

Conceptual Data Modeling of Multimedia Database Applications

S. Aygün, A. Yazıcı and N. Arıca

Department of Computer Engineering, Middle East Technical University

e-mail: {aygun,nafiz,yazici}@ceng.metu.edu.tr

Abstract

In this paper, we present a conceptual data model for multimedia database applications based on ExIFO₂ model. The ExIFO₂ data model is chosen as the conceptual model since it handles both complex objects along with their uncertain and imprecise properties. We enhanced this conceptual model in order to meet the multimedia data requirements. In addition to uncertain and imprecise information, we present a way of handling relationships among objects of multimedia database applications. Events that might be extracted from video or audio are also considered in this study. Finally, the conceptual model is mapped to a logical model, which the fuzzy object oriented data (FOOD) model is chosen, for storing and manipulating the multimedia objects. This mapping is done in a way that it preserves most of the information represented at the conceptual level.

1. Introduction

In recent years, management of multimedia information has been focus of many applications such as on-demand services, digital libraries, museums, on-line shopping and document management. These applications require development and design of new databases where the multimedia data is the resource. Multimedia databases are expected to fulfill the requirements of users for efficiently and flexibly managing multimedia applications. However, the development of multimedia databases is unlikely to follow the footsteps of traditional databases due to the characteristics of multimedia objects, such as image, graphics, audio, and video. As pointed out in [6], one of the challenging issues that multimedia database researchers have to face is the development of conceptual models for multimedia information especially for video, audio, and image. The conceptual model should be rich in its semantic capabilities to represent complex media objects and express their synchronization requirements. Following the conceptual data model, transformation from the model to a logical database scheme is required for storing and manipulating of these objects.

Until now, many conceptual approaches are proposed to model complex data at conceptual level [3,8,13,17]. Since, the conceptual models were developed for domain-specific applications, further steps should be taken to enhance conceptual models to represent the complexity and requirements of multimedia objects. There have been a number of attempts to develop conceptual models for representation of multimedia objects. In [6], these models were classified into several categories; graphical models, Petri-Net based models, object oriented models. In graphical models, information represented as a page that is consisting of a segment of text, graphic codes, executable programs or even audio/video data such as in [5]. All the pages are linked via a labeled graph called hyper graph. In other words, this model specifies higher level browsing features of a multimedia system. Petri-Net models represent various components of multimedia objects and describe their inter-relationships in the form of transitions. In object oriented data models, the basic idea is to represent a real world thing or concept as an object. These models present ways of compositions and semantic modeling of multimedia data at various levels of information granularity.

The conceptual model exploited in modeling multimedia database applications usually needs to be transformed into a logical model in a way that it is straightforward and preserves most of information that is embedded in the conceptual model. In the early methods images were classified as physical image and logical image to adapt the relational database requirements. Logical images are attributes about the image. In GRAIN, the physical and logical images are connected via three tables [4]. In REDI, the structures of image are kept by relational graphs and these graphs are converted to binary relations [4]. Another study is performed for the semantic modeling of image databases to describe inner structure and contents of images [15]. In this study, Yang and Wu develop a semantic image database by utilizing the IFO data model [1]. The semantic model developed is mapped to relational algebra. PROBE [10] is a spatial object-oriented database system based on the architecture of extensibility. Its basic types are object entities and functions where functions represent relationships, properties and operations associated with entities.

It is not always possible to describe all the semantics of real world information precisely, since the observation and capturing of some real world objects are not perfect, hence its modeling and representation are deficient. Therefore, it is inevitable to incorporate uncertainty in many complex applications including multimedia database applications. Uncertainty might arise from the data itself and/or the relations between multimedia objects. For instance, attributes of an image such as color, description of an object and spatial relations between objects may be interpreted in various ways. Consequently, queries involving imprecise and uncertain information may be unavoidable. Consider the following queries:

- **Q1:** “Find all hand images of *pre-teen* males that are *abnormal*”
- **Q2:** “Retrieve images that have colors *similar to* a sunset photograph”.
- **Q3:** “Retrieve all facial images with a *short round* chin and *thin* hair”
- **Q4:** “Retrieve an object *partially-surrounded-by* a *little* of object A”
- **Q5:** "Retrieve videos of all races of Carl Lewis where he is running *very fast*"

Q1 is an example query in a medical image database. In Q1, *pre-teen* and *abnormal* are fuzzy templates. In Q2, *similar-to* is a fuzzy operator, which has to employ similarity matching. Q3 is a query for facial image database in which *short round* and *thin* are fuzzy concepts. Q4 is an example for Geographic Information Systems that contains fuzzy concepts such as *partially-surrounded-by* and fuzzy quantifier *little*. Q5 is an example of query that might be needed in a video. Here *fast* is a fuzzy concept and *very* is a fuzzy modifier. If uncertain, imprecise, and fuzzy information are forced to be precise, then there might be some loss of information. Uncertainty databases has been studied by many researchers such as [11,18].

However, there has not been much research on the development of conceptual models which also deal with uncertainty for multimedia databases. In this study, the conceptual model that we present for multimedia database applications is based on ExIFO₂ data model. ExIFO₂ [17] is a fuzzy object-oriented conceptual data model that includes representation of fuzzy information and handles object-oriented concepts. This model attempts to preserve the acquired strengths of semantic approaches, while integrating concepts of the object-oriented paradigm and fuzziness. ExIFO₂ supports three types of uncertainty: fuzzy, incomplete and null valued attributes. After constructing a data model for multimedia database applications, we also show how the ExIFO₂ schema is mapped to the object oriented data model, FOOD [16,18].

One of the main focuses of this study is modeling of relationships among objects in multimedia data. These

relationships can be either spatial in an image or real-life relationships which might not be extracted directly from the image. Another focus is modeling of mapping from real objects to multimedia objects. We also modeled events that might take place in a video/audio since most of the applications require querying according to the events. Event concept incorporated in our study is similar to the one in [2].

This paper is organized as follows: The following section discusses uncertainty and object-oriented databases. In Section 3, multimedia database applications are explained. The footsteps of conceptual modeling of multimedia applications are presented along with ExIFO₂ in Section 4. In Section 5, mapping of ExIFO₂ data model to FOOD is given. Finally, our conclusions with the further issues are stated.

2.Uncertainty and Object-Oriented Databases

The need for management of uncertainty in database environment is obvious especially for application areas such as multimedia databases, engineering applications, decision problems, control systems, and rule-based knowledge based systems. Most of the studies on fuzzy databases are developed by either using or extending the ordinary relational models. There are only few studies to handle uncertainty and fuzziness under object-oriented databases [11,16]. Fuzzy and uncertain object-oriented databases support rich semantic expressiveness.

The similarity-based fuzzy object-oriented data model (FOOD) [16] facilitates the enhanced representation of different types of imprecision. In FOOD, similarity relation is utilized to generalize equality to similarity. Similarity allows impreciseness in data to be presented and imprecision in data results in uncertainty in classifications. In this model, a similarity matrix is kept for each fuzzy attribute to declare the similarity between the values under the corresponding domain. For example, if the attribute is *color*, then the contents of similarity matrix will be the similarity of *colors* and similarity among these values are in the range [0,1]. The similarity matrix [19] is reflexive, symmetric, and min-max transitive.

In the FOOD model, three types of uncertainty are distinguished at the attribute level, fuzzy, incomplete, and null. *Fuzziness* exists when true data is available in descriptive term rather than as a precise data. The values can be related to each other in imprecise way. For example, *face_expression* of a person in an image can be fuzzy attribute. It is sometimes hard to distinguish *worried* and *sad* person. The *face_expression* can be declared as {*worried, sad*} to state that the *face expression* can be any of *worried, sad*, or combination of these. Another type of uncertainty is *incompleteness* which occurs when true data may belong to a specific set of values. For example, in a film, the main character has the same attributes as a

person. He will have a *name*, *age*, etc. But we might not know the exact age of this person. Instead, we can predict his age as a range such as between 25 and 35. The last type of uncertainty at the attribute level is *null*. If an attribute's type is null, this can imply one of the following: the attribute's value does not exist or it exists but its value is not known or it is not even known whether the attribute's value exists. Consider an example of an event that takes place in video. We might have no information if the event has a reason or not, or we might know that the event has no specific reason or the event has a reason but we might have no information about what it is.

The second level of uncertainty exists on the object and class level. At this level, two kinds of membership are defined to express uncertainty. First one is the membership of an object to a class. For example, in an image database which contains landscape images, all images can be classified into three classes; *forest landscapes*, *sea landscapes* and *urban landscapes*. The image of a *village* which is by the *sea* and next to a *forest* can be an object of all the landscape image classes with different membership degrees. For example, if we consider only the color histogram of the image, we can assign a different membership degree ([0,1]) to each of the classes according to dominating colors. The second type of uncertainty involves in class/subclass (or class/superclass) memberships. The degree of membership of a class to its superclass can vary. For example, class *video-insects* can be a subclass of *video-winged-animals* with a membership degree in [0,1]. This type of membership introduces the notion of *fuzzy classes*.

Attributes that have imprecise values determine the membership degrees of objects in a class. These attributes have *relevance* values that indicate the effect of attribute's value in determination of the likelihood of the objects to its superclass. The membership of object o_j in class C is determined by using the formulation [16] as follows:

$$\mu_C(o_j) = \mathbf{g}[\mathbf{f}(\text{RLV}(a_i, C), \text{INC}(\text{rng}_C(a_i)/o_j(a_j)))]$$

where $\text{RLV}(a_i, C)$ indicates the relevance of attribute a_i to the class C , and $\text{INC}(\text{rng}_C(a_i)/o_j(a_j))$ denotes the degree of inclusion of the attribute values of o_j in the formal range of a_i in the class C . \mathbf{f} represents the aggregation over the n attributes in the class and \mathbf{g} reflects the nature of semantic link existing between an object and a class/superclass. The formulas for computation of the membership degrees are given in [16].

3. Multimedia Database Applications

Multimedia database applications differ from the traditional database applications in the structure of the multimedia objects (MOBs) and media (e.g., audio, video, image) where the MOBs are stored and extracted from.

The properties of MOBs and the media are important in the design of multimedia database applications.

MOBs are usually complex objects composed of other objects that can also be complex. The object-oriented model is expected to be one of the best models to express the object hierarchies. MOBs are audio-visual in nature [9]. This results in imprecise and incomplete description of MOBs and subjective interpretation of the users. This is the case of MOBs where uncertainty and fuzziness needs to be embedded. It is clear that the user will have no idea about how MOBs are stored in the database. The language that the user will use to retrieve multimedia objects or data might consist of uncertain, incomplete, and fuzzy expressions. MOBs are multidimensional and hierarchically structured, and can have some relations among them. Temporal relation is related about the time that objects become active. Some of the basic temporal relations can be stated as *during*, *before*, *overlapping*, *meet* etc. The spatial relation is related with the distribution of the objects in space. Some of the basic spatial relations are *left of*, *behind*, *inside*, etc. These two types of relations are expected to exist in multimedia databases to handle content-based retrieval.

Most research on media focused on image, video, and audio. Image has spatial behavior. Image data does not have temporal behavior since there is no time associated with it. An image can be displayed 10 seconds or 10 minutes. Video is a sequence of images. Each image of a video is called video frame. Video has both temporal and spatial behavior. Video has temporal behavior in the sense that image sequences of the video should be displayed in order and in some dedicated time. Audio has temporal behavior in the sense that it should be played in order and in some dedicated time. Audio/Video is combination of audio and video where it might be necessary to play video and audio in harmony. A video clip is a subset of the sequence of images that form the video such that it has meaning and events that can be defined. The number of video frames does not have to be constant and events can be extracted from variable number of sequential images. An audio clip is some partition of the audio and it has the same above characteristics like a video clip. We assume that each video clip has a corresponding audio clip. An audio clip can also be empty, that is, it does not exist or cannot be played.

4. Conceptual Modeling of Multimedia Database Applications

In this study, we utilized ExIFO₂ data model for conceptually modeling multimedia database applications. Rather than considering every type of media, we only concentrate on video, image, and audio. Our model will be extensible and have the power of representing other structures of multimedia applications. Our main focus is

on handling video and audio which have both temporal and spatial behavior where image has only spatial behavior. We will include uncertain and incomplete information as well as fuzzy objects and classes. In the following subsections, how to handle types, constructors and fragments of ExIFO₂ data model and some additional constructors to form complex objects from multimedia data are discussed along with examples.

4.1 Types

There are three basic atomic types in ExIFO₂ data model: *printable*, *free*, and *abstract*, shown in Figure 1. The file format, such as bitmap, gif, jpeg, etc. of an image can be considered as printable type. The image can be the abstract type. The video frame is an example of free type in the model.



Figure 1. The Basic Atomic Types

In order to handle uncertainty at the attribute level, three more types are introduced: *incomplete-valued*, *null-valued*, and *fuzzy-valued*. The reason for an event can be *null-valued*, whereas the time of the event can be *fuzzy-valued* (Figure 2).



Figure 2. The Uncertain Atomic Types

4.2 Constructors

Two of the constructors that exist in ExIFO₂ data model are *aggregation* and *grouping*. The aggregation constructor represents the aggregation abstraction of semantic models defined by the Cartesian product. This constructor connects a subtype representing a part of an object to the type representing the entire object; thus, building a higher level object. This abstraction ignores some individual differences of the aggregated types. Semantically, the aggregation constructor derives only objects fully supported by the type, but, unlike the grouping constructor, the objects are ordered. For instance, *aggregation* constructor can be utilized in combining *Image attributes* such as *image title*, *format*, *width* and *height* (e.g., "Ferrari", gif, 256, 256) as in Figure 3. These attributes can be extended according to the requirements of the multimedia application. The grouping constructor depicts the type corresponding to sets of data values of an attribute. This constructor is said to have 'AND' semantics, since each member of the set necessarily

and precisely belongs to the set. Therefore, the grouping type of the constructors can capture the attributes that are multi-valued and crisp. All objects in the grouping abstraction are of the same type and not ordered. A set of images that constitute all pictures of a certain automobile, e.g., Ferrari, can be handled by *grouping* constructor.

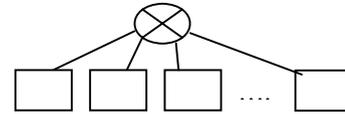


Figure 3. Aggregation Constructors

Composition and *collection* constructors require exclusivity in *aggregation* and *grouping* respectively. The *composition* constructor represents the aggregation abstraction of semantic models defined by the Cartesian product. The only difference of this constructor from aggregation constructor is that, it provides an exclusive structure. The *composition* constructor can be used when components of an object (e.g. image id number) can be uniquely identified. The only difference of *collection* constructor from *grouping* constructor is that, it includes an exclusivity constraint for the grouping. On the other hand, the *collection* constructor can be utilized in order to group the objects in an image (e.g., in a picture of a computer on a table, computer and table are two objects grouped using *collection* constructor, Figure 4a.)

In ExIFO₂, structurally different types can be handled by using the alternative type concept. This constructor represents the IS_A generalization link enhanced with a disjunction constraint between the generalized types.



Figure 4. Collection and Sequence Constructor

4.3 Fragments

The goal of the fragment is to describe the properties of the principal types. Functions can be (simple, complex - multivalued-) or (partial-0:N link-, total-1:N link-) [16].

In our data model, fragments handle relations among the objects in an image. An object might have more than one relation with more than one object (e.g. an object may have two neighbors, *left-of* and *right-of*). Thus we used *total*, *complex-total* and *complex-partial functions*. Each object in an image must have some properties (*total function*) (e.g. color, texture). The object may have zero or more relationships with other objects (*complex partial*

function). Lastly, a relation may consist of one or more objects (*complex-total function*). Examples for the various fragments are shown in Figure 5.

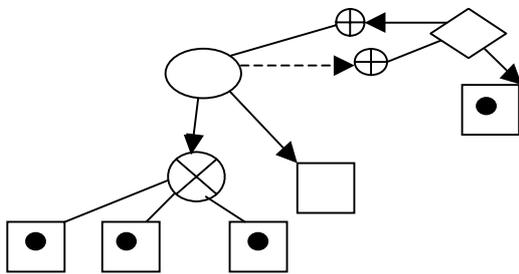


Figure 5. Fragments

4.4 ISA Relationships

The last structural component of the ExIFO₂ model is the representation of ISA-relationships interpreted in two ways as *specialization* and *generalization*. We used specialization link to map objects in an image to the real objects (e.g. ObjectType has the same attributes with the ObjectType1, ..., ObjectTypeN, shown in Figure 6).

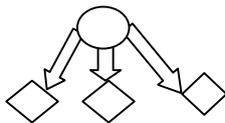


Figure 6. Specialization Link

4.5 Modeling Multimedia Database Applications

In our model, a video is a sequence of clips. A clip is composed of audio and video clips. Events can be defined both for audio and video. Video has attributes such as *title*, *duration*. In this work, we did not list all of the attributes of video and audio, but these can be found in [7, 12]. An event consists of objects that take part in an event, the *reason*, *what*, *how* and *when* it happened. “John runs slowly in the forest to be healthy in the morning” is a possible event in a video clip. *Run in a forest* corresponds to the action (what), *John* represents the object, *to be healthy* denotes the reason for his run, *slowly* corresponds to how fast he runs and *in the morning* represents when he runs. In fact, this event structure is similar to the one in [2]. *Audio clip* is a sequence of audio frames. Each audio frame is a one-dimensional signal that has *frequency* and *amplitude* as physical properties and *title* and *keywords* as description of the signal. Each video clip is a sequence of video frames. The study done by Day et al. [5] also includes a model of video clips, which is somewhat similar

to our approach. In their work, each video clip is composed of segments. Each segment is distinguished from other segments in the objects that they contain. The model that they propose represents a video clip in the form of a directed graph. However, in our approach, clips are distinguished when a new event starts. The video and audio clips have starting and ending frames (times) to be used for synchronization. Temporal relations among objects are retrieved according to starting and ending times of their clips. Each clip has also its corresponding duration. In our model, each clip has at least one event. *Video frame* is an image. Image has a collection of objects and physical properties about the attributes of the image and features extracted from the image. Attributes are an aggregation of *title*, *format*, *width* and *height* of the image. Features of an image are *histogram*, *raw data*, *texture*, and also fuzzy *texture* and fuzzy *color* descriptions. Objects in an image are related to each other with a relation. Objects have minimum bounding rectangles for determining its location and spatial relations with other objects in the image. A relation keeps the object and the relation type with the corresponding objects. Relation type can be *spatial* relation (e.g., *object A is partially-surrounded by object B* is a relation of type *spatial*) which can be obtained from minimum bounding rectangles, or any other relation that might exist such as *family* relation. Objects have dynamic properties that can change in various images. These properties can be *color*, *face expression*, and *state* of the object. Objects also have roles which might differ according to the context. For example, in a video, a person can be an *employee* at his work during the day and a *husband* at home after work. In our representation, an object in the image is, in fact, a specialized version of a real time object such as *person*, *animal*, *staff*.

The result of conceptual modeling of multimedia database applications is represented as a generic conceptual schema using the ExIFO₂ model as shown in Figure 7. This conceptual model does not handle all of the details related to all multimedia applications. It only gives the basics of conceptual modeling of multimedia database applications. The model might need to be extended for a specific application or some parts from the model may be removed for other applications. Here we considered mapping from real objects to objects in multimedia application, handling relationships among objects, and handling uncertainty as the crucial parts for conceptual modeling. We used the *sequence* operator since there exists chronological order between the video frames of a video and audio frames of audio as well as clips of audio/video.

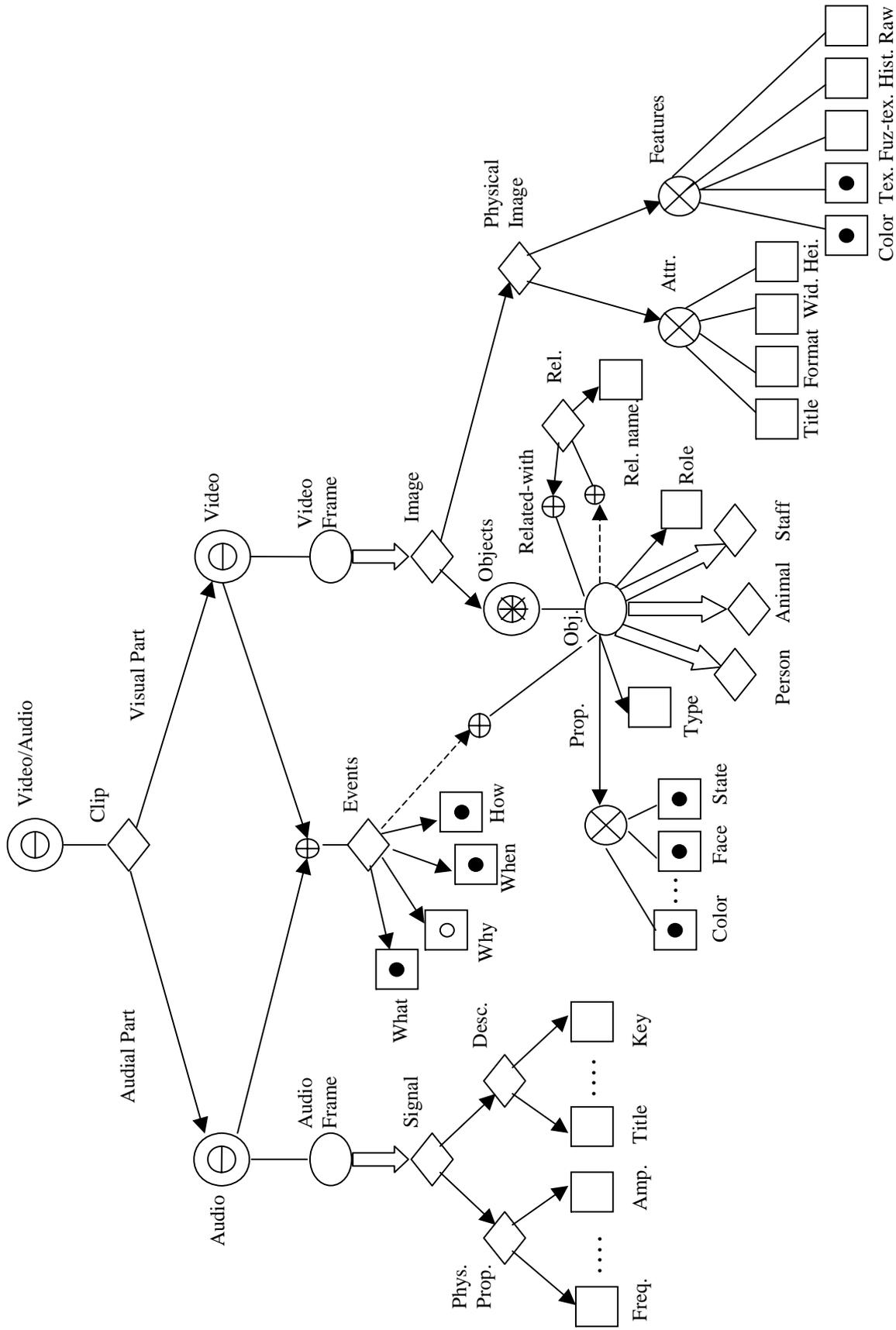


Figure 7: The Whole Schema

5. Mapping Conceptual Model into FOOD

We map the conceptual model to FOOD model as it is described in [17]. The mapping algorithm may be summarized as follows: Firstly, all roots (*abstract* types which are not part of any other type) that exist in the schema are found. Next the fragments from and to these roots are determined. After that IS_A relationships are handled in a way according to specialization and generalization links. In this section, we will apply the algorithm into the each conceptual types and constructors discussed above. We also explain how we handled the sequence operator.

In our model, video/audio is a sequence of clips. The sequence constructor does not exist in ExIFO₂ data model. For sequence operator, a new class is created and it directly inherits from a public class SequenceList. The translation of conceptual structure of Figure 4b can be mapped in the following logical structure of FOOD model.

```
Class C_Video/Audio : Public SequenceList {
  Private:
    C_Clip A_Clip;
}
```

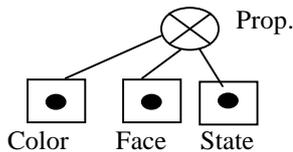


Figure 8. Mapping Fuzzy Attributes

Assume that the FUZZY class exists in the system and class *C_Properties* which include fuzzy attributes and it will inherit from the class FUZZY. For multivalued attributes, the semantics is defined using method FSetOperator. This class will also have membership functions for object/class level and class/superclass level. When aggregation constructor is met, for each link determined, a type is defined as an attribute in source's class. The ranges and relevances for fuzzy values are handled using methods. The Figure 8 results in a code as the following:

```
Class C_Properties : FUZZY {
  Private:
    Multivalued<char *> A_ColorDress;
    Multivalued<char *> A_faceExpression;
    Multivalued<char *> A_State;
    Static C_Properties Ranges;
    Static RLVProperties Relevance;
  Public:
    void SetRanges();
    void CalcMShip();
    void CalcCSCMemShip();
    void SetRelevance();
    void FSetOperator();
}
```

Assume that class SET exists. A class is created for the collection constructor, and it inherits from the class SET. For collection type of class, a method to check the exclusivity is added. Then, Figure 4a will have the following FOOD code:

```
Class C_ObjectCollection : Public Set {
  Private:
    C_Object A_Object;
  Public:
    boolean check_collec_objectcollection();
}
```

When a specialization link is encountered, an inheritance from the target to the source is assumed. For example, Figure 9a will result into the following code:

```
Class C_VideoFrame : Image {
}
```

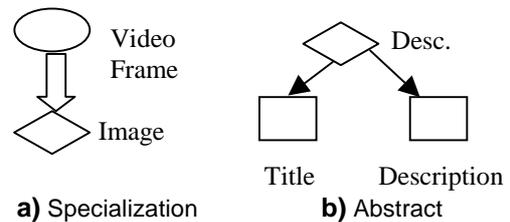


Figure 9. Specialization Link and Abstract Type

The part of schema which includes abstract type like in Figure 9b will be mapped to FOOD code:

```
Class C_SignalDescription {
  Private:
    char * A_Title;
    char * A_Keywords;
}
```

We handled complex partial functions and complex total functions by defining additional methods. For example, Figure 10 depicts that a video should have at least one event and it will be mapped as follows:

```
Class C_Video:Public SequenceList{
  Private:
    C_VideoFrame A_VideoFrame;
    C_Events A_Events;
  Public:
    boolean check_complex_total_function_events();
}
```

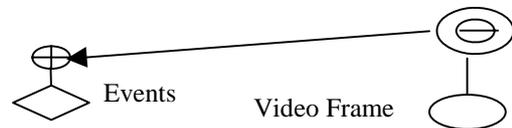


Figure 10. Complex Total Function

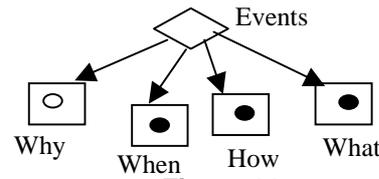


Figure 11. Events

In Figure 11, how events can be modeled is shown. In this model, the ranges, the relevance values of the fuzzy attributes, the semantics that exist between the values and similarity matrix for the fuzzy attribute are also included. The following includes the declaration and assignments of relevance values of the events class, the ranges for the fuzzy attributes and the semantics for the fuzzy attributes.

```
typedef RLVEvents {
    float    RLVWhat;
    float    RLVWhen;
    float    RLVHow;
    float    RLVWhy;
}
void C_Events :: SetRelevance() {
    Relevance.RLVWhat = 0.6;
    Relevance.RLVWhen = 0.4;
    Relevance.RLVHow = 0.2;
    Relevance.RLVWhy = 0.1;
}
void C_Events :: SetRanges {
    Ranges.What= {walking, running, eating, swimming}
    Ranges.When= {morning, afternoon, evening, night};
    Ranges.How={veryslowly,slowly,quickly, veryquickly}
    Ranges.Why= {unk, dne, ni, open, go, see, look}
}
void C_Events :: FSetOperator() {
    A_What.Semantics = OR;
    A_When.Semantics = OR;
    A_How.Semantics = OR;
}
```

These conversions are examples of how we map the schema of our conceptual data model to FOOD. The details of the mapping algorithm is given in [17].

6. Conclusion

In this paper, we showed how a multimedia database application can be conceptually modeled utilizing ExIFO₂ data model. The ExIFO₂ data model allows to represent both complex and uncertain information along with fuzzy objects and classes. We enhanced this model by using a new sequence constructor to manipulate chronological order. In addition to uncertain and imprecise information, we presented a way of handling relationships among objects of multimedia database applications. Besides managing the relationships, we modeled how to map objects in the database to real objects. Then, we also discussed how multimedia database applications could be conceptually modeled. We think that the modeling approach presented here can completely or partially be used in the design of a multimedia application. Finally, we transformed the conceptual model formed by utilizing the ExIFO₂ data model into a logical database model, the fuzzy object oriented data (FOOD) model.

The structure that we build to conceptually model multimedia database applications does not manipulate all of the details related to all multimedia database applications. This model gives the basics of conceptual modeling of such applications. Mapping from real objects to objects in a multimedia database application, handling relationships among objects, and dealing with uncertainty are all considered in this study. But, surely, more need to be done with respect to other possible issues in multimedia database applications.

References

- [1] Abiteboul S., Hull R. (1987) IFO: A formal semantic database model. ACM Trans. on Database Systems. 12: 525-565.
- [2] Adali S., Candan S., Chen S., Erol K., Subrahmanian S. (1996) The Advanced video information systems: data structures and query processing. Multimedia Systems. 4:172-186.
- [3] Bouzeghouband M, Métais E. (1991) Semantic modeling of object-oriented databases. In Proc. VLDB.
- [4] Chang SK, Fu KS. (eds.) (1980) Pictorial Information Systems, LNCS 80 (Springer, 1980).
- [5] Day YF, Dagtas S, Iino M, Khokhar A, Ghafoor A. (1995) Object-oriented conceptual modeling of video data. 11th Int. Conf. On Data Engineering, March, 1995, Taiwan: 401-408
- [6] Ghafoor A. Multimedia Database Course Notes, ACM Multimedia Conf'94.
- [7] Gibbs S., Breiteneder C., Tschritzis D. (1993) Audio/video databases: an object-oriented approach.
- [8] Loucoupoulos P, Zicari R. (1992) Conceptual modeling, databases and CASE: an integrated view of information systems development. Wiley Professional Computing.
- [9] Narasimhalu A.D. (1996) Multimedia databases. Multimedia Systems 4: 226 - 249.
- [10] Orenstein JA, Manola FA. (1988) PROBE spatial data modeling and query processing in an image database application. IEEE Trans. Software Eng. 14(5): 611-629.
- [11] Petry F. (1996) Fuzzy Databases, Principles and Applications, Kluwer Academic Publishers, Boston (USA).
- [12] Schloss G., Wynblatt M.(1995) Providing definition and temporal structure for multimedia data. Mult. Sys.. 3: 264-277.
- [13] Sernadas C, Fiadeiro J. (1991) Towards object-oriented conceptual modeling. Data&Knowledge Engineering, 6:479-508.
- [14] Teisseire M, Poncelet P, Cicchetti. (1994) Towards event-driven modeling for database design. In Proc. of the 20th VLDB Conference, Santiago, Chile: 285-196.
- [15] Yang Li, Wu J. (1997) Towards a semantic image database system. Data & Knowledge Engineering 22: 207-227.
- [16] Yazici, A, R. George, and D. Aksoy, Design and implementation issues in the fuzzy object-oriented data model. Information Sciences (International Journal) (to appear).
- [17] Yazici, A and A. Cinar. Conceptual modeling for the design of fuzzy OO databases. Knowledge Management in Fuzzy Databases, (to appear) O. Pons, A. Vila and J. Kacprzyk, Eds, Physica- Verlag, Heidelberg.
- [18] Yazici A and R. George, Fuzzy Database Modeling. Physica-Verlag, Heidelberg (to be published).
- [19] Zadeh, L.A. (1971) Similarity relations and fuzzy orderings. Information Sciences, Vol.3, No.2: 177-200.