with Application to News Video Domain

Dilek Kucuk, N. Burcu Ozgur, Adnan Yazici
Dept. of Computer Engineering
METU, Ankara - Turkey
{dkucuk, burcu.ozgur, yazici}@ceng.metu.edu.tr

Murat Koyuncu
Dept. of Computer Engineering
Atilim University, Ankara - Turkey
mkoyuncu@atilim.edu.tr

Abstract

The size of multimedia data is increasing fast due to the abundance of multimedia applications. Modeling the semantics of the data effectively is crucial for proper management of it. In this paper, we present a fuzzy conceptual data model for multimedia data which is also generic in the sense that it can be adapted to all multimedia domains. The model takes an object-oriented approach and it handles fuzziness at different representation levels where fuzziness is inherent in multimedia applications and should be properly modeled. The proposed model also has the nice feature of representing the structural hierarchy of multimedia data as well as the spatial and temporal relations of the data. The model is applied to the news video domain and implemented as a fuzzy multimedia database system where it turns out to be effective in representing this domain and thereby provides an evidence for the general applicability of the model.

Keywords: Conceptual design, fuzzy multimedia modeling, multimedia databases, news videos.

1 Introduction

Multimedia applications become increasingly common which leads to huge sizes of multimedia data to be managed. In order to manage multimedia data in an effective way, it is imperative to model the semantics of it in the most appropriate way before storing it in a database. When modeling multimedia data, two important points should be paid particular attention: first, multimedia data is more complex than most other types of data that are stored in conventional relational database systems, modeling it by taking an object-oriented approach seems a more plausible alternative compared to the relational modeling approach. The second point is the inherently fuzzy nature of multimedia data for semantics of it may not always be expressed in a precise way; furthermore, users of this data may demand information by providing flexible queries. A generic model for multimedia data should provide a way to handle fuzziness as well as possessing object-oriented characteristics.

Conceptual model of multimedia data is an abstract representation of its semantic contents by identifying complex semantic entities and relationships among these entities. In most of the existing proposals of conceptual models for video data, its hierarchical structure is also represented in addition to the semantics of it. In one of such studies [5], the authors employ the commonly discussed *sequence-scene-shot hierarchy* where a *shot* is defined as a contiguous sequence of frames representing a continuous action in time and space; *scenes* are constructed by shots which are related in time and space, and finally, semantically closer scenes are combined to construct a *sequence* which describes a continuing story.

Entity-Relationship Diagrams (ERD), Extended Entity-Relationship Diagrams (EERD), and Unified Modeling Language (UML) [2] are three commonly used tools in conceptual data modeling. There are some studies [9, 11, 15, 17] that further extend these tools with some new fuzzy constructs to represent uncertain information. There exist several studies on fuzzy data models including [8, 12, 13] and a recent survey of these models is presented in [10]. Yet, to our best knowledge, there are only few studies [1, 3] on fuzzy conceptual modeling of multimedia data. In the study [1], the authors claim that fuzziness could be represented at

L. Magdalena, M. Ojeda-Aciego, J.L. Verdegay (eds): Proceedings of IPMU'08, pp. 1741–1748
Torremolinos (Málaga), June 22–27, 2008

three levels in an object-oriented model: attribute level, object/class level, and class/superclass level.

In [17], three different types of fuzziness are addressed at the attribute level. The first type is called *incompleteness* and corresponds to the case when the value of an attribute is specified as a range value. The second one which is called *null type* is used to represent the cases when the value of an attribute is not known, does not exist, or there is no information on whether a value exists or not. The last type, which is called *fuzzy type*, corresponds to the cases when the value of an attribute is vaguely specified. Some new uncertain data types for each of these types of fuzziness are defined in [18]: *UT_fy* represents fuzzy data type, *UT_nu* represents null type, and *UT_in* represents incomplete data type. Fuzziness at the object/class level is used to represent imprecise information due to the existence of a partial membership of an object to its class. If the boundary of a class is uncertain, then the objects of this class may be a member of the class with a membership degree between 0 and 1. The relevance of an attribute to an object can also be used to represent imprecise information at the object/class level. Class/subclass level fuzziness refers to the existence of a partial membership of a class to its superclass(es). This type of uncertainty indicates that the fuzziness occurs at the class inheritance hierarchy since we might not be able to construct a class hierarchy precisely. For instance, in the biology domain, the class *Virus* will be a subclass of the more general class *LivingThing* with a membership degree of about 0.5 since viruses do not demonstrate all the characteristics of living things.

In this paper, we propose a generic fuzzy conceptual data model for multimedia data. We employ an object oriented approach and use extended UML [1, 17] with fuzzy constructs to represent this model. The proposed model also represents the hierarchical structure of multimedia data as described previously with the spatial and temporal relations of the data as well. The model is generic in the sense that it can be easily applied to any domain such as news and sports. We applied the model to news video domain and implemented it as a fuzzy multimedia database system to see its capabilities on a specific domain. To our best knowledge, this is the first fuzzy multimedia data model with a real application on news videos to observe its effectiveness on real data.

The rest of the paper is organized as follows: In Section 2, details of the proposed fuzzy model are presented. Section 3 is devoted to the application of the model to news video domain. The implementation of the system with sample user queries is described in Section 4. Conclusions and further directions to pursue are presented in Section 5.

2 A Fuzzy Conceptual Data Model for Multimedia Data

Considering the hierarchical and semantic characteristics of video data, we arrive at the abstract generic video data model presented in Figure 1.

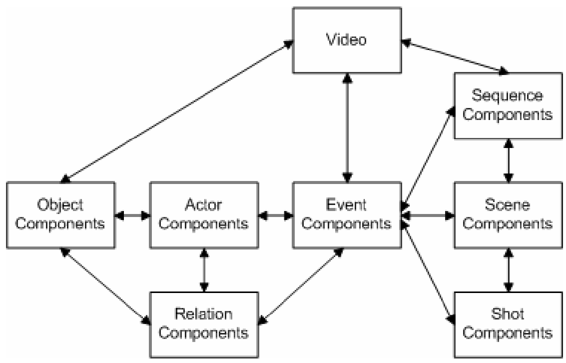

Figure 1: Abstract Representation of the Video Data Model

Among the components in the representation; *Sequence*, *Scene*, and *Shot* model the hierarchical components of the video which will be acquired as a result of a proper segmentation process. The remaining components, namely, *Object*, *Actor*, *Event*, and *Relation Components* are proposed to model the semantic aspects of the video such as semantic entities, roles, actions, and relations.

Our fuzzy conceptual model is presented in Figure 2 as a UML class diagram based on the abstract model in Figure 1. The attributes and methods of the classes are described in the upcoming paragraphs with detailed class diagrams.

In Figure 3, the detailed class diagram with the classes *Fuzzy*, *Video*, *Event*, *Object*, and *Actor* are presented where the relevant attributes and methods of these classes are also provided. In this figure, the class *Fuzzy* contains only two attributes *objectMembershipDegree* and *classMembershipDegree* where each attribute should have a floating-point value between 0 and 1.0, inclusively. The former attribute is used to hold the membership value of an object to a class, so that this model can handle fuzziness in the object/class level. The latter one is for the membership value of a class to its superclass in order to represent fuzziness in the class/superclass level. Examples of attribute level fuzziness representation could also be observed in Figure 3 as the *description* attributes of the *Video* and *Object* classes, and *how* attribute of *Event* class are of type *UT_nu*, and *where* and *when* attributes of the *Event* class are of *UT_fy* type. As specified before, in order to represent fuzziness in the object/class level, the *relevance* of the attributes of a class to an object could be specified. In our model, we use *setRelevances* method to set floating point values ranging from 0 to 1.0 as the relevance of each attribute of a class to its object instances. For this purpose, in Figure 3, the *Fuzzy* class has a *setRelevances* method which is inherited by the subclasses *Video*, *Event*, *Object*, and *Image*.

Since objects may be involved in several events and have different roles in each event, it is a better approach to store event-specific information in an entity other than object entity which leads to the *Actor* entity idea introduced in [4]. Among the attributes of the *Actor* class, *semanticRole* holds the semantic role of an actor in an event such as *speaker*, *walker*… The *linguisticRole* attribute is used to hold the linguistic role of an *Actor* in an event where the value of this attribute may be *agent*, *object*, or *recipient* [7]. Agent is the entity performing the activity in an event, the object is the one directly affected from the event, and lastly recipient is the indirectly affected entity.

The detailed class diagram of the classes *Structure*, *Sequence*, *Scene*, and *Shot* of our generic model are presented in Figure 4. These classes serve to represent the hierarchical structure of multimedia data as described in the previous section. Each of the classes *Sequence*, *Scene*, and *Shot* inherit from the generic class *Structure* and there is a hierarchy between the former three classes. That is, each *Sequence* consists of one or more *Scene*s and similarly each *Scene* comprises one or more *Shot*s.

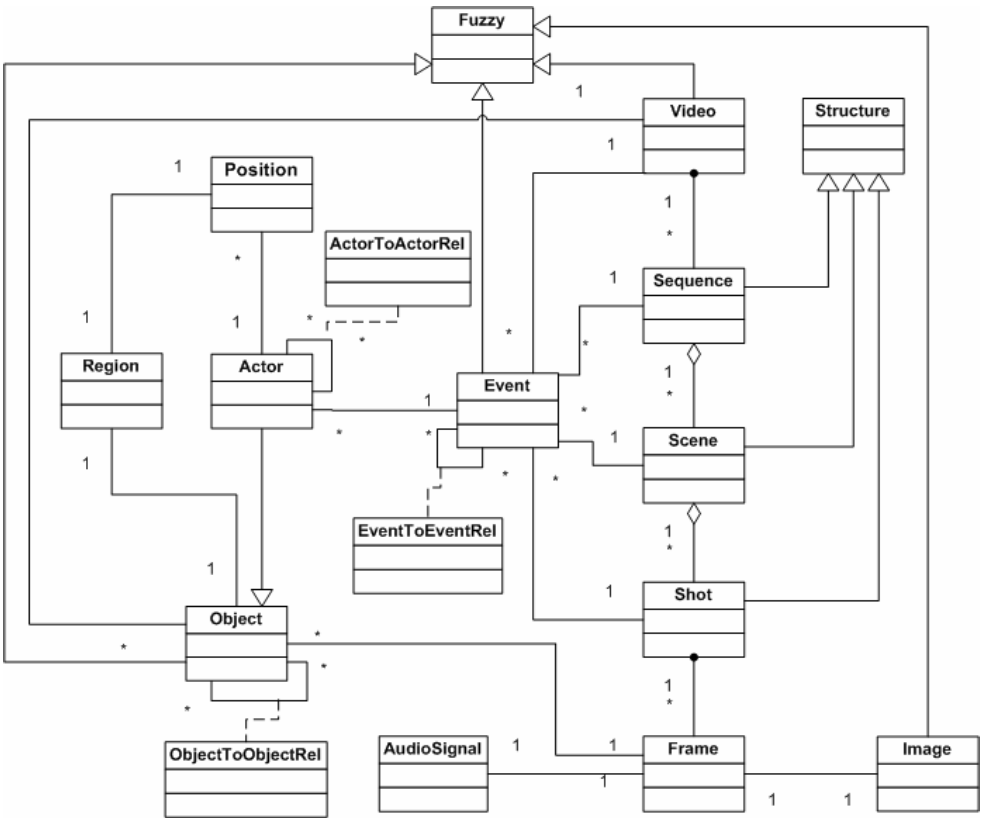

Figure 2: Fuzzy Conceptual Data Model for Multimedia Data Represented as a UML Class Diagram

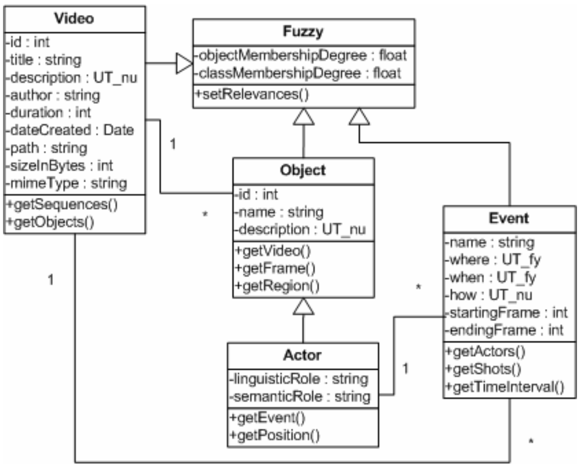

Figure 3: Class Diagram of *Fuzzy*, *Video*, *Event*, *Object*, and *Actor*

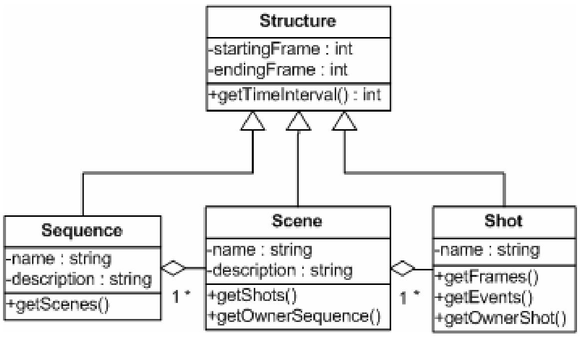

Figure 4: Class Diagram of *Structure*, *Sequence*, *Scene*, and *Shot*

In Figure 5, the class diagram showing the details of the classes *Shot, Frame, AudioSignal*, and *Image* is presented. In this diagram, the construct combining the *Shot* and *Frame* classes is the newly proposed *sequence* relationship incorporated into standard UML in the study [1]. The *sequence* construct is a special case of aggregation where the constituents have a chronological order. In the figure, the construct conveys the meaning that a *Shot* instance is composed of one or more *Frame* instances but with the constraint that these *Frame* instances should be ordered. The *sequence* relationship also exists between *Video* and *Sequence* classes as it is shown in Figure 2 where it denotes that a *Video* instance comprises one or more ordered *Sequence* instances.

The classes *Actor*, *Position*, and *Region* are represented with their most relevant attributes and methods as a class diagram in Figure 6. According to this diagram, an *Actor* instance has one or more *Position* instances and each of these *Position* instances has a *Region* instance.

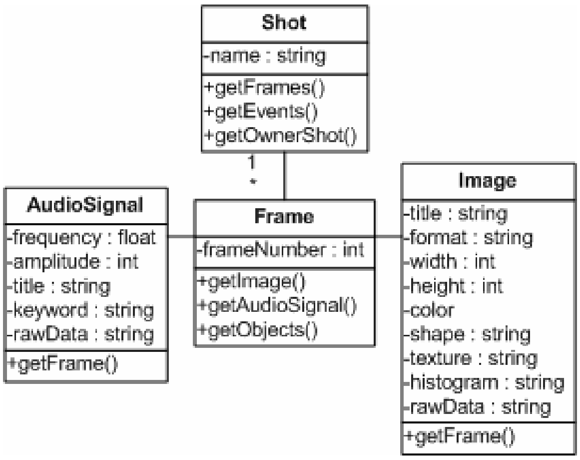

Figure 5: Class Diagram of *Shot*, *Frame*, *AudioSignal*, and *Image*

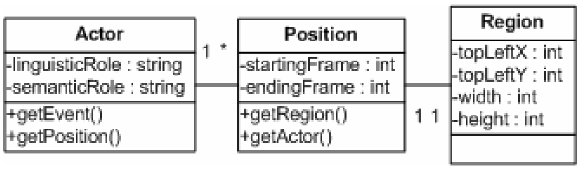

Figure 6: Class Diagram of *Actor*, *Position*, and *Region*

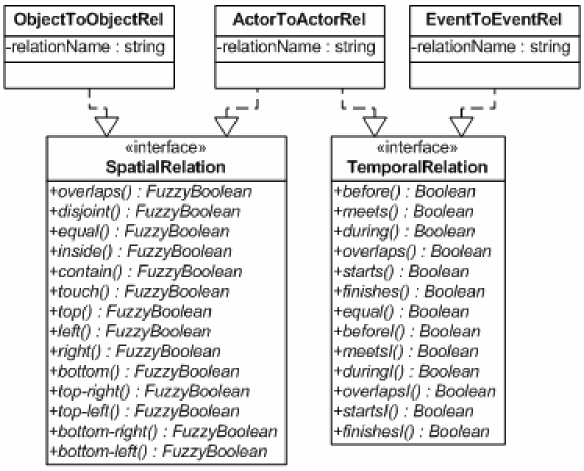

Figure 7: Relation Classes and Interfaces

In Figure 7, detailed class diagram of the relation classes is presented. *ObjectToObjectRel* is an association class between two *Object* classes and implements *SpatialRelation* interface which has several methods related to spatiality. Similarly, *EventToEventRel* is an association class between two *Event* classes and implements *TemporalRelation* interface which has several methods related to temporality. The last association class, *ActorToActorRel*, is between two *Actor* classes and implements *SpatialRelation* and *TemporalRelation* interfaces. The interfaces *SpatialRelation* and

TemporalRelation are similar to *SpatialObject* and *TemporalObject* classes of VIDEX model presented in [16].

In the *SpatialRelation* interface, return values of all the methods are of type *FuzzyBoolean* which is a class with a single attribute (called *membershipValue*) of floating-point type ranging from 0 to 1.0. *FuzzyBoolean* is the class version of the *Fuzzyboolean* literal introduced in [14].

3 Application of the Model to News Video Domain

In this section, we present an application of our generic model to the news video domain. In order to model this domain, we introduce some new classes extended from the classes in the generic model. To differentiate between the classes in the original generic model and the newly introduced ones, the latter type of classes are painted in grey.

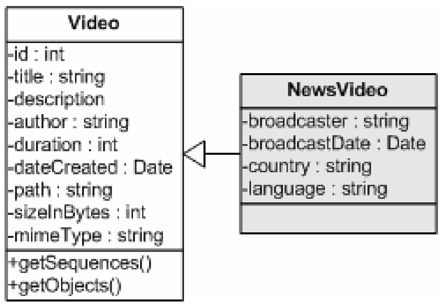

Figure 8: The *NewsVideo* Class

Our first class is *NewsVideo* extended from the *Video* class as presented in Figure 8. It has four attributes specific to the video instances in the news domain: *broadcaster, broadcastDate, country* and *language*. As their names imply, *broadcaster* is used to hold the station which broadcasts the video like *Channel 1 etc.*, *broadcastDate* holds the date when the video is actually broadcasted like *03-05-2007 07:30 PM*, *country* holds the name of the country where broadcasting station resides and finally the *language* attribute holds the language used in the video.

In order to hold information specific to news domain at the sequence level, we propose the classes in Figure 9. Three different *Sequence* subclasses exist for different purposes: *NewsIntroSequence*, *NewsSequence,* and *NewsEndingSequence*. The first class is aimed to model the sequence of a news video from the beginning of the news video up to the first actual news item. Each individual news items are modeled with the *NewsSequence* class and finally, the *NewsEndingSequence* class is used to model the portion of the news video beginning from the end of the last news item (which is modeled with a *NewsSequence* instance) to the end of the whole news video.

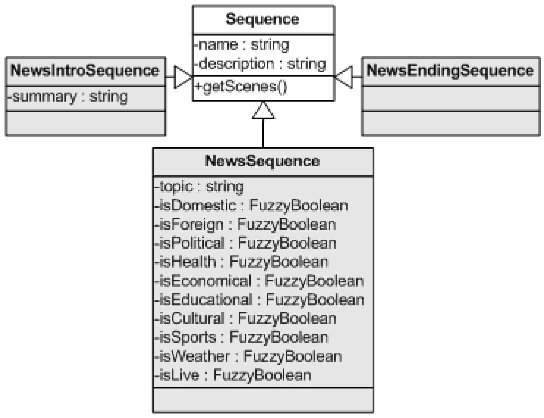

Figure 9: Three *Sequence* Subclasses

Apart from the *topic* attribute, all attributes of the *NewsSequence* class are of type *FuzzyBoolean*. For instance, if a news sequence has live scenes about half of its whole duration, then the value of its *isLive* attribute is a *FuzzyBoolean* instance with a *membershipValue* of 0.5. These attributes are also examples of handling fuzziness at the attribute level.

We do not introduce any further classes to apply our generic model to news video domain, and in the rest of this section, we provide instances of these classes to model particular news videos.

A *NewsSequence* instance called *News-Sequence1* is provided in Figure 10, with its scenes as well. *NewsSequence1* represents a news item on the talk of the Minister of Education of Turkish government on a change of a law in the parliament. This sequence is undoubtedly political and domestic, therefore relevant attributes have values with *membershipDegree*s of 1.0. The sequence is also moderately educational; and therefore its *isEducational* attribute has a membershipDegree of 0.6. The sequence is composed of three scenes, *AnchormanScene*, *ReporterScene*, and *ParliamentScene*. The starting and ending frames, names and descriptions are provided as the values of the relevant attributes for each of these scenes.

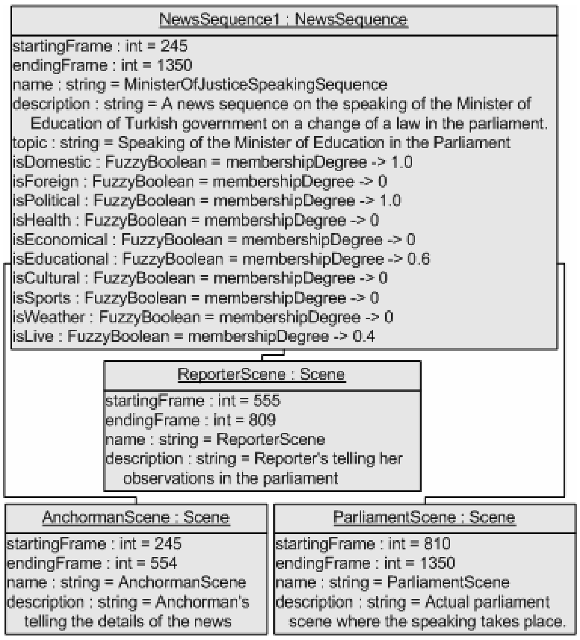

Figure 10: An Instance of the *NewsSequence* Class with Its *Scene* Instances

Figure 11 demonstrates the *ParliamentScene* of *NewsSequence1* with its shots. This scene is composed of three shots, namely, *SpeakerShot1*, *AudienceShot*, and *SpeakerShot2*.

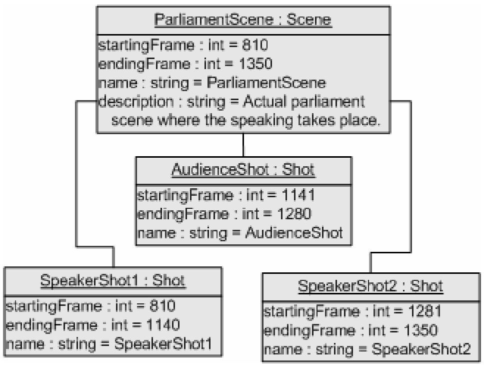

Figure 11: The *ParliamentScene* Instance of the *Scene* Class and Its *Shot* Instances

Finally, *SpeakerShot1* instance is presented in Figure 12 with its events, *SpeakingEvent1* and *DrinkingEvent1*, and actors in these events, namely, *Speaker1, Drinker1* and *Drinkee1*.

In Figure 12, *DrinkingEvent* and *SpeakingEvent* are subclasses of the *Event* class, each with a *classMembershipDegree* of 1.0, i.e., they are crisp subclasses. Yet, SpeakingEvent1 is an instance of *SpeakingEvent* with an *objectMembershipDegree* of 0.9 and DrinkingEvent1 is an instance of *DrinkingEvent* with an *objectMembershipDegree* of 0.6. Hence, these two instances demonstrate fuzziness at the object/class level.

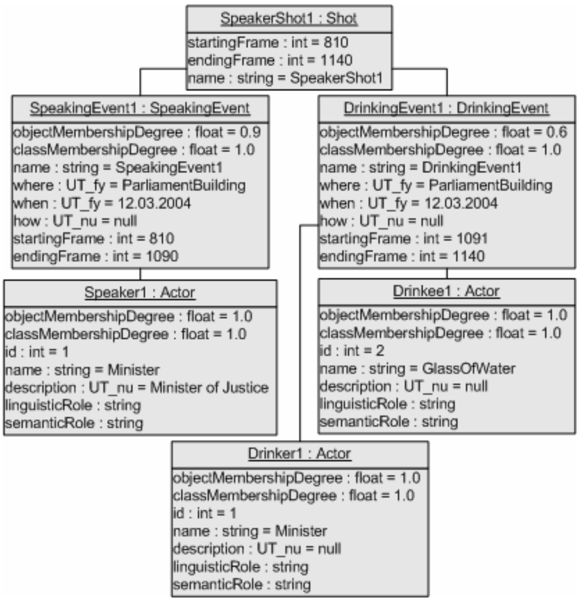

Figure 12: The *SpeakerShot1* Instance of the *Shot* Class

4 System Implementation

The conceptual data model presented in Section 2 and 3 is implemented as a fuzzy multimedia database system. The implementation is carried out with Java programming language using DB4O [6] as the underlying object oriented database. The general architecture of the implemented system is presented in Figure 13.

In the implemented architecture, the fuzzy object oriented database holds the fuzzy objects extracted from videos, such as objects and events. DB4O is extended with a fuzzy processor to handle uncertain data. Java servlet technology is used to query the system on the Internet. The web application comprising the required Java servlets is deployed on Apache’s Tomcat web server. The users can query the system using their browsers through Java applets or by the use of Java Web Start (JWS) which enables the latest version of the user interface to be deployed to the local system so that it can be run as a local Java application immediately. In either case, the users need to employ Java Media Framework (JMF) to display the videos, video sequences, scenes, or shots that are retrieved as the results of the user queries. Annotation of the videos is done manually except segmentation of the videos into shots. However, a local application which eases the annotation task is developed and used to insert data into the database.

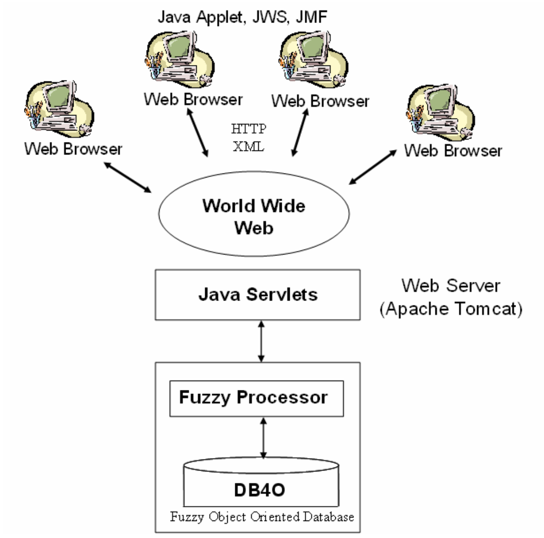

Figure 13: System Architecture

The users can supply crisp as well as fuzzy queries to the implemented system. We conclude this section with a demonstration of three sample semantic queries and their respective results to further clarify the execution of the system. Our first query, *Q1*, is a semantic and crisp query over all videos in the fuzzy database.

Q1: Retrieve the titles of all news videos where the singer Tarkan appears.

This query, when provided through the user interface, is translated into an Object Query Language (OQL)-like query language and then executed to retrieve the matching video titles.

select *V.title* ***from*** *Video V, Actor A*
where *V.id = A.getVideo()* ***and***
A.name = 'Tarkan' ***and***
A.semanticRole = 'singer';

Before presenting the remaining two fuzzy queries, mapping of the fuzzy terms used in the queries to the corresponding membership degrees is presented in Table 1. Eight example *NewsSequence* instances are included in Table 2 with only their relevant attributes to the queries. With the objects in Table 2, the execution of two fuzzy user queries, *Q2*, and *Q3*, is illustrated in the following paragraphs.

Q2: Retrieve all news sequences that are somewhat domestic.

The expression corresponding to this fuzzy query is presented below:

select *NS* ***from*** *NewsSequence NS*
where *NS.isDomestic > 0;*

Among the *NewsSequence* objects in Table 2, this query returns all objects except those with object ids of 2 and 7 since the membership degrees of their *isDomestic* attributes are both 0 and thereby they do not satisfy this query.

Table 1: Mapping of the Fuzzy Terms to Their Corresponding Membership Degrees

Fuzzy Term	Membership Degree
Exactly/totally	1.0
Mostly	> 0.7
Moderately	> 0.4
Somewhat	> 0

Table 2: Eight Example *NewsSequence* Objects in the Fuzzy Object Oriented Database

Object ID	isDomestic	isHealth	isEconomical
1	1.0	0	0.7
2	0	1.0	0.3
3	0.4	0	0
4	1.0	0.2	0
5	1.0	0.5	0.8
6	0.6	0	0.9
7	0	1.0	0.5
8	1.0	0.8	0.9

The next fuzzy query is provided below with its corresponding query expression in the OQL-like language:

Q3: Retrieve all news sequences that are mostly economical, moderately health-related, and totally domestic.

select *NS* ***from*** *NewsSequence NS*
where *NS.isEconomical > 0.7* ***and***
NS.isHealth > 0.4 ***and***
NS.isDomestic= 1.0;

When *Q3* is run over the *NewsSequence* objects in Table 2, it returns the objects with object ids of 5 and 8 since they are the only objects that satisfy the requirements of the query *Q3*.

5 Conclusion

In this paper, we present a fuzzy conceptual data model for multimedia data and its application to news video domain. The proposed model is also generic in the sense that it could easily be adapted to any multimedia domain. It takes an object oriented approach with the ability to handle fuzziness at the attribute, object/class and class/superclass levels. In addition to representing the semantics of the data, the model also handles its hierarchical structure and the spatial and temporal relations among the data. In order to apply the model to the domain of news videos, we define several new classes inheriting from the classes in the generic model to add domain-specific attributes. In order to see the effectiveness of the model on news videos, we implement the proposed model as a fuzzy multimedia database system for news videos and present several crisp as well as fuzzy semantic queries that the system handles. Our application of this generic conceptual model to news domain is an evidence for its generality and ease of applicability.

As further studies, we are planning to incorporate a fuzzy knowledge base to the implemented system to handle a wider range of queries to the fuzzy database, including spatial, temporal, and trajectory queries.

Acknowledgements

This work is supported in part by a research grant from TUBITAK EEEAG 106E012.

We owe special thanks to the following members of our research project for their contributions: Hakan Oztarak, Umit L. Altintakan, Utku Demir, Muhsin Civelek, Ozan Kusmen, Neriman Atar, and Gozde B. Akar.

References

[1] R. S. Aygun and A. Yazici (2004). Modeling and Management of Fuzzy Information in Multimedia Database Applications. *Multimedia Tools and Applications*, Vol. 24, No. 1, pp. 29–56.

[2] G. Booch, J. Rumbaugh, and I. Jacobson (1999). The Unified Modeling Language User Guide. Addison-Wesley, Reading MA.

[3] G. de Tre and R. de Caluwe (2003). Modelling Uncertainty in Multimedia Database Systems: An Extended Possibilistic Approach. *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 11, No. 1, pp. 5–22.

[4] A. Ekin, A. M. Tekalp, and R. Mehrotra (2004). Integrated Semantic–Syntactic Video Modeling for Search and Browsing. *IEEE Transactions on Multimedia*, Vol. 6, No. 6, December 2004.

[5] R. Hjelsvold and R. Midtstraum (1994). Modelling and Querying Video Data. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 686–694.

[6] http://www.db4o.com/

[7] ISO/IEC Committee Draft 15 938-5 Information Technology - Multimedia Content Description Interface: Multimedia Description Schemes, ISO/IEC/JTC1/SC29/WG11/N3966, Mar. 2001.

[8] M. Koyuncu and A. Yazici (2003). IFOOD: An Intelligent Fuzzy Object-Oriented Database Architecture. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 5, 2003.

[9] Z. M. Ma (2004). Fuzzy Information Modeling with the UML. In *Advances in Fuzzy Object-Oriented Databases: Modeling and Applications*, IDEA Group Publishing.

[10] Z. M. Ma and L. Yan (2008). A Literature Overview of Fuzzy Database Models. *International Journal of Information Science and Engineering*, Vol. 24, No. 1, pp. 189–202.

[11] Z. M. Ma, W. J. Zhang, W. Y. Ma, and Q. Chen (2001). Conceptual Design of Fuzzy Object-Oriented Databases Using Extended Entity-Relationship Model. *Int. Journal of Intelligent Systems*, Vol. 16, pp. 697–711.

[12] N. Marin, O. Pons, and M. A. Vila (2001). A Strategy for Adding Fuzzy Types to an Object-Oriented Database System. *Int. Journal of Intelligent Systems*, Vol. 16, No. 7, pp. 863–880.

[13] J. M. Medina, O. Pons, and M. A. Vila (1994). GEFRED: A Generalized Model of Fuzzy Relational Databases. *Information Sciences*, Vol. 76, No. 1–2.

[14] S. Nepal, M. V. Ramakrishna, and J. A. Thom (1999). A Fuzzy Object Query Language (FOQL) for Image Database. In *Proceedings of Sixth International Conference on Database Systems for Advanced Applications*, pp. 117–124.

[15] M. A. Sicilia, E. Garcia, and J. A. Gutierrez (2004). Introducing Fuzziness in Existing Orthogonal Persistence Interfaces and Systems. In *Advances in Fuzzy Object-Oriented Databases: Modeling and Applications*, IDEA Group Publishing.

[16] R. Tusch, H. Kosch, and L. Böszörményi (2000). VIDEX: An Integrated Generic Video Indexing Approach. In *Proceedings of the 8th ACM Multimedia Conference*, Los Angeles (USA), ACM Press, pp. 448–451.

[17] A. Yazici, Q. Zhu, and N. Sun (2001). Semantic Data Modeling of Spatiotemporal Database Applications. *Int. Journal of Intelligent Systems*, Vol. 16, No. 7, pp. 881–904.

[18] A. Yazici, R. George, and D. Aksoy (1998). Design and Implementation Issues in the Fuzzy Object-Oriented Data Model. *Information Sciences (Int. Journal)*, Vol. 108, No. 1–4, pp. 241–260, July 1998.