Middle-Tier for Multimedia Synchronization [*]

Ramazan Savaş Aygün
Dept. of Computer Science and Eng.
State University of New York at Buffalo
aygun@cse.buffalo.edu

Aidong Zhang
Dept. of Computer Science and Eng.
State University of New York at Buffalo
azhang@cse.buffalo.edu

ABSTRACT

The gap between the synchronization specification and the synchronization model limits user interactions for a multimedia presentation. The middle-tier for multimedia synchronization handles the synchronization rules that are directly extracted from the specification. In addition to these rules, the middle-tier also manages implicit synchronization rules which are not specified but can be extracted from other rules. The synchronization rules generated by the middle-tier assists the synchronization model to provide user interactions while keeping the synchronization specification minimal. We give examples of how these rules are generated from SMIL expressions.

1. INTRODUCTION

There has been a lot of research done on inter-stream multimedia synchronization. First models require the specification in terms of time [4]. Specification by only using time was not satisfactory because of synchronization requirements in distributed environments. There has been models which require constraints, relationships among media streams [5]. Models which consider events were also proposed [6, 8]. As the specification is provided by events, constraints, synchronization expressions or relationships among media streams, the user interactions were either limited or the specification became more complex than required [3].

If we just consider the most recent models like NSync [3], PREMO[6] and FLIPS [8], it is realized that user interactions are limited or the specification is complex or both. If we only consider VCR-type user interactions, NSync does not provide the backward functionality and the skip functionality requires the user to specify the operations to be performed for each interval in the presentation. PREMO is an event-based model which also handles time operations. PREMO only allows simple user interactions like play, pause and resume. FLIPS can only handle event-based operations and does not provide backward functionality. It provides a limited skip functionality. Multimedia applications like distant education over Internet and sports applications require all the VCR-type functionalities. In the previous (time-based) models, most of the information given by the user was satisfactory for the model and user interactions. As the synchronization requirements are specified by events, constraints or synchronization expressions, the information to the system became implicit. To overcome this, either the user has to specify more information as in NSync or the functionalities are limited as in FLIPS and PREMO.

We propose a middle layer between the specification and the synchronization model which will assist the synchronization model to provide user interactions while keeping the synchronization specification minimal. We will call this middle layer as *middle-tier*. The middle tier is previously defined as the logical layer in a distributed system between a user interface or Web client and the database. It is a collection of business rules and functions that generate and operate upon receiving information. The middle-tier for multimedia synchronization handles synchronization rules that can be extracted explicitly from the user specification and synchronization rules that can be deduced implicitly from explicit synchronization rules. The middle-tier reduces the amount of user specification while increasing the power of the synchronization model. A synchronization rule is in the form of an Event-Condition-Action (ECA) rule.

In the next section, we will explain the middle-tier and its functionalities. In Section 3, we express the events, conditions and actions of a synchronization rule. In Section 4, we explain how synchronization rules are extracted from SMIL [1] expressions and how implicit rules are deduced. The last section concludes our paper.

2. THE MIDDLE-TIER

The middle-tier for multimedia synchronization first handles the rules that can be extracted from the synchronization specification. Synchronization requirements are stored in rules since each synchronization rule is simple and can be processed easily to generate other rules. Once the rules from the specification are extracted, the synchronization rules for the backward presentation are generated. The extracted rules are fed into the synchronization model. The synchronization model contains a rule manager to manage these rules. The timeline for events and actions are generated in case the course of the presentation changes after user

[*] This research is supported by NSF grant IIS-9733730.

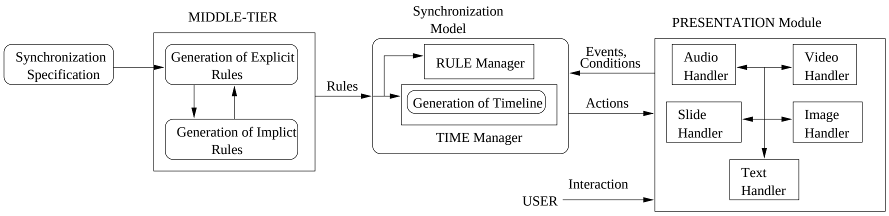

Figure 1: The role of middle-tier.

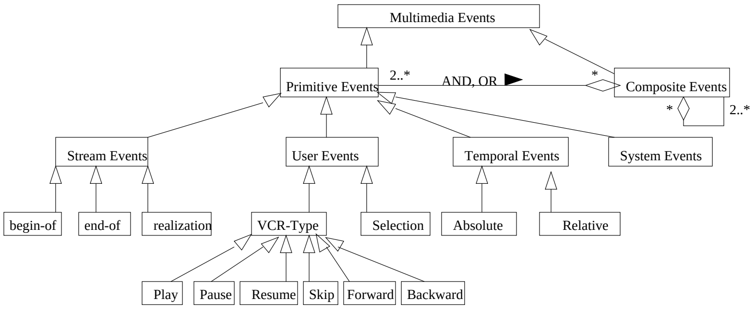

Figure 2: The event hierarchy.

interactions like skip and change direction. When the presentation module receives an event from the user or one of the stream handlers, it informs the event and the current condition of the presentation to the synchronization model. The synchronization model determines if any of the rules are satisfied and if a rule is satisfied it informs the necessary actions to the presentation module. The framework is shown in Figure 1.

3. COMPONENTS OF SYNCHRONIZATION RULES

Each synchronization rule is in the form of an ECA rule. It has an event expression, condition expression and action expression. Once the event expression is satisfied, the condition expression is checked. If the condition expression is also satisfied, the actions are executed. We will explain the events, conditions and actions of a multimedia presentation in this section. In a multimedia system, the events may be triggered by a media stream, the user or the system.

3.1 Events

The goal in inter-stream synchronization is to determine when to start and end streams. The start and end of streams depend on multimedia events. The hierarchy of multimedia events are depicted in Figure 2. The user has to specify information related with the stream events. Allen specifies 13 temporal relationships [2]. Relationships *meets, starts* and *equals* require the *beginning-of* event for a stream. Relationships *finishes* and *equals* require the *end-of* event for a stream. Relationships *overlaps* and *during* require *realization* event to start (end) another stream in the mid of a stream. The relationships *before* and *after* require temporal events since the gap between two streams can only be determined by time. Temporal events may be *absolute* with respect to a specific point in a presentation (e.g. the beginning of a presentation). Temporal events may also be *relative* with respect to another event. A multimedia event is specified by its generator, event type and event data. Event data may be the frame number in a video stream.

Users can also cause events such as start, pause, resume, (fast-slow) forward, (fast-slow) backward and skip. These events have two kinds of effects on the presentation. Skip and backward change the course of the presentation. Others only affect the duration of the presentation. These events may change the state of a stream such as playing the audio stream in the mute state.

Composition of events may be required to trigger actions instead of a single event. Composite events can be created by boolean operators AND and OR:

$AND(e_1, e_2, ..., e_n)$: All events(e_1 to e_n) should be signaled to trigger an action.
$OR(e_1, e_2, ..., e_n)$: At least one of the events should be signaled to trigger an action.

3.2 Condition

A condition indicates the status of the presentation and its media objects. The most important condition is the direction of the presentation. The receipt of the events matter when the direction is forward or backward. Other types of conditions include the states of the media objects. For example, an audio stream may be turned off during fast-forward presentation and events that are expected by this stream may not be received.

3.3 Action

An action indicates what to execute when conditions are satisfied. *Starting* and *closing* a stream, and *displaying* or *hiding* images, slides and text are sample actions. There are two kinds of actions: *Immediate* Action and *Deferred* Action. *Immediate* action is an action that should be applied as soon as the conditions are satisfied. *Deferred* action is associated with some specific time. The deferred action can only start after this specific time has been elapsed. If an action has started and had not finished yet, that action is considered as an alive action.

4. SYNCHRONIZATION RULES AND SMIL

Synchronization rules determine the rules for synchronization requirements. It has 3 parts, event expression, condition expression and action expression. A multimedia presentation usually contains more than one rule. The streams may be grouped as parallel or sequential in SMIL. The *par* statement enables parallel grouping of streams. The *seq* statement enables sequential grouping of streams. Start time of a stream may be relative with respect to another stream or a reference point. This reference point may be the beginning of the group. End time of a sequential group is determined by the last child in the group. A parallel group can end when its first, last or specific element ends. The following example (Figure 3) will be used for explaining synchronization rules: 1) start audio ”cnn.aiff” when the presentation starts 2) start video ”cnn1.mpv” 0.5 seconds after the beginning of the presentation 3) display text ”leader_title.html” after 0.5 seconds after video ”cnn1.mpv” starts 4) start video ”cnn2.mpv” 2 seconds after the end of video ”cnn1.mpv” 5) video ”cnn1.mpv” and ”cnn2.mpv” are played sequentially but parallel with audio ”cnn.aiff” and text ”leader_title.html” 6) when the last element of this parallel session (which is audio ”cnn.aiff”) ends, start audio cnn2.aiff” and video ”cnn3.mpv” in parallel. The SMIL expression where region information is ignored will be as follows:

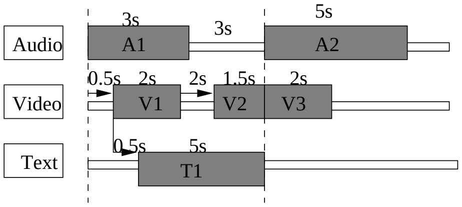

Figure 3: A sample presentation.

```
<seq>
   <par endsync=”last”>
      <audio id=”a1” src=’cnn.aiff” />
      <seq>
        <video id=”v1” src=”cnn1.mpv” begin=”0.5s” />
        <video id=”v2” src=”cnn2.mpv” begin=”2s” />
      </seq>
      <text id=”t1” src=”leader_title.html”
        begin=”id(v1)(0.5s)” dur=”5s”/>
   </par>
   <par>
      <video id=”v3” src=”cnn3.mpv” />
      <audio id=”a2” src=’cnn2.aiff” />
   </par>
</seq>
```

4.1 Explicit Synchronization Rules

Explicit rules are generated by processing the synchronization specification. The beginning of a group is determined by an event signaled from another group, a stream or the user. If the group is the first group that will be presented in the multimedia presentation, the user event *user-start* determines the beginning of the presentation.

The parallel grouping corresponds to the list of all actions that will be started when the group will start. Thus, if the parallel grouping is like

```
<par>
   <... id=”id1” ...>
   <... id=”id2” ...> ...
   <... id=”idn” ...>
</par>,
```

the synchronization rule will be as follows:

R_{par} :	ON	...
	CONDITION	direction=forward
	ACTION	start id1 start id2 ... start idn

In the sequential grouping, the end of a stream triggers start of another stream. If the sequential group has n elements, there will be n-1 rules for the group. Thus, if sequential grouping is like

```
<seq>
   <... id=”id1” ...>
   <.. id=”id2” ...> ... <... id=”id(n-1)”> <.. id=”idn” ...>
</seq>
```

The synchronization rules that will be generated will be as follows:

R_{seq1} :	ON	...
	CONDITION	direction=forward
	ACTION	start id1
R_{seq2} :	ON	the end of id1
	CONDITION	direction=forward
	ACTION	start id2
	...	
$R_{seq(n-1)}$:	ON	the end of id(n-1)
	CONDITION	direction=forward
	ACTION	start idn

Notice that the direction is considered as forward in the condition part since the user specifies the requirements for the forward presentation. If time is associated in the start of a stream, time is considered part of the action rather than part of the event. Because including time in the event expression increases the number of rules significantly, since the same event may also trigger other actions.

There are also some problems in SMIL specification. A stream may start after specific time has been elapsed with respect to a reference point. For example, it is possible to specify to start audio stream 10 seconds after the beginning of a video stream at 30 frames/second. But it is not possible to specify to start audio stream after 300 frames of video stream is displayed. In SMIL, only following rule can be generated

ON	the beginning of video
CONDITION	direction=forward
ACTION	start audio after 10 seconds

But, the following rule might need to be generated and this is why realization event is required:

ON	frame 300 realization of video
CONDITION	direction=forward
ACTION	start audio

The following synchronization rules are the explicit rules for forward presentation for the example specified before:

R_1 :	ON	the user start of presentation
	CONDITION	direction=forward
	ACTION	start a1
		start v1 after 1.0s
R_2 :	ON	the beginning of v1
	CONDITION	direction=forward
	ACTION	start t1 after 0.5s
R_3 :	ON	the end of v1
	CONDITION	direction=forward
	ACTION	start v2 after 1.0s
R_4 :	ON	the end of t1
	CONDITION	direction = forward
	ACTION	end a1
		end v2
		start a2
		start v3

4.2 Implicit Synchronization Rules

The rules for the backward presentation can be generated using the rules for the forward generation. Three elements must be considered for backward presentation: events, actions and time. The goal is to determine what streams to start on what events. The relations between the forward presentation and backward presentation are listed in [7]. It depends on what events causes which actions.

Another component is the timeline. Time is explicitly given in time-based applications and models. Timeline has to be determined for user interactions which change the course like skip and change direction. Timeline has to determine when events generated, when event expressions are satisfied, when actions start and end in nominal presentation. The generation of timeline is given in [7]. The deduced rules based on the relationships and the time are as follows:

R_5 :	ON	the user backward presentation
	CONDITION	direction=backward
	ACTION	backward a2
		backward v3 after 3s
R_6 :	ON	the beginning of a2 and v3
	CONDITION	direction=backward
	ACTION	backward v2
		backward t1
		backward a1 after 2.5 s
R_7 :	ON	the beginning of v2
	CONDITION	direction=backward
	ACTION	backward v1 after 2s

5. CONCLUSION

In this paper, we introduced the middle-tier for inter-stream multimedia synchronization. The middle-tier handles the rules that are extracted from the synchronization specification explicitly and the rules that are deduced from these rules (like rules for backward presentation). The timeline for events and actions are necessary components for multimedia synchronization if user interactions which change course of the presentation are provided. We give examples how rules are generated from SMIL expressions. SMIL expressions lack synchronization in terms of realization events which is necessary in distributed presentations. Our model is not limited by SMIL specification. We gave examples from SMIL to show the applicability of our model.

6. REFERENCES

[1] The cwi smil page.http://www.cwi.nl/smil.

[2] J. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of ACM*, 26(11):823–843, November 1983.

[3] B. Bailey, J. Konstan, R. Cooley, and M. Dejong. Nsync - a toolkit for building interactive multimedia presentations. In *Proceedings of ACM Multimedia*, pages 257–266, September 1998.

[4] S. Gibbs, C. Breiteneder, and D. Tsichritzis. Data Modeling of Time-Based Media. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 91–102, Minneapolis, Minnesota, May 1994.

[5] R. Hamakawa and J. Rekimoto. Object composition and playback models for handling multimedia data. *Multimedia systems*, 2:26–35, 1994.

[6] I. Herman, N. Correira, D. A. Duce, D. J. Duke, G. J. Reynolds, and J. V. Loo. A standard model for multimedia synchronization: Premo synchronization objects. *Multimedia systems*, 6(2):88–101, 1998.

[7] R. S. Aygun and A. Zhang. Interactive multimedia presentation management in distributed multimedia systems. In *Proc. of Int.Conf. on Information Technology: Coding and Computing*, Las Vegas, Nevada, April 2001.

[8] J. Schnepf, J. Konstan, and D. Du. FLIPS: Flexible Interactice Presentation Synchronization. *IEEE Selected Areas of Communication*, 14(1):114–125, 1996.