International Journal of Uncertainty,
Fuzziness and Knowledge-Based Systems
Vol. 17, Suppl. 1 (August 2009) 135–152

A FUZZY CONCEPTUAL MODEL FOR MULTIMEDIA DATA WITH A TEXT-BASED AUTOMATIC ANNOTATION SCHEME

DİLEK KÜÇÜK*, N. BURCUÖZGÜR† and ADNAN YAZICI‡

Department of Computer Engineering, Middle East Technical University,
Ankara, 06531, Turkey
** dkucuk@ceng.metu.edu.tr*
† burcu.ozgur@ceng.metu.edu.tr
‡ yazici@ceng.metu.edu.tr

MURAT KOYUNCU

Department of Computer Engineering, Atılım University,
Ankara, 06836, Turkey
mkoyuncu@atilim.edu.tr

Received 3 November 2008
Revised 26 January 2009
Accepted 23 April 2009

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 the domain and thereby provides an evidence for the general applicability of the model. The model is accompanied by an automatic multimedia annotation scheme which makes use of information extraction techniques on the corresponding multimedia texts.

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

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, 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.[1]

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 that further extend these tools with some new fuzzy constructs to represent uncertain information.[3–6] There exist several studies on fuzzy data models including Refs. 7–9 and a recent survey of these models is presented in Ref. 10. Yet, to our best knowledge, there are only few studies as presented in Refs. 11 and 12 on fuzzy conceptual modeling of multimedia data. In Ref. 12, the authors claim that fuzziness could be represented at three levels in an object-oriented model: attribute level, object/class level, and class/superclass level. Three different types of fuzziness are addressed at the attribute level in Ref. 4. 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 Ref. 13: *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 along with a text-based automatic annotation scheme.[a] We employ an object oriented approach and use UML extended with fuzzy constructs presented in Refs. 4 and 12 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 natural language texts corresponding to multimedia data, which could be caption texts as well as transcription texts obtained through automatic speech recognition (ASR) techniques, are valuable sources of information for semantic modeling of the data. Hence, in order to aid in the annotation of the data in the implemented system, we also present an automatic annotation scheme which employs information extraction techniques to deduce semantic information from the corresponding multimedia texts.

The rest of the paper is organized as follows: In Sec. 2, details of the proposed fuzzy conceptual model are presented. Section 3 is devoted to the application of the model to the domain of generic news videos. The system architecture corresponding to the implementation of the proposed model as a fuzzy multimedia database system is described in Sec. 4 with a set of sample user queries. A text-based semantic annotation scheme which has been incorporated into the implemented system architecture is presented in Sec. 5. Finally, Sec. 6 includes concluding remarks and future research directions to pursue.

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 Fig. 1.

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 Fig. 2 as a UML class diagram based on the abstract model in Fig. 1. The attributes and methods of the classes are described in the upcoming paragraphs with detailed class diagrams.

[a]Preliminary version of the study has been presented in Ref. 14.

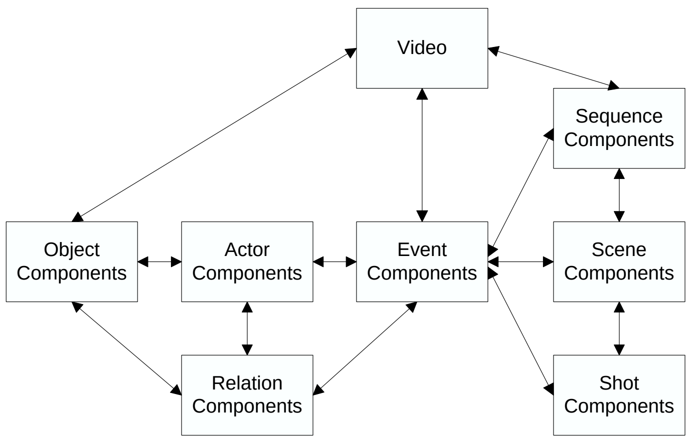

Fig. 1. Abstract representation of the video data model.

In Fig. 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 Fig. 3 as the *description* attributes of the *Video* and *Object* classes, and the *how* attribute of the *Event* class are of type *UT_nu*, and the *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 Fig. 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 Ref. 15. Among the attributes of the Actor class, *semanticRole* holds the semantic role of an actor in an event such as *speaker*, *walker*, etc. 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*.[16] 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.

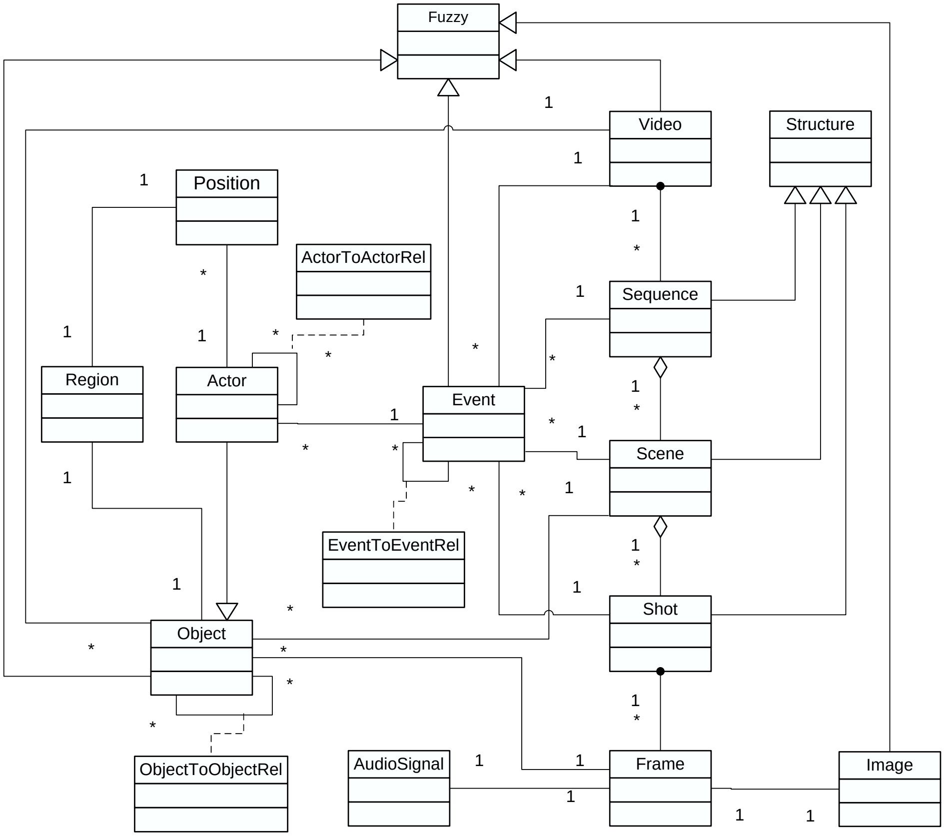

Fig. 2. Fuzzy conceptual data model for multimedia data represented as a UML class diagram.

The detailed class diagram of the classes *Structure*, *Sequence*, *Scene*, and *Shot* of our generic model are presented in Fig. 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 *Scenes* and similarly each *Scene* comprises one or more *Shots*.

In Fig. 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 Ref. 12. 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

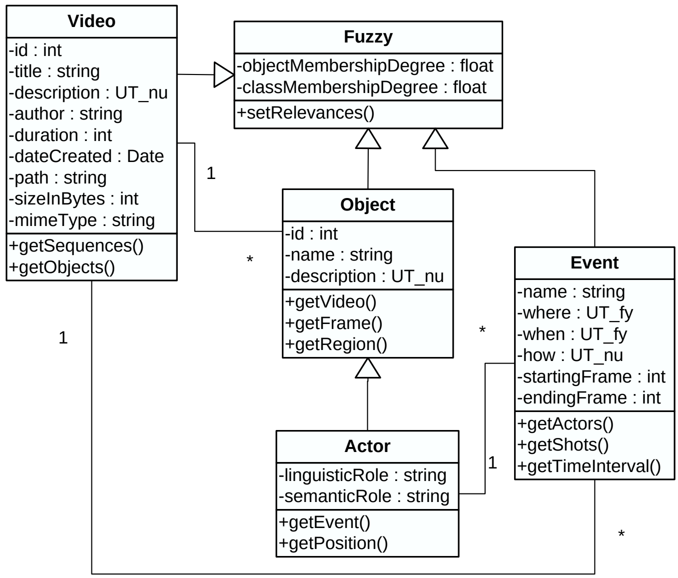

Fig. 3. Class diagram of *Fuzzy*, *Video*, *Event*, *Object*, and *Actor*.

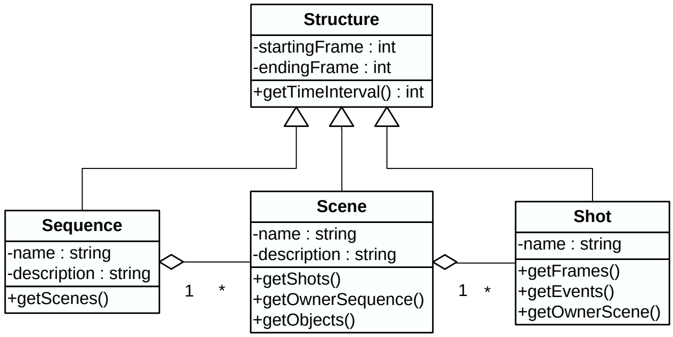

Fig. 4. Class diagram of *Structure*, *Sequence*, *Scene*, and *Shot*.

in Fig. 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 Fig. 6. According to this diagram, an *Actor* instance has one or more Position instances and each of these *Position* instances has a *Region* instance.

In Fig. 7, detailed class diagram of the relation classes is presented. *Object-ToObjectRel* is an association class between two *Object* classes and implements *SpatialRelation* interface which has several methods related to spatiality. Similarly,

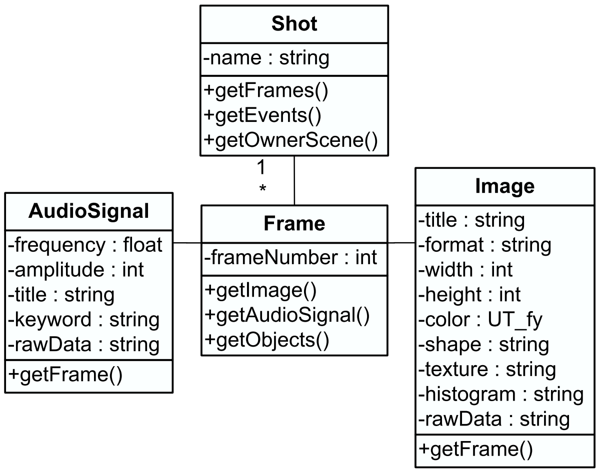

Fig. 5. Class diagram of *Shot*, *Frame*, *AudioSignal*, and *Image*.

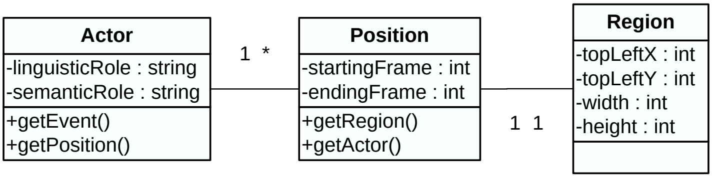

Fig. 6. Class diagram of *Actor*, *Position*, and *Region*.

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 the *SpatialObject* and *TemporalObject* classes of VIDEX model presented in Ref. 17.

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 Ref. 18.

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.

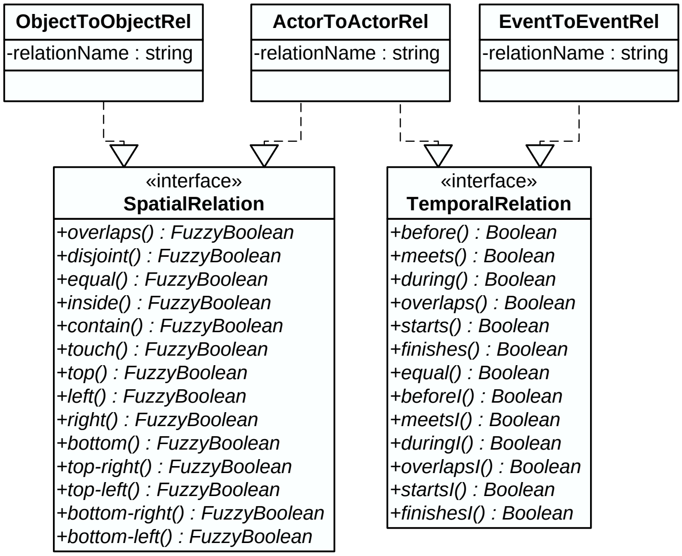

Fig. 7. Relation classes and interfaces.

Our first class is *NewsVideo* extended from the *Video* class as presented in Fig. 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 Fig. 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.

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.

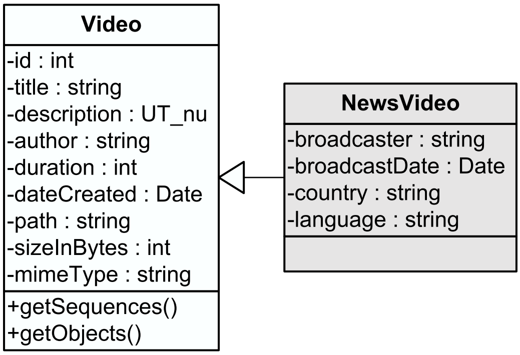

Fig. 8. The *NewsVideo* class.

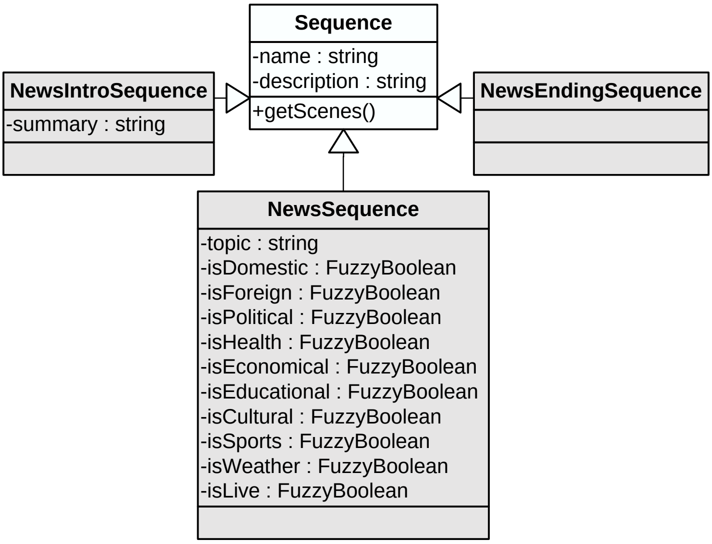

Fig. 9. Three *Sequence* subclasses.

A *NewsSequence* instance called *NewsSequence1* is provided in Fig. 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 *membershipDegrees* of 1.0. The sequence encompasses some live portion of talk of the minister which leads to a *membershipDegree* of 0.4 for the *isLive* attribute of the sequence. The sequence is also moderately educational; and hence its *isEducational* attribute has a *membershipDegree* of 0.6. The sequence is composed of three scenes, *AnchormanScene*, *ReporterScene*, and *ParliamentScene*. The starting

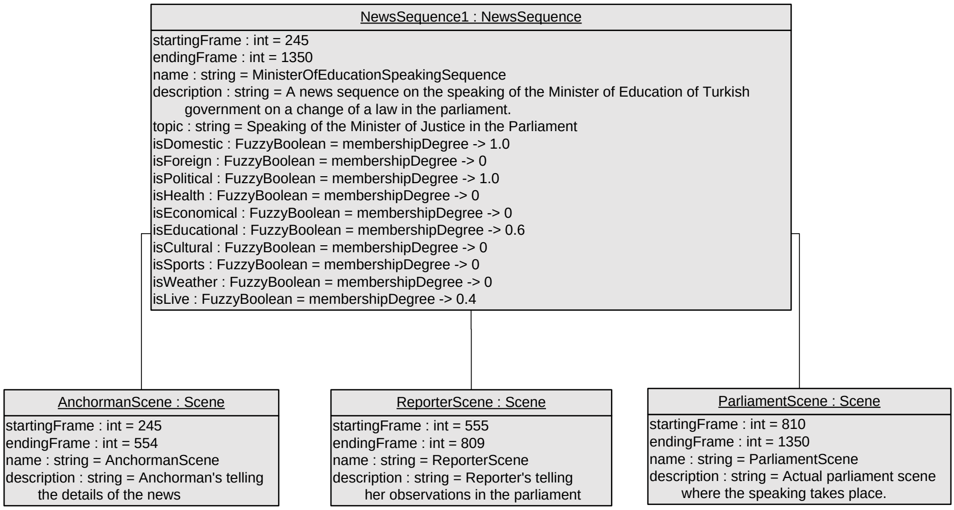

Fig. 10. An instance of the *NewsSequence* class with its *Scene* instances.

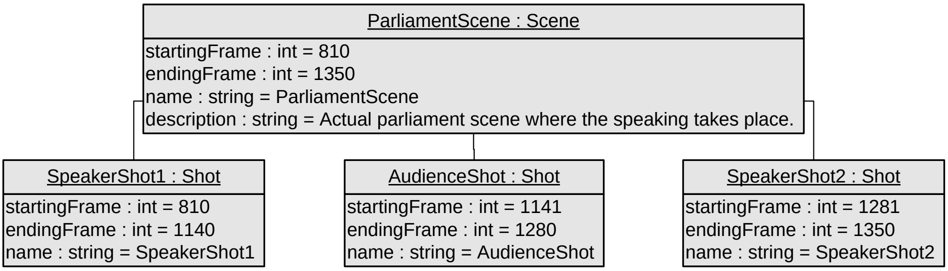

Fig. 11. The *ParliamentScene* instance of the *Scene* class and its *Shot* instances.

and ending frames, names, and descriptions are provided as the values of the relevant attributes for each of these scenes.

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

Finally, *SpeakerShot1* instance is presented in Fig. 12 with its events, *SpeakingEvent1* and *DrinkingEvent1*, and actors in these events, namely, *Speaker1*, *Drinker1* and *Drinkee1*. In this figure, *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. The *objectMembershipDegree* of 0.9 for *SpeakingEvent1* shows the fuzziness of its membership to the *SpeakingEvent* class. That is *SpeakingEvent1* could not be classified exactly as a *SpeakingEvent*, but with

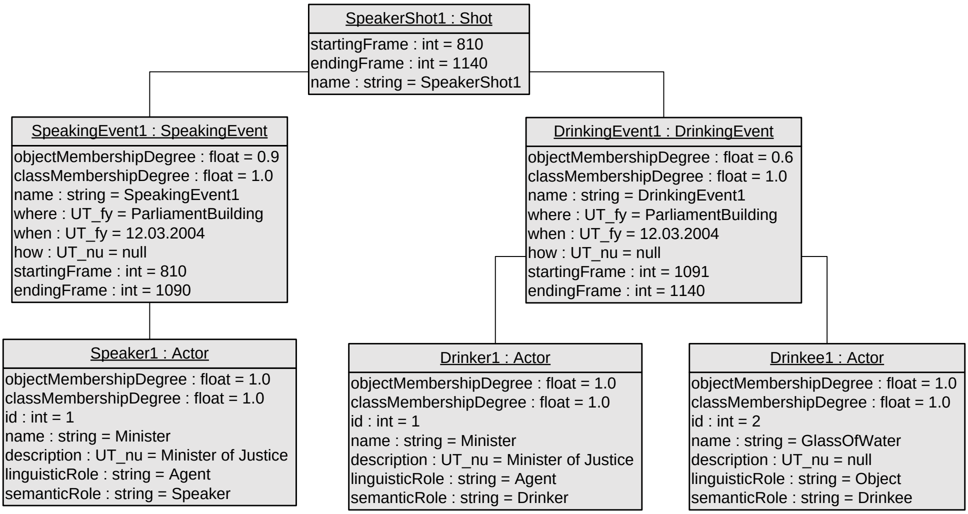

Fig. 12. The *SpeakerShot1* instance of the *Shot* class.

a certain truthness. The same situation holds true for the case of *DrinkingEvent1* instance with a membership degree of 0.6. Hence, these two instances demonstrate fuzziness at the object/class level.

4. System Architecture

The conceptual data model presented in Sec. 2 and 3 is implemented as a fuzzy multimedia database system. The implementation is carried out with Java programming language using DB4O[19] as the underlying object oriented database. The general architecture of the implemented system is presented in Fig. 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 for the segmentation of the videos into shots and the extraction of semantic objects provided that the corresponding video texts are available as will be explained in Sec. 5. However, a local application which eases the manual annotation task is developed and used to insert data into the database.

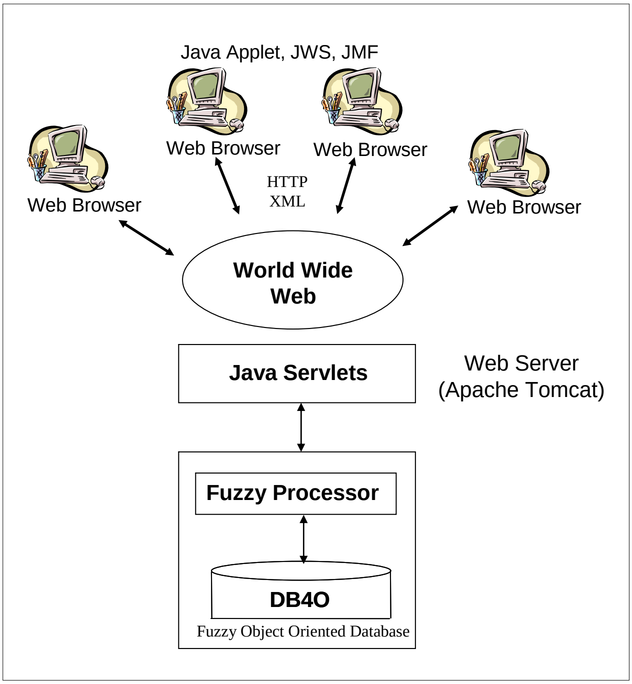

Fig. 13. System architecture.

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 (queries regarding the high level semantic content of the videos as opposed to those queries based on low level features of the videos) 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  NV.title from NewsVideo NV, Actor A
where   NV.id = A.getVideo() and
        A.name = ‘Tarkan’ and
        A.semanticRole = ‘singer’;
```

Before presenting the remaining two fuzzy queries, fuzzy quantifiers employed in these queries are presented below as fuzzy sets where the *fuzzy_attribute* is a placeholder for fuzzy attributes such as *isEconomical* and *isDomestic* for the *NewsSequence* class as used in the upcoming queries and *obj* is any object possessing the fuzzy attribute under consideration.

$$Exactly/Totally = \{obj \mid \mu_{fuzzy\_attribute}(obj) = 1.0\}$$
$$Mostly = \{obj \mid \mu_{fuzzy\_attribute}(obj) > 0.7\}$$
$$Moderately = \{obj \mid \mu_{fuzzy\_attribute}(obj) > 0.4\}$$
$$Somewhat = \{obj \mid \mu_{fuzzy\_attribute}(obj) > 0\}$$

Eight example *NewsSequence* instances are included in Table 1 with only their relevant attributes to the remaining queries. With the objects in Table 1, the execution of the fuzzy queries, Q2, and Q3, is illustrated in the following paragraphs.

Table 1. 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

`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 1, 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.

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 1, 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. A Text-Based Automatic Annotation Scheme

The semantic annotation of multimedia data is mostly performed in manual or semi-automatic manner. The automatic or semi-automatic annotation schemes often make mere use of the audio-visual components of the data but the technology to extract semantic information from these components is still far from being mature as outlined in Ref. 20.

Multimedia data often has a corresponding text either in the form of closed-caption texts or transcription texts obtained through automatic speech recognition (ASR) techniques. The proper utilization of multimedia texts through information extraction techniques could aid extensively in the automatic annotation of multimedia data. Information extraction (IE) is the task of obtaining useful semantic information such as entities, relations, and events from free texts.[21] The authors of the studies presented in Refs. 20 and 22 employ IE techniques and tools to extract useful semantic information from football video texts in order to automatically annotate football videos. Their works also comprise the utilization of domain-specific ontologies during the IE procedure.

The study in Ref. 23 also points out the importance of using IE techniques, particularly utilization of coreference resolution during semantic modeling of videos, choosing news videos as the domain under consideration. Coreference is a linguistic phenomenon where two or more entities in the texts refer to the same real world entity, and coreference resolution is the identification of the sets of entities (so called coreferential chains) in an input text that corefer.[25] Identification of coreferential chains in the video texts is especially important for semantic annotation of the corresponding video texts with entities, relations, and events since it prevents superfluous extraction of the same underlying semantic objects as distinct ones. To illustrate, the text component of a news video sequence about the president of the USA will probably include expressions like '*president Bush*', '*president*', '*Bush*', '*George W. Bush*', '*president of the USA*' etc. scattered along the whole text. These coreferring expressions are extracted as distinct semantic objects of the video if no coreference resolution attempt is made during the extraction process. With the employment of coreference resolution, these coreferring expressions could be pinned down to a single semantic object. Nonetheless, these expressions should also be retained to aid in the execution of the prospective user queries. In order to represent these retained expressions (or surface forms) in our conceptual data model in Fig. 2, we need to revise our *Object* class by adding a new attribute called *setOfSurfaceForms* and a new method called *getSetOfSurfaceForms()* to access the expressions.

In order to extract semantic entities in political news sequences, a set of lexical resources such as the set of country, city and town names, the set of political status, etc. can be utilized.[24] These semantic entities not only correspond to the semantic *Object* instances in the conceptual data model in Fig. 2, but also some of these instances will be taking part in *Relation* and *Event* instances in the model. However, as pointed out in the previous paragraph, some of these entities may be extracted superfluously due to the fact that they refer to the same underlying real world entity and this situation could be avoided by the identification of the coreferential chains after the salient entity extraction phase.[24] For political news sequences, a simple coreference resolution scheme based on the matching of the tokens (individual linguistic units in text such as words, numbers, and punctuation.[22]) of the previously extracted salient entities turns out to perform well to avoid superfluous extractions.[24]

We have incorporated the automatic entity extraction scheme presented in Ref. 23 to our fuzzy multimedia database system where in its current form it supports videos in Turkish. For instance, if a user poses a query like Q4 to the system after the automatic entity extraction from a political news sequence about the president of the USA, not only the scenes having the expression '*George W. Bush*' in their texts but also the scenes with text including the expressions such as '*president of the USA*', '*president Bush*', etc. will be returned to the user.

```
Q4: Retrieve all news scenes where George W. Bush appears.
```

With the revised *Object* class, the underlying query expression corresponding to the query, Q4, will be as follows:

```
select  S from Scene S, Object O
where   O in S.getObjects() and
        'George W. Bush' in O.getSetOfSurfaceForms();
```

Although it is a promising step towards the automatic annotation of news videos, there are a number of outstanding problems to be dealt with:

- This annotation scheme currently considers only the extraction of semantic entities corresponding to *Object* instances in our conceptual data model. In order to annotate the videos with their complete semantics, the scheme should also attempt to extract and annotate relations and events in the videos. Yet, relation and event extractions are more difficult problems compared to entity extraction in that they require more sophisticated natural language processing (NLP) techniques.
- The automatic annotation scheme requires the video texts with a considerable precision which necessitates the employment of automatic speech recognition systems (ASR) if those texts are not readily available. This is an important problem for languages for which no general-purpose ASR system has been proposed so far, such as Turkish.

The problems described above motivated us to outline the following plans for future research. In order to make the text-based annotation scheme a practical alternative to the manual annotation, we will enhance the automatic annotation component of the fuzzy multimedia database system with the relation and event extraction capabilities by employing the necessary NLP techniques and implementing the NLP tools which are not available. Furthermore, we will extend the component by incorporating a convenient ASR system for English to the component and extend it to support videos in English as well.

6. 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.

The proposed fuzzy multimedia database system has a manual annotation component as well as an automatic annotation counterpart which currently identifies semantic entities if the texts corresponding to the videos in the database are available. We will extend this text-based annotation component to extract and annotate semantic relations and events utilizing more sophisticated natural language processing techniques. We will also incorporate a convenient ASR system for English and extend the annotation component to support videos in English as well.

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, Gozde B. Akar, Turgay Yilmaz, and Yakup Yildirim.

References

1. R. Hjelsvold and R. Midtstraum, Modelling and querying video data, *Proc. 20th Int. Conf. on Very Large Data Bases*, 1999, pp. 686–694.
2. G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide* (Addison-Wesley, Reading MA, 1999).

3. Z. M. Ma, W. J. Zhang, W. Y. Ma, and Q. Chen, Conceptual design of fuzzy object-oriented databases using extended entity-relationship model, *Int. Journal of Intelligent Systems* **16** (2001) 697–711.
4. A. Yazıcı, Q. Zhu, and N. Sun, Semantic data modeling of spatiotemporal database applications, *Int. J. Intelligent Systems* **16** (2001) 881–904.
5. Z. M. Ma, Fuzzy information modeling with the UML, in *Advances in Fuzzy Object-Oriented Databases: Modeling and Applications* (IDEA Group Publishing, 2004).
6. M. A. Sicilia, E. Garcia, and J. A. Gutierrez, Introducing fuzziness in existing orthogonal persistence interfaces and systems, in *Advances in Fuzzy Object–Oriented Databases: Modeling and Applications*, (IDEA Group Publishing, 2004).
7. J. M. Medina, O. Pons, and M. A. Vila, GEFRED: a generalized model of fuzzy relational databases, *Information Sciences* **76** (1994).
8. N. Marin, O. Pons, and M. A. Vila, A strategy for adding fuzzy types to an object-oriented database system, *Int. J. Intelligent Systems* **16** (2001) 863–880.
9. M. Koyuncu and A. Yazıcı, IFOOD: an intelligent fuzzy object-oriented database architecture, *IEEE Trans. Knowledge and Data Engineering* **15** (2003).
10. Z. M. Ma and L. Yan, A literature overview of fuzzy database models, *Int. J. Information Science and Engineering* **24** (2008) 189–202.
11. G. de Tre and R. de Caluwe, Modelling uncertainty in multimedia database systems: an extended possibilistic approach, *Int. J. Uncertainty, Fuzziness and Knowledge-Based Systems* **11** (2003) 5–22.
12. R. S. Aygun and A. Yazıcı, Modeling and management of fuzzy information in multimedia database applications, *Multimedia Tools and Applications* **24** (2004) 29–56.
13. A. Yazıcı, R. George, and D. Aksoy, Design and implementation issues in the fuzzy object-oriented data model, *Information Sciences* **108** (1998) 241–260.
14. D. Küçük, N. B. Özgür, A. Yazıcı, and M. Koyuncu, A fuzzy conceptual model for multimedia data with application to news video domain, in *Proc. Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, Malaga (Spain), 2008.
15. A. Ekin, A. M. Tekalp, and R. Mehrotra, Integrated semantic-syntactic video modeling for search and browsing, *IEEE Trans. on Multimedia* **6** (2004).
16. *ISO/IEC Committee Draft 15 938–5 Information Technology - Multimedia Content Description Interface: Multimedia Description Schemes, ISO/IEC/JTC1/SC29/WG11/N3966* (2001).
17. R. Tusch, H. Kosch, and L. Bözsörményi, VIDEX: an integrated generic video indexing approach, *Proc. 8th ACM Multimedia Conference*, Los Angeles (USA), ACM Press, 2000, pp. 448–451.
18. S. Nepal, M. V. Ramakrishna, and J. A. Thom, A fuzzy object query language (FOQL) for image database, in *Proc. Sixth Int. Conf. Database Systems for Advanced Applications*, 1999, pp. 117–124.
19. http://www.db4o.com/
20. H. Saggion, H. Cunningham, K. Bontcheva, D. Maynard, O. Hamza, and Y. Wilks, Multimedia indexing through multi-source and multi-language information extraction: the MUMIS project, *Data and Knowledge Engineering* **48** (2004) 247–264.
21. R. Grishman, Information extraction, in *The Oxford Handbook of Computational Linguistics*, ed. R. Mitkov, (Oxford Univ. Press, 2003).
22. R. Mitkov (ed.), *The Oxford Handbook of Computational Linguistics*, (Oxford Univ. Press, 2003).

23. T. Declerck, J. Kuper, H. Saggion, A. Samiotou, P. Wittenburg, and J. Contreras, Contribution of NLP to the content indexing of multimedia documents. *Proc. Int. Conf. Image and Video Retrieval (CIVR)*, 2004, pp. 686–694.
24. D. Küçük and A. Yazıcı, Identification of coreferential chains in video texts for semantic annotation of news videos, *Proc. 23rd Int. Symp. Computer and Information Sciences (ISCIS)*, İstanbul (Turkey), 2008.
25. R. Mitkov, *Anaphora Resolution* (Longman, 2002).