Dividable Dynamic Timeline-based Authoring for SMIL2.0 Presentations

Chun-Chuan Yang, Chen-Kuei Chu, and Yung-Chi Wang
Multimedia and Communications Laboratory
Department of Computer Science and Information Engineering
National Chi Nan University, Taiwan, R.O.C.
ccyang@csie.ncnu.edu.tw

ABSTRACT

Timeline-based editing provides users an intuitive and friendly way for multimedia authoring, but the schedule-based and deterministic property of timeline results in the lack of the ability for supporting non-deterministic temporal behavior, which is one of the key features of SMIL2.0. This paper presents the effort of supporting non-deterministic temporal behavior by timeline-based editing. The concept of *Dividable Dynamic Timeline (DDTL)* is proposed in the paper, which includes two novel features: *dividable timeline* and *dynamic section*. With DDTL, authors can create interactive multimedia presentations while enjoying the convenience of timeline. Mechanisms of converting from DDTL editing results to SMIL2.0 are also presented in the paper.

1. INTRODUCTION

Synchronized Multimedia Integration Language (SMIL) [1-3] developed by *WWW Consortium (W3C)* provides Internet users a mechanism to compose multimedia documents. With SMIL, authors can create multimedia presentations integrating video, audio, animation, image, text, etc. There are two versions of SMIL specification that had been released. The current version of SMIL (SMIL2.0) enhances the previous scheduled-based version (SMIL1.0) by a strong support of user interaction with a declarative event-based timing.

The direct way to create a SMIL document is to use a text editor and start writing SMIL tags as most of the programmers do. For non-professionals, it is much better to have an authoring system that helps users compose SMIL documents in a visualized (WYSIWYG) way. Two major categories for visualized SMIL1.0 authoring are (1) structure-based editing, and (2) timeline-based editing. Structure-based editing is primarily based on the visualization of SMIL temporal relations (i.e. <seq> and <par>), and users need to organize nested <seq> and <par> blocks. On the other hand, timeline-based editing hides the language structure of SMIL by visualizing the playback time and duration of each object in the timeline manner providing users a more intuitive way to understand and easily control the timing of each object.

Few efforts for SMIL2.0 authoring have been made so far, and most of the existing solutions [9-11] provide structure-based editing. In our previous work, we proposed an efficient modeling technique [4, 5] for SMIL1.0, and based on that technique, we designed and implemented a friendly and powerful timeline-based SMIL1.0 authoring system called *SMILAuthor* [6]. In light of easy learning of timeline-based editing, we intend to extend our effort in SMIL1.0 authoring to support SMIL2.0. However, the feature of event-based timing in SMIL2.0 introduces non-deterministic temporal behavior in a presentation which means the accurate playback time (and duration) of some media objects as well as the total length of the presentation cannot be determined before run-time. Apparently, the original timeline-based scheme cannot support authoring of non-deterministic temporal behavior. Thus, two novel features, namely *dividable timeline* and *dynamic section*, are proposed in this paper to support non-determinism in timeline-based editing. The new editing scheme is thus called *Dividable Dynamic Timeline-based (DDTL-based)* authoring.

The rest of the paper is organized as follows. First of all, the concept of DDTL as well as the kernel mechanisms in DDTL-based authoring is presented in section 2. Converting of DDTL data to SMIL2.0 format is explained in section 3. Finally, section 4 concludes this paper.

This work was supported in part by the National Science Council, Taiwan, R.O.C., under grant NSC 92-2213-E-260-028

2. DIVIDABLE DYNAMIC TIMELINE (DDTL)

Normally authors must specify the exact playback time and duration of each object in timeline-based editing. According to an object's playback time and duration, a timeline segment representing that object is displayed at corresponding position on the time axis. In order to introduce non-deterministic temporal behavior in timeline and decide the extent that timeline-based editing can do for SMIL2.0 authoring, we need to investigate the synchronization characteristics in SMIL2.0.

To systematically deal with SMIL2.0, an extension of our SMIL1.0 modeling technique had been proposed in our previous work [7, 8]. The new model was called *Extended Real-Time Synchronization Model (E-RTSM)*. Figure 1 shows a sample SMIL2.0 code snippet and its corresponding E-RTSM model. It is easy to understand from the model that non-deterministic temporal behavior in the code snippet comes from the non-deterministic events (denoted by a double circle with a "?"). Moreover, two types of the non-deterministic events were identified in the E-RTSM model: *splitting event* and *non-splitting event* [8]. An event is a splitting event when removing the event results in two separate parts in the model. The player must wait for the occurrence of a splitting event before it can continue playing the rest of the presentation. Event *Btn3.Click* in Figure 1 is a splitting event. Events *Btn1.Click* and *Btn2.Click* are non-splitting events.

Inspired by the two different types of non-deterministic events, we introduce two novel features in timeline-based editing: *dividable timeline* (from the idea of splitting event) and *dynamic section* (from the idea of non-splitting event), which are explained respectively in the following subsections.

2.1. Dividable Timeline

We define *TL-Divide* operation enabling users to divide a timeline table into two sequentially separated timeline tables and associate a non-deterministic splitting event (e.g. user's mouse action) with the beginning of the latter timeline table. An example of TL-Divide operation is shown in Figure 2. To divide a timeline table, the author

```
<seq>
  <par endsync = "A1">
    <audio id=A1>
    <video id=V1 begin="Btn1.Click"
                 end="Btn2.Click">
  </par>
  <par begin="Btn3.Click">
     <audio id=A2>
     <video id=V2>
  </par>
</seq>
```

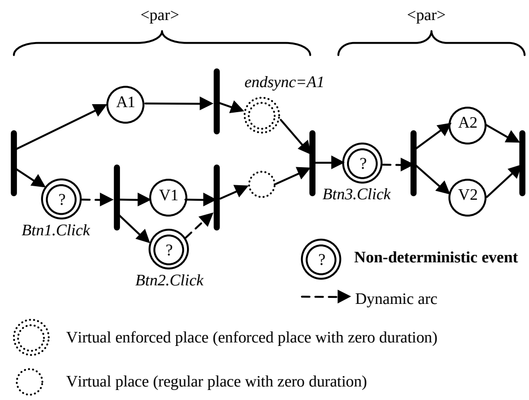


Figure 1. A sample SMIL2.0 code snippet and its corresponding E-RTSM model

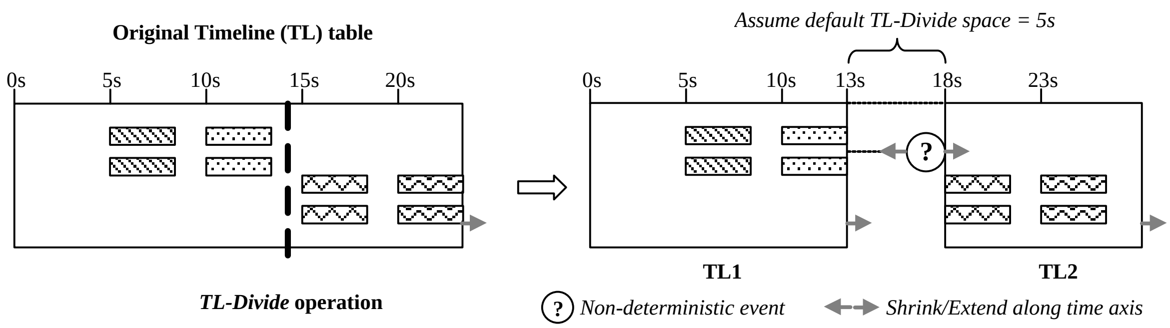


Figure 2. TL-Divide operation

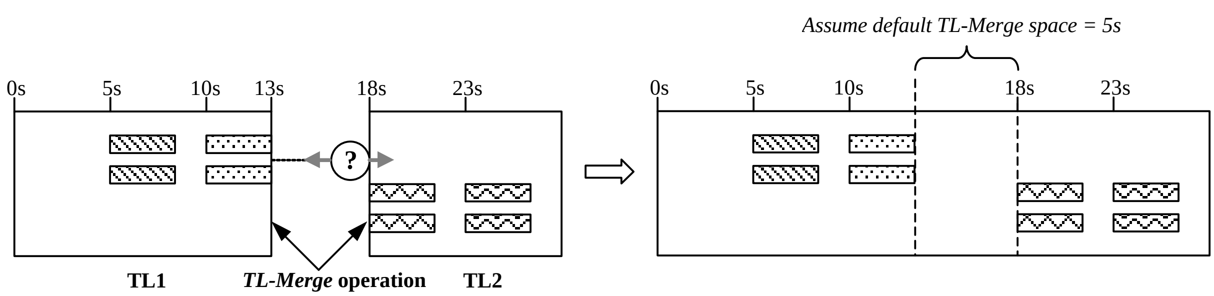


Figure 3. TL-Merge operation

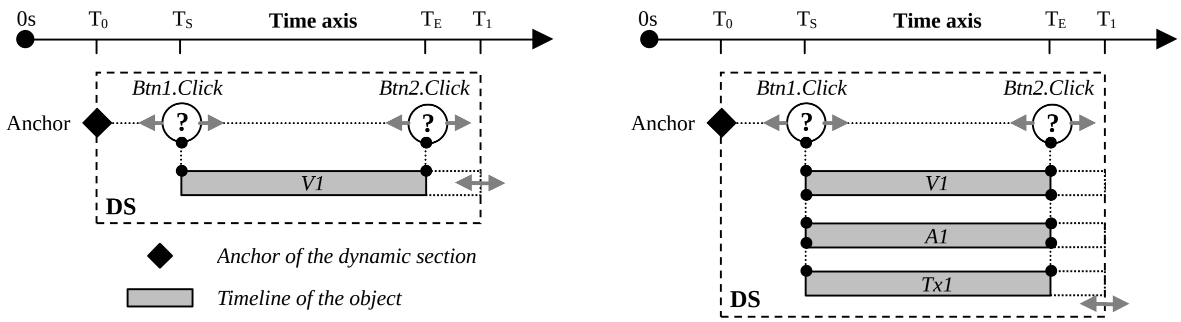
has to specify a proper cutting point on the time axis such that no objects will be cut into two pieces. Moreover, in order to properly display information of the time axis for the latter timeline table, we need to choose a preset (default) value (5s in Figure 2) for the occurrence time of the splitting event. The preset value is only for reference in authoring stage since the exact occurrence time of the event is unknown before run-time. For more flexibility, the authoring system should enable users to change the reference occurrence time of the event as user's wish.

The reverse of TL-Divide is *TL-Merge* operation. The author uses the operation to remove the non-deterministic event and merge two separated timeline tables. Figure 3 gives an example of TL-Merge operation.

2.2. Dynamic section

A dynamic section (DS) is defined for a dynamic object whose beginning and/or ending are triggered by non-splitting events. As shown in Figure 4-(a), to create a DS for an object, the user has to specify the position (anchor) and the length of the DS on the time axis. Two events are associated with beginning and ending of the object in the DS. That is, DS defines the time range of playback for an object, but the actual starting time and ending time of the object are determined until run-time. Thus, the positions of the two events on the time axis are only for reference in authoring stage. As illustrated in Figure 4-(b), DS can also control a group of parallel objects such that the group acts just like a single object in the DS. The authoring system should provide the option to disable/enable events in a DS for more flexibility.

Given that the author may expect a dynamic object depending on a former dynamic object. We propose a DS operator called *connector* to support dependency between different dynamic sections. As illustrated in Figure 5, beginning of DS2 depends on DS1's ending, and DS1's ending depends on event *Btn2.Click*. By using DS connector, the author can easily create a sequence of dependent dynamic objects.



(a) For a single object **(b) For a group of parallel objects**

Figure 4. Dynamic section in a timeline table

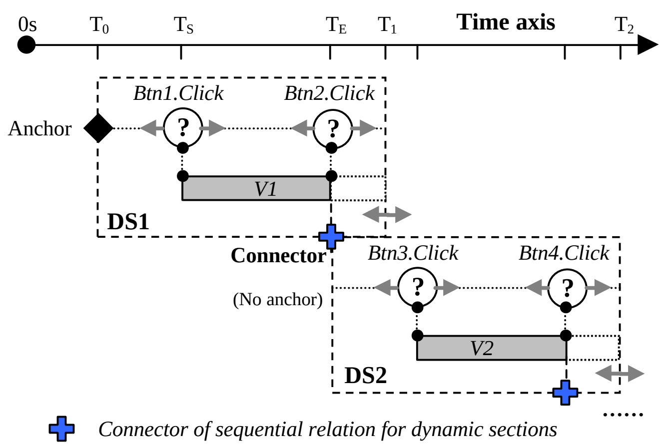


Figure 5. Connecting two dependent DS

3. CONVERTING DDTL TO SMIL2.0

The editing result of DDTL needs to be converted to SMIL2.0 format and saved as a SMIL document for future playback on a SMIL player. We had developed a converting algorithm from deterministic timeline data to SMIL1.0 in our previous work [6]. We only present the converting mechanisms for dividable timeline and dynamic section in this paper, and the mechanisms can be easily integrated into the original converting algorithm in our previous work.

3.1. Converting algorithm for DDTL

Since the two separated timeline tables resulted from TL-Divide operation have a sequential relation, a parent *<seq>* time container is created for the two timeline tables, and the root time container of the latter timeline table must set its *<begin>* attribute the non-deterministic event. Figure 6 shows the code snippet for the result of TL-Divide in Figure 2.

To convert a DS to SMIL2.0, we need to know that there are two run-time cases ending a DS as well as the dynamic object in the DS: (1) the viewer triggers the ending event for the object, or (2) no ending event triggered but the preset ending time of the DS (which is according to the length of the DS) is reached. Therefore, in the conversion, we need to create a parent time container *<par>* for the object and set the *<end>* attribute of the *<par>* to reflect the above two cases. Figure 7-(a) shows the code snippet for the DS in Figure

```
<seq>
     <TL1>
     <TL2 begin="Btn.Click">
</seq>
```

Figure 6. SMIL code snippet for *TL-Divide*

```
<par begin= "T0-0s"end="Btn2.Click; T1-0s">
     <video id=V1 begin="Btn1.Click" >
</par>
  /*** note that DS's time base is 0s ***/
```

(a) Single object in DS

```
<par begin="T0-0s"end="Btn2.Click; T1-0s">
     <par begin="Btn1.Click">
          <video id=V1>
          <audio id=A1>
          <text id=Tx1>
     </par>
</par>
  /*** note that DS's time base is 0s ***/
```

(b) Group of objects in DS

Figure 7. SMIL code snippet for the DS in Fig. 4

4-(a), in which the *<end>* attribute of the parent element *<par>* is a list of two cases to end the DS (i.e. event *Btn2.Click* or time T_1 is reached). The *<begin>* attribute in the parent <par> specifies the location of the DS on the time axis, and the *<begin>* attribute of the object defines the triggered event for starting the object. Similarly, the conversion of the DS containing a group of objects (Figure 4-(b)) is shown in Figure 7-(b).

For a sequence of dynamic sections connected by DS connectors, we need to create a parent *<seq>* for all DS in the sequence. The code snippet for a sequence of DS (Figure 5) is displayed in Figure 8. Please note that the time base of the first DS (DS1, with an anchor) and the time base of the other DS (DS2 and the following) are different, which is reflected in the calculation of the ending time of the root <par> element for each DS.

Due to the limit of paper length, the detailed algorithm for converting DDTL to SMIL2.0 is not presented in the paper. Instead, we give a typical example of the conversion. A sample DDTL editing result is

```
<seq>
  <par id=DS1 begin=T0–0s end="Btn2.Click; T1–0s">
      <video id=V1 begin="Btn1.Click">
  </par>
  <par id=DS2 end="Btn4.Click; T2-TE (DS2 Length)">
      <video id=V2 begin="Btn3.Click">
  </par>
  ……  /* for following DS, if any */
</seq>
  /*** DS1's time base is 0s ***/
  /*** DS2's time base is the end of DS1 ***/
```

Figure 8. SMIL code snippet for the sequence of DS in Figure 5

displayed in Figure 9. The corresponding SMIL2.0 code snippet for that sample is shown in Figure 10.

4. CONCLUSION AND FUTURE WORK

In this paper, we present our first effort towards supporting of SMIL2.0 authoring by timeline-based editing. The concept of *Dividable Dynamic Timeline (DDTL)* is proposed, which includes two novel features, *dividable timeline* and *dynamic section*, to extend the original timeline scheme with the support of non-deterministic temporal behavior. Mechanisms for converting DDTL data to SMIL2.0 format are also proposed in the paper. In summary, DDTL features the easy-learning characteristic of timeline and to some extent allows authors to compose interactive multimedia presentations. The implementation of extending our previous SMIL1.0 authoring system to support DDTL is currently under way.

Apparently, DDTL (and other timeline-based editing schemes) cannot cover the whole set of non-deterministic temporal behaviors supported by SMIL2.0. Thus, in addition to the implementation, we will also explore more about the potential of DDTL in supporting SMIL2.0 authoring. To make authoring more efficient, an authoring system needs to provide reuse of SMIL scripts, which involves the conversion of SMIL2.0 scripts to DDTL. Reuse of SMIL scripts in DDTL-based authoring is addressed in another paper [12].

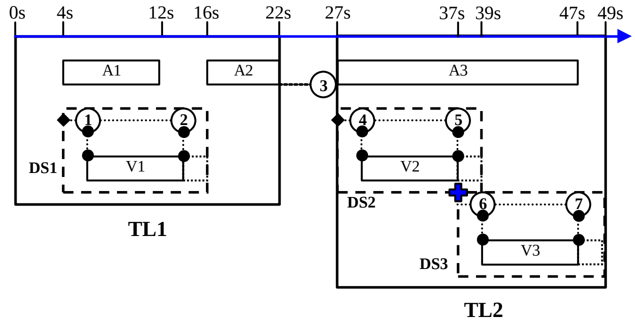


Figure 9. Sample DDTL editing result

```
<seq>
  <par id=TL1>
    <seq>
      <audio id=A1 begin="4s" dur="12s-4s">
      <audio id=A2 begin="16s-12s" dur="22s-16s">
    </seq>
    <par id=DS1 begin="4s", end="Btn2.Click;16s">
      <video id=V1 begin="Btn1.Click">
    </par>
  </par>
  <par id=TL2 begin="Btn3.Click">
    <audio id=A3 begin="0s" dur="47s-27s">
    <seq>
      <par id=DS2 begin="0s" end="Btn5.Click;39s-27s">
           <video id=V2 begin="Btn4.Click">
      </par>
      <par id=DS3 end="Btn7.Click;49s-37s">
           <video id=V3 begin="Btn6.Click">
      </par>
    </seq>
  </par>
</seq>
```

Figure 10. SMIL code snippet for the example in Figure 9

REFERENCES

[1] *Synchronized Multimedia Integration Language (SMIL) 2.0 Specification*, W3C Recommendation, 2001, http://www.w3.org/TR/smil20

[2] D.C.A. Bulterman, "SMIL 2.0 part 1: overview, concepts, and structure," IEEE Multimedia, Volume: 8 Issue: 4, Oct.-Dec. 2001, Page(s): 82 –88

[3] D.C.A. Bulterman, "SMIL 2.0. 2. Examples and comparisons," IEEE Multimedia, Volume: 9 Issue: 1, Jan.-March 2002, Page(s): 74 –84

[4] C. C. Yang, "Design of the Data-Retrieving Engine for Distributed Multimedia Presentations," Proceedings, IEEE International Conference on Communications, 2001 (ICC2001), pp. 3237-3243.

[5] C. C. Yang, "User-Interaction Supported Data- Retrieving Engine for Distributed Multimedia Presentations," Proceedings, IEEE International Conference on Communications, 2001 (ICC2001), pp. 3244-3250.

[6] C. C. Yang and Y. Z. Yang, "SMILAuthor: an Authoring System for SMIL-based Multimedia Presentations," Journal of Multimedia Tools and Applications, vol. 21, no. 3, December 2003, pp. 243-260. http://www.csie.ncnu.edu.tw/~ccyang/Publication/MTAP2003.pdf

[7] C. C. Yang, C. W. Tien, and Y. C. Wang, "Modeling of the Non-deterministic Synchronization Behaviors in SMIL2.0 Documents," Proceedings, IEEE International Conference on Multimedia and Expo (ICME), Vol. 3, 6-9 July 2003, pp. 265-268.

[8] C. C. Yang, C. W. Tien, and Y. C. Wang, "Supporting VCR-like Operations in SMIL2.0 Players," Proceedings, IEEE International Conference on Multimedia and Expo (ICME), Vol. 2, 6-9 July 2003, pp. 761-764.

[9] Yoshihisa Gonno, "SMIL 2.0 Extension for Professional Multimedia Authoring: Preliminary Investigation," W3C Note, 12 May 2003. http://www.w3.org/TR/2003/NOTE-SMIL2-AuthExt-20030512

[10] O. Gaggi and A. Celentano, "A visual authoring environment for prototyping multimedia presentations," Proceedings, 4th International Symposium on Multimedia Software Engineering, 11-13 Dec. 2002, pp. 206 –213.

[11] GRiNS, http://www.oratrix.com/GRiNS/index.html

[12] C. C. Yang, Y. C. Wang, and C. K. Chu, "Reuse of SMIL2.0 scripts in Dividable Dynamic Timeline-based Authoring," also in Proceedings of IEEE International Conference on Multimedia and Expo (ICME), 2004.