

# Synchronization Attributes and Rules of Active Object-Oriented Database for Multimedia Presentation\*

Seok Jae Yun  
Agency for Defense Development  
PoB 35, Yusong Taejon, Korea

Woo Saeng Kim  
Dept. of Computer Science  
Kwangwoon University  
Wolgye-dong, Seoul, Korea  
woosaeng@dblab.kwangwoon.ac.kr

## Abstract

Recent years have seen much growth in interest for multimedia systems. In line with such growth, the necessity of more effective methodologies to handle multimedia presentations are becoming important. In this paper, we propose synchronization attributes and rules for the multimedia presentation that can be easily accommodated to existing systems. Our proposed method solves the inherent problem of multimedia synchronization by utilizing event-condition-action (ECA) rules that carried out automatically by the system.

## 1 Introduction

The introduction of new development of hardware and software, such as high resolution bitmap displays, CD-ROMs, and image compression techniques have brought radical changes in the way we store and process the data. In line with such trend, databases are being accommodated with various data formats such as audio, video, graphic, image, and text. These new databases are referred as multimedia databases. Initial research of multimedia databases concentrated on the representation of multimedia data based on file system[17]. Soon after, the attempts to incorporate multimedia data within the database systems were carried out[3, 13, 18]. [13] investigates the way to manage media information and to represent the temporal and the spatial information about multimedia objects based on an object-oriented approach. A set of functional requirements is identified for a multimedia server considering database management, object syn-

chronization and integration, and query processing in [3].

As the storage and access techniques for multimedia data are progressed, great interest are now shown in improvements in multimedia synchronization and presentation techniques. This interest has now created a demand for proper synchronization methodology for multimedia data and some solutions are presented. Little and Ghafoor [11] have proposed a synchronization scheme to facilitate storage and retrieval of media elements based on Object Composition Petri Net (OCPN) model. The restricted blocking concept used in the language-based synchronization specification was proposed by Steinmetz[16]. [8] proposed the Synchronization Relation Tree (SRT) as a methodology for the integration of multimedia data. They proposed a message passing protocol to guarantee the temporal relationships among various media objects as specified in a SRT. [6] discussed the treatment of composition and synchronization within an object-oriented framework for multimedia applications. [14] presented algorithm by which a file server can create a relative time system and synchronize media units transmitted by different sources on a network to construct a multimedia object. They developed a feed back technique using which the file server can detect asynchronies in display devices during retrieval of multimedia objects. Recently, [15] addressed the issue of providing flexible multimedia presentation with user participation and suggests synchronization models. However, most researches on multimedia presentation methodologies are static in the sense that the synchronization of multimedia data is managed through the passive way.

Therefore, in this paper, we present a new methodology in which a system automatically solves a synchronization problem based on an active object-oriented database (AOODB). AOODB is a database system that automatically responds to any internal or

\*This research was supported in part by the Korea Science & Engineering Foundation under grants 931-900-042-2

Proceedings of the Fourth International Conference on Database Systems for Advanced Applications (DASFAA'95)  
Ed. Tok Wang Ling and Yoshifumi Masunaga  
Singapore, April 10-13, 1995  
© World Scientific Publishing Co. Pte Ltd

external events without the direct interference of the user. AOODB monitors the occurrence of events such as database updates, examines whether the current database satisfies a predefined condition, and invokes an action such as database updates or invoking other transactions. However, AOODB presented in this paper is free from transaction management in that it also allows easy and flexible management of multimedia presentation based on the concept of synchronization attributes and synchronization rules. Each of synchronization rules is composed of an event-condition-action such that an action is triggered when an event caused by the system satisfies the corresponding condition specified through synchronization attributes. This new action will in turn create a new event and recursively trigger new actions until the multimedia presentation is successfully finished without the dependency of user interaction and the burden of the application program.

This paper is organized as follows. Section 2 presents the basic concept of an active database. In section 3, we describe synchronization attributes and rules based on an active object-oriented database. In section 4, the process of a synchronization operation is illustrated followed by a multimedia presentation example. The last section concludes.

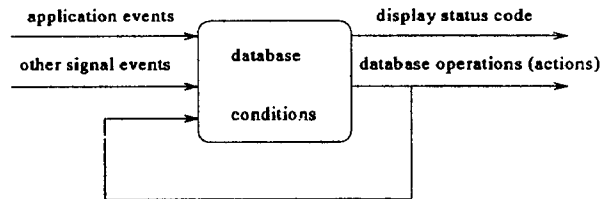
## 2 Active Databases

Conventional databases are passive databases that execute queries or transactions only when a user or an application program requests operations explicitly. Many applications, such as computer integrated manufacturing (CIM), process control, program trading, and network management that require timely response to critical situations are not well served by these passive databases. For these time-constrained applications, it is important to automatically monitor conditions defined on states of the database, and once these conditions occur, to invoke specified actions subject to timing constraints.

Active databases have been defined as database systems that respond automatically to events generated internally or externally to the system itself without user intervention[2]. System responses are expressed using event-condition-action (ECA)-rules[5]. ECA rules have an event that triggers the rule, a condition describing a given situation, and an action to be performed if the condition is satisfied. In this way, the system knows not only how operations are performed but also when operations are performed. Thus, ECA rules are being increasingly used in real-time systems.

An active database monitor conditions triggered

by events, which may be application events(database events such as updates) or other events(nondatabase events such as hardware failure detected by a diagnostic program); if the condition evaluates to true then the action is executed. Generally, the condition and action are expressed in terms of either the high-level language that supports the database or the database user interface language. Figure 1 shows active database example in which the student data is automatically reordered by the rule if the newly entered score is higher than the previously entered one.



Event : Insert Student\_Info (grade)  
 Condition : Student\_Info\_New (grade) > Student\_Info\_Old (grade)  
 Action : reorder (grade)

Figure 1: Example of an Active Database

Much of the current research on active databases focuses upon the inclusion and uses of object-oriented database rules. Several mechanisms are being adopted to support active object-oriented rules within the database system. These can be largely divided into three mechanisms: method-based mechanism[4] where rules are defined within the method; object-based mechanism[5] where rules are treated just as if they were objects and; external mechanism[2, 9] where rules are defined and managed externally with respect to the main system. The synchronization rules proposed in this paper are based on a method-based mechanism.

## 3 Multimedia Presentation in an AOODB

In this section, we present a synchronization methodology in which a multimedia presentation can be carried out automatically once a user specifies a temporal relationships among the media data. Synchronization attributes and rules are defined to show how the proper synchronization can be achieved. We then explain the object-oriented schema graph from which a user can manage a multimedia presentation interactively.

Table 1: Synchronization Attributes

Attribute	Case	Value
StartT	. Start at the time $\alpha$ after scene starts	$\alpha$
	. Start after time $d$ is elapsed from the end of the media data $X(d \geq 0)$	$X+d$
RunT	. Execute only for time $\alpha$	$\alpha$
	. Execute for its run time	null
SyncT	. End upon key is pressed by a user	key
	. End simultaneously with media data $X$	$X$
	. No synchronization is set	null

### 3.1 Synchronization Attributes and Rules

A multimedia presentation is a stage in which the multimedia data are presented through several serial scenes in such a manner that this information is easily perceived by the user. A scene is the smallest presentation unit and each scene can be composed of various media data types. Therefore, each media data must be logically composed to produce significant scenes.

In order to express the variety of temporal relationships among the media data, three attributes are defined: StartT (start time attribute), RunT (run time attribute), and SyncT (synchronization time attribute). As the names suggest, StartT and RunT are the attributes that contain the start time and run time of the media data, respectively. A media data whose StartT attribute has the time value will be executed at the specified time after the scene starts. If StartT attribute of  $M_i$  stores the id of  $M_j$  with the delay time,  $M_i$  will be executed after the specified delay time is passed from the end of the media data  $M_j$ . The execution of the media data whose SyncT attribute has the id of the media data  $M_j$  is terminated as soon as  $M_j$  is ended. Also, the execution of the media data whose SyncT attribute stores a predefined key is terminated when the key is being pressed by a user. The types of these synchronization attributes and their functions are shown in Table 1.

A user can express the temporal relationship of the various media data by assigning the appropriate values to the synchronization attributes within each media data. A user may specify such synchronization by using a Data Definition Language (DDL) or a Data Manipulation Language (DML) supported by the system. One example utilizing DML based on ORION system[7] is shown below. This example shows the scenario of the Scene1 of the presentation P1 where the graphic media data G1 starts after the time unit 10 is passed from the beginning of the Scene1 and executes until a user press the ENTER key.

Table 2: Synchronization Rules

RULE	EVENT	CONDITION	ACTION
R1	time_event	$StartT_i = Current\_time$	Display <sub>i</sub>
R2	time_event	$RunT_i < Current\_time$ $\neg StartT_i$	Terminate <sub>i</sub>
R3	media_end_event	$SyncT_i = M_j$	Terminate <sub>i</sub>
R4	media_end_event &time_event	$StartT_i = M_j + delay$	Display <sub>i</sub>
R5	external_event	$SyncT_i = Predefined\_key$	Terminate <sub>i</sub>

(make Presentation

:Name P1

:Scenes (make Scene

:Name Scene1

:Medium (make Graphic :id G1 :StartT 10

:RunT null :SyncT Enterkey)

...))

When a scene starts, events are occurred and synchronization rules are executed. The type of events are time\_event that is occurred by the system clock, media\_end\_event that is occurred at the end of the execution of a media data, and external\_event that is occurred when the predefined key is being pressed by a user. The synchronization rules are shown in Table 2. Rule R1 is to execute media data having the same attribute value as the current time event. Rule R2 is to terminate execution of media data that have exceeded its specified run time. Rule R3 is triggered whenever a media\_end\_event is occurred. For example, if the value of SyncT attribute of the media data  $M_i$  is equal to the id of the media data  $M_j$ , the system will terminate the execution of  $M_i$  whenever the execution of  $M_j$  is ended in order to attain synchronization. Rule R4 is to execute a media data after the given delay time is elapsed from the end of the specified media data. Finally, rule R5 is to terminate execution of media data whose SyncT attribute value is equal to the predefined key that is being pressed. Rules can trigger other rules until all the media data pertaining to the scene are executed in a synchronized fashion.

Given any two intervals in the development of a multimedia presentation, there are thirteen distinct ways in which they can be related[1]. These relations introduced by J. F. Allen represent how two intervals relate in time; whether they overlap, during, before, etc. Figure 2 shows that these temporal relations can be mapped into the various synchronization attributes and rules defined in this section. Before and meets relationships correspond to rule R4. Overlaps and dur-

ing relationships correspond to a combination of rules R1 and R2. While starts corresponds to rule R1, finishes and equals are mapped to the combination of rules R1 and R3. Here, variable  $t$ ,  $r$ , and  $d$  represent the relative time, the run time, and the delay time, respectively.

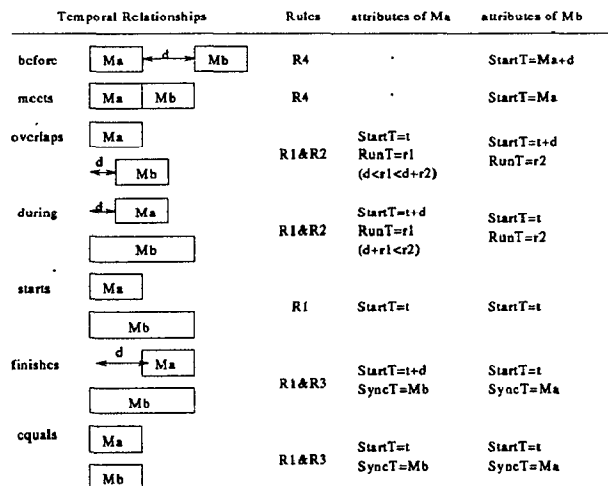


Figure 2: Temporal Relationships Between Two Objects and Corresponding Rules and Attributes

### 3.2 Active Object-Oriented Data Model

It is difficult to represent the complex structure of data and their relationships in a relational data model. Whereas, in an object-oriented data model, one can easily construct complex structure and relationships of data and treat heterogeneous data in a consistent way. Furthermore, an object-oriented data model can easily incorporate the various ECA rules by utilizing methods. The object-oriented schema graph that manages multimedia presentation along with synchronization attributes and rules is shown in Figure 3.

The presentation class includes various attributes and methods from which a user can manage a multimedia presentation interactively. It includes attributes such as the name of a presentation object and methods to perform the various presentation operations such as play, stop, exit, and so on. The scene class which has a part-of relationship with a presentation class includes attribute such as the name of a scene object and medium that reference to various media objects. The media class is a class that incorporates various synchronization attributes and rules. Graphics, sound, image and text have their unique attributes and methods in addition to common attributes and methods such as synchronization attributes and rules inherited

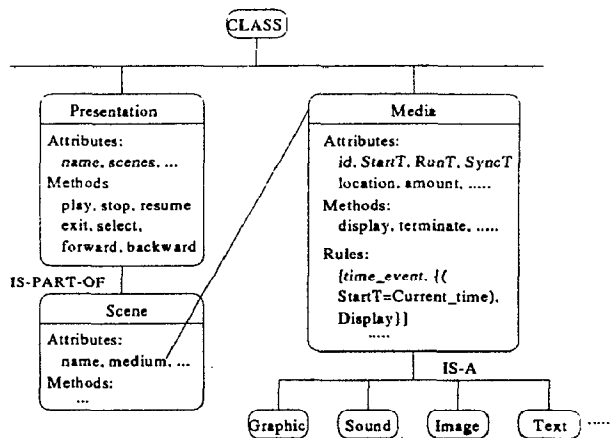


Figure 3: Schema Graph for Multimedia Presentation

by the media class. Functions that do not appear in this figure also can be easily added as either attributes or methods by an object-oriented paradigm. Once a user starts a scene, all media objects will be automatically executed in a synchronized fashion by the rules.

## 4 Multimedia Presentation Example

In this section, we demonstrate how our proposed method can automatically solve the multimedia synchronization problem. We first shows one multimedia presentation scenario. We then illustrates the process of the synchronization operation.

### 4.1 Presentation Scenario

Figure 4 is one example of a multimedia presentation consists of four scenes. Presentation1 is to introduce the natural science college. Each scene contains a brief introduction of the respective department.

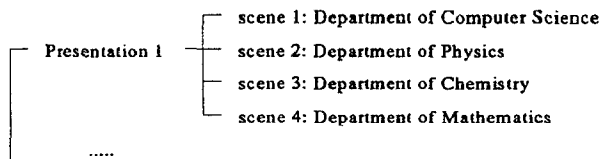


Figure 4: Presentation Scenario

The temporal relationship between all the media data that are composed in a scene can be represented in a Time Line Diagram (TLD). The left end of TLD represents the beginning of the scene, while the right end represents the end of the scene. Figure 5 shows the temporal relationships of various media data of the

scenel in a TLD format. This scenario uses both text and sound for explanations, a still image to show the department building, and audio and video images to introduce the members of each laboratory. As soon as the scene starts, media data Audio1, taking time unit 3 to complete execution, is first executed. During the execution of Audio1, Text starts at time unit 2 and proceeds for time unit 30. Both Audio2 and Image is started as soon as Audio1 is ended. While Audio2 proceeds for the time unit 5, Image proceeds until the Enter key is pressed. After time unit 4 has passed from the end of Audio2, both Audio3 and Video start simultaneously. Finally, Audio3 is terminated when Video is ended to attain synchronization.

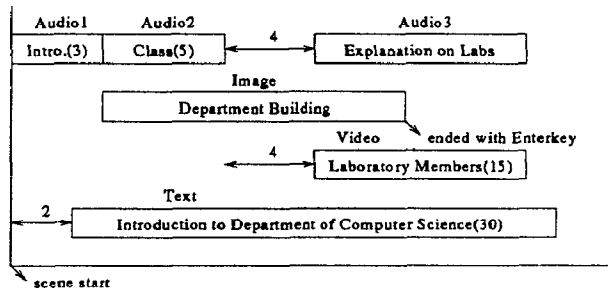


Figure 5: Time Line Diagram

A user should express the temporal relationship of the various media data by assigning the appropriate values to the synchronization attributes within each media data before the presentation starts. Following example shows how a temporal relationships among media data shown in Figure 5 can be specified by a DML like language.

```
(make Presentation
:Name Presentation1
:Scenes(
...
(make Scene
:Name Scene4
:Medium(make Audio :id A1 :StartT 0 :RunT 3 :SyncT null)
      (make Audio :id A2 :StartT A1 :RunT 5 :SyncT null)
      (make Audio :id A3 :StartT A2+4 :RunT null :SyncT V)
      (make Image :id Ii :StartT A1 :RunT null :SyncT Enterkey)
      (make Video :id Vi :StartT A2+4 :RunT 15 :SyncT A3)
      (make Text :id Ti :StartT 2 :RunT 30 :SyncT null) )))
```

## 4.2 Process of Synchronization Operation

The process of the synchronization operation of the scene introducing the Department of Computer Science, through the use of various synchronization

rules is illustrated in Figure 6. In this figure, the time.event that is generated as the scene starts first executes Audio1 because its StartT value is equal to zero. Once the time unit becomes to 2, Text is executed in the similar fashion. When the execution of a media data is initiated, its run time is continuously checked. Once a media data satisfies its runtime condition, the system terminates the execution of the media data. Therefore, the execution of Audio1 is terminated at the time unit 3. At that moment, the media\_end\_event generated by Audio1 initiates the execution of both Audio2 and Image, whose StartT attribute values are equal to the id of Audio1. However, the termination conditions of those two media data are different. In the case of Image, since the value of SyncT attribute is Enterkey, it ends when the Enterkey is being pressed; while, Audio2 whose RunT attribute has value 5 ends at the time unit 8. After the media\_end\_event is generated by Audio2, the time unit 4 must be elapsed to initiate the execution of both Audio3 and Video. Since the runtime specified in the Video data is 15, the execution of Video will be terminated at the time unit 23. As soon as Video is terminated, Audio3 is also terminated because the SyncT value of Video is equal to the id of Audio3. Finally, Text executes for the time unit 30 and then ends. In this fashion, a user can easily achieve a complex and diverse multimedia presentation with the use of appropriately defined synchronization attributes and synchronization rules.

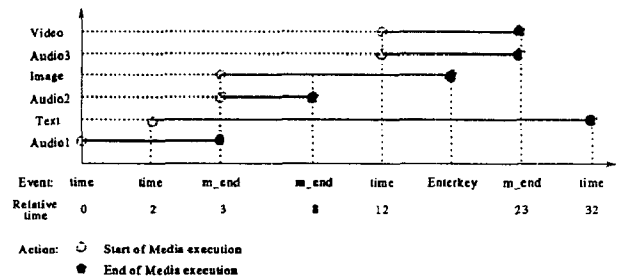


Figure 6: Process of Synchronization Operation

## 5 Conclusion

In short, we have presented a synchronization methodology for multimedia presentations based on an active object-oriented database. In order to achieve temporal synchronization, we defined synchronization attributes and rules, and described how such rules are processed automatically without burden of the user or application program. However, our proposed method is not restricted to the specific platform. It can be easily adapted to other systems or even programming paradigm.

We plan to investigate the possibilities of combining the synchronization rules and conventional ECA rules to explore new database applications. We are currently implementing and evaluating these ideas

## References

- [1] J. Allen, "Maintaining Knowledge about Temporal Intervals," *Communications of the ACM*, Vol.26, No.11, pp.823-843, Nov. 1983.
- [2] C. Bauzer, et al., "A Mechanism for Managing Rules in an Object-oriented Databases," *Altair Technical Report*, 1990.
- [3] P.Berra, C.Y.AR.Chen, A.Ghafoor, C.C.Lin, T.D.C.Little, and D.Shin. "Architecture for Distributed Multimedia Database System". *Computer Communications*, May 1990.
- [4] B. Catriel, et al., "A Model for Active Object-Oriented Database," *Proc. 17th Int. Conf. on VLDB*, September, pp.337-349, 1991.
- [5] U. Dayal, et al. "McCarthy Rules Are objects Too: A knowledge Model for An Active, Object-Oriented Database System," *Pro. 2nd. Int. Workshop on OODBMS*, K.R. Dittrich (ED.), Springer-Verlag, pp.129-143, 1988.
- [6] Gibbs, S., Dami,L., and Tsichritzis.D, "An Object-Oriented Framwork for Multimedia Composition and Synchronization", *Eurographics Multimedia Workshop*, 1991.
- [7] Won Kim, "Introduction to Object-Oriented Databases", *MIT Press*, 1990
- [8] W. S. Kim, E. Lim, and J. Srivastava. "Timing Specification and Synchronization for Multimedia Information System," *Proceedings of the IEEE Workshop on Architectural Aspects of Real-Time Systems*, Dec, 1991.
- [9] A. M. Kotz, et al., "Mulle Supporting Semantic Rules by a Generalized Event/Trigger Mechanism in Advance in Database Technology, EDBT, Venice, pp.76-91, 1988.
- [10] T. D. C. Little and A. Ghafoor, "Multimedia Object Models for Synchronization and Databases," in *Proc. 6th Int. Conf. on Data Engineering*, pp.20-27, 1990.
- [11] T.D.C. Little and A.Ghafoor. Synchronization and Storgae Models for Multimedia Objects. *IEEE Journal on Selected Areas in Communications*, April 1990.
- [12] T.D.C. Little, A.Ghafoor, C.Y.R. Chen, and P.B.Berra. *Multimedia Synchronization. Data Engineering Bulletin*, 1991
- [13] Masunaga. Y. "An Object-Oriented Approach to Multimedia Database Organization and Management. *International Symposium on Database Systems for Advanced Applications*, April, 1989.
- [14] P. Venkat Rangan, Srinivas Ramanathan, Har-rick M. Vin, and Thomas Kaepner, "Media Synchronization in Distributed Multimedia File Systems", *IEEE ComSoc International Workshop on Multimedia Communications*, April, 1992.
- [15] B. prabhakaran and S.V.Raghavan, "A Synchronization models for Multimedia Presentation with User Participation," *ACM Multimedia 93*, August, 1993.
- [16] R. Steinmets. *Synchronization Properties in Multimedia Systems. IEEE Journal on Selected Areas in Communications*, April 1990.
- [17] D. Tsichritzis, et al., "A Multimedia Office Filing system," in *Proc. VLDB*, pp.2-7, 1983.
- [18] D. Woelk, et al., "Multimedia Applications and Database Requirements," in *Pro. IEEE Computer Society Office Automation Symposium*, pp.180-189, 1987.