Reuse of SMIL2.0 scripts in Dividable Dynamic Timeline-based Authoring

Chun-Chuan Yang, Yung-Chi Wang, and Chen-Kuei Chu

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

Dividable Dynamic Timeline (DDTL) was proposed in our previous work to address the non-deterministic temporal behavior in SMIL2.0 scripts. This paper presents the mechanisms for the reuse of SMIL2.0 scripts in authoring a new presentation. To reuse a SMIL2.0 script, the script must be converted to elements in DDTL. By the reuse of existing SMIL2.0 scripts and the flexible features of DDTL, an efficient and friendly authoring environment for SMIL2.0-based interactive multimedia presentations can be provided.

1. INTRODUCTION

World Wide Web Consortium (W3C) has developed a scripting language for multimedia integration, *Synchronized Multimedia Integrated Language (SMIL)*. With the promotion of W3C and the increasing demand of multimedia over Internet, SMIL has been largely applied for the design of interactive multimedia presentations to be distributed over the web. There are two versions of SMIL specification that had been released by W3C. SMIL version 1 (SMIL1.0) is basically a schedule-based model. The second version of SMIL (SMIL2.0) [1-3] enhances SMIL1.0 by providing a strong support for user interaction with a declarative event-based timing.

An author can write a SMIL presentation using a simple text editor according to SMIL syntax. However, text editor-based method for SMIL authoring is difficult for nonprofessional users. A visualized authoring method is more preferred by casual users. Two major categories for visualized SMIL authoring are (1) structure-based, and (2) timeline-based. 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.

Most of the authoring systems [4, 5] for SMIL2.0 are structure-based, since timeline-based editing is basically for supporting deterministic (scheduled-based) playback time and it is difficult to support the non-deterministic temporal behavior introduced by SMIL2.0, in which 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. To extend timeline-based editing for SMIL2.0 authoring, we proposed a novel concept called *Dividable Dynamic Timeline (DDTL)* [6] that features the easy-learning characteristic of timeline and the ability to allow users composing interactive multimedia presentations with non-deterministic temporal behavior.

This paper is a supplement to the work of DDTL for dealing with the reuse of SMIL2.0 scripts. In order to reuse a SMIL2.0 script in authoring process, the script must be converted to the form of DDTL. There are two steps in the conversion. First, the script is converted to our previously proposed model namely *Extended Real-Time Synchronization Model (E-RTSM)* [7-9], which provides a systematic view for the temporal information in the script. In the second step, the temporal information of each object in the script is extracted by processing E-RTSM and represented in the form of DDTL.

The remainder of the paper is organized as follows. First of all, we make a brief survey for DDTL in section 2. Section 3 gives a typical example for converting SMIL2.0 to E-RTSM. Mechanisms of extracting the temporal information from E-RTSM and converting the temporal information to DDTL are presented in section 4. Finally, section 5 concludes this paper.

2. SURVEY OF DDTL

Two novel features, namely *dividable timeline* and *dynamic section*, are proposed for supporting non-determinism in timeline-based editing [6]. *TL-Divide* operation is defined to enable users to divide a timeline table into two sequentially separated timeline tables and associate a non-deterministic 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 1. Objects in the latter timeline table are not played until the beginning event (*Btn.Click*) of the timeline table is triggered.

A dynamic section (DS) is defined for a dynamic object or a group of objects whose beginning time and ending time are triggered by non-deterministic events. As illustrated in Figure 2, DS defines the time range of playback (T_0 ~ T_1) for object *V1*, but the actual starting time (T_S) and ending time (T_E) of *V1* are triggered by run-time events *Btn1.Click* and *Btn2.Click*.

We also defined *DS connector* to create dependency between dynamic sections. A DS connector connects two dynamic sections such that the starting time of the latter DS depends on ending of the former DS. Figure 3 shows an example for using DS connector to create dependency between two dynamic sections. By using DS connector, the author can easily create a sequence of dependent dynamic objects.

Combining the features of dividable timeline and dynamic section, the author can easily create interactive multimedia presentations with non-deterministic temporal behavior in the timeline-based manner. The overview of the DDTL-based authoring process is displayed in Figure 4.

3. CONVERTING SMIL2.0 TO E-RTSM

E-RTSM and related converting algorithm were proposed in our previous work [8] to provide an efficient modeling technique for SMIL2.0 temporal relationship. In this section, we give a typical example for the conversion from SMIL2.0 to E-RTSM. A sample SMIL2.0 code snippet is shown in Figure 5. The corresponding

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

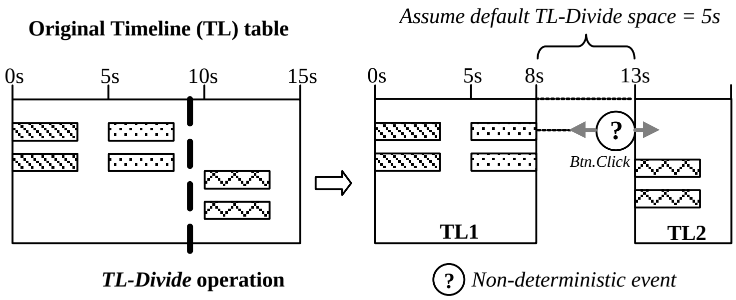


Figure 1. TL-Divide operation

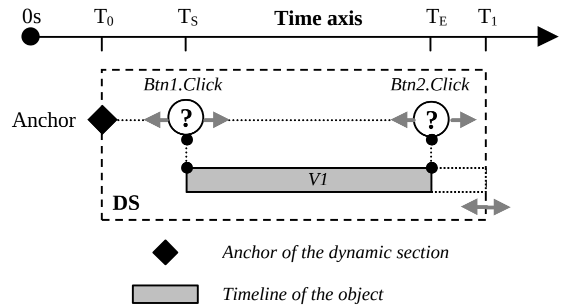


Figure 2. Dynamic section for a single object

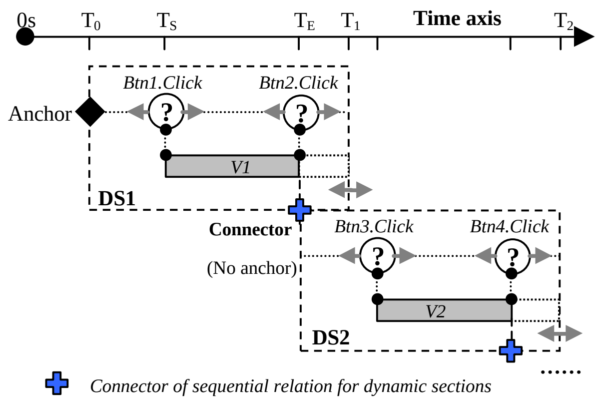


Figure 3. Dependent dynamic sections

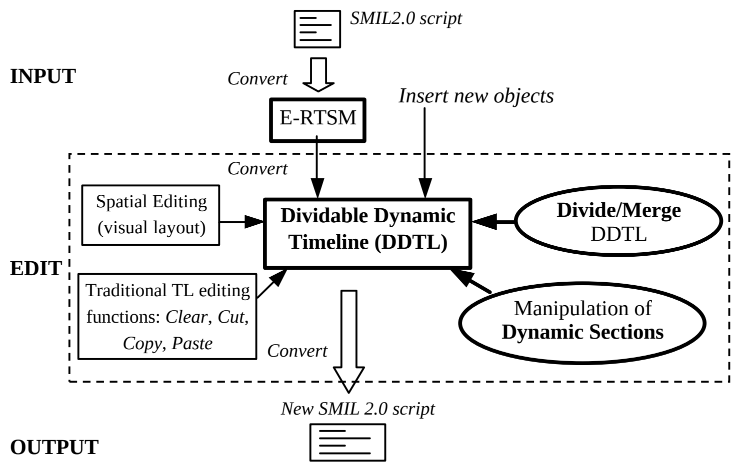


Figure 4. Overview of the DDTL-based authoring

E-RTSM for that sample is displayed in Figure 6. In E-RTSM, an *enforced place* (denote by a double circle) dominates the firing time of its following transition, which means that once an enforced place becomes unblocked (in other words, the related action with the place is completed), the transition following it will be immediately fired regardless the states of other places feeding the same transition. A *virtual place* is a regular place with zero duration. It is quite easy to understand by E-RTSM that the non-deterministic temporal behavior of a SMIL2.0 presentation is introduced by those enforced places mapped to a non-deterministic event (denoted by a double circle with a "?" within).

4. FROM E-RTSM TO DDTL

For an input SMIL script to be reused in timeline-based authoring, the playback time and duration of each object in the script must be determined. 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 our previous work for SMIL1.0 authoring [10], we had developed the mechanisms to calculate the playback time and duration (deterministic values) for each object in an input SMIL1.0 script. However, the previously proposed mechanisms cannot be directly applied in the case of SMIL2.0, since part of the objects in the script may have non-deterministic temporal behavior. Therefore, we propose modified mechanisms to convert E-RTSM to DDTL in this paper.

4.1. Identifying splitting events

As mentioned in section 2, DDTL incorporates the non-deterministic temporal behavior with traditional deterministic timeline segments. Non-deterministic temporal behavior in DDTL comes from the features of dividable timeline and dynamic sections, which can be mapped to two types of non-deterministic events in E-RTSM. An event dividing a timeline table into two separate tables must be a *splitting event* in E-RTSM. A splitting event is a non-deterministic event (an enforced place with a "?") and when removing the event will divide the E-RTSM model into two separated parts.

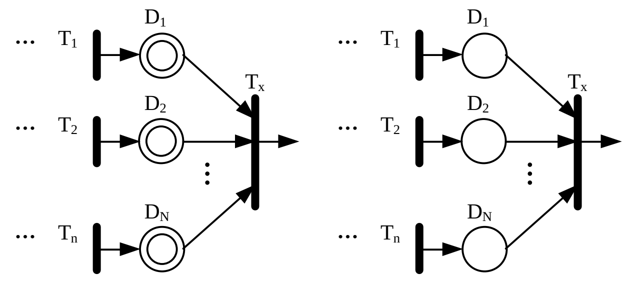
We have to find all splitting events in the E-RTSM model in order to properly identify all divided timeline tables in the input script. The simplest way to decide whether an event is a splitting event or not is to temporally remove the event and check the reachability of the end of the model. If the end of the E-RTSM model is unreachable when removing an event, the event is a splitting event. For example, *Btn3.Click* in Figure 6 is a splitting event, and thus two timeline tables (from E-RTSM-1 and E-RTSM-2 respectively) emerge.

4.2. Identifying dynamic sections

After the sub-E-RTSM models for different timeline tables are determined, the next step is to traverse these sub-models respectively in order to calculate the firing time of each transition. The firing time of the transition right before an object is the starting time of the object, and the firing time of the transition followed by the object is the ending time of object. Since enforced places dominates the firing time of transitions, a sub-E-RTSM model is reduced by removing the regular places that feed into the same transition with one or more enforced places. Figure 7 shows the reduced sub-E-RTSM models for the model in Figure 6.

```
<seq>
  <par id=TL1>
    <seq>
      <audio id=A1 begin="4s" dur="8s">
      <audio id=A2 begin="4s" dur="6s">
    </seq>
    <par id=DS1 begin="4s">
      <video id=V1 begin="Btn1.Click" end="Btn2.Click;12s">
    </par>
  </par>
  <par id=TL2 begin="Btn3.Click">
    <audio id=A3 dur="20s">
    <seq>
      <video id=V2 begin="Btn4.Click" end="Btn5.Click;12s">
      <video id=V3 begin="Btn6.Click" end="Btn7.Click;12s">
    </seq>
  </par>
</seq>
```

Figure 5. Sample SMIL2.0 code snippet



(a) $T_x = Min(T_1+D_1, \ldots, T_n+D_n)$ (b) $T_x = Max(T_1+D_1, \ldots, T_n+D_n)$

Figure. 8. Determine the firing time of transition T_x

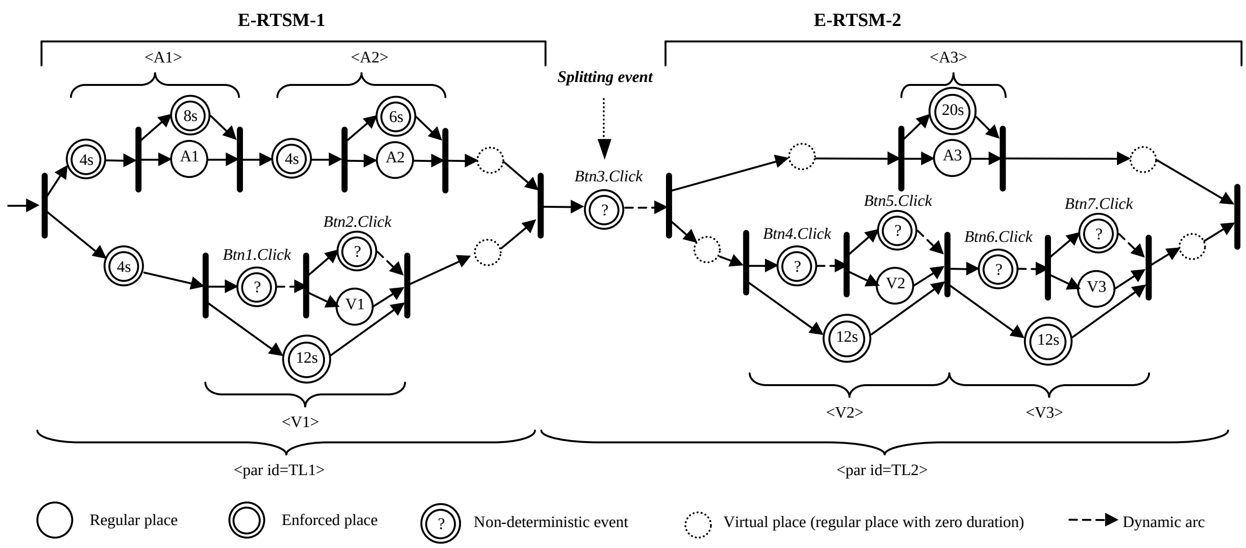


Figure 6. E-RTSM model for the sample SMIL code snippet

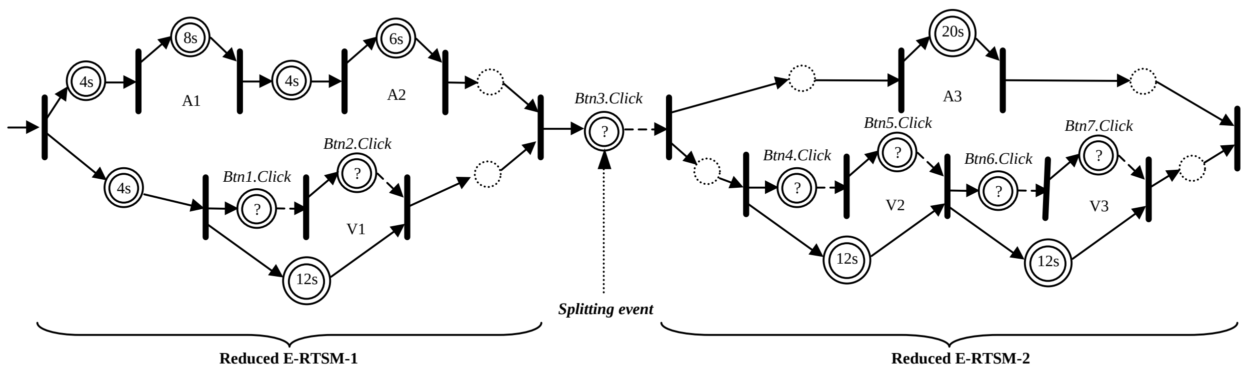


Figure 7. Reduced sub-E-RTSMs of the model in Figure 6

There are only two possibilities for one transition in a reduced E-RTSM model: (1) places that feed to the transition are all enforced places, or (2) places that feed to the transition are all regular places. For possibility (1), the firing time of the transition is the minimal value of "the firing time of the preceding transition" plus "the duration of the following place of the preceding transition", which is illustrated in Figure 8-(a). Figure 8-(b) shows the case of possibility (2), in which transition T_x is fired only after all its preceding regular places finish playing. Therefore, for possibility (2), the firing time of transition T_x is the maximum value of "the firing time of the preceding transition" plus "the duration of the following place of the preceding transition". The duration of each place depends on the type of the media object. For an enforced place of time medium, the duration of the place is the value of the duration. For static media objects, such as <img> and <text>, the duration of the place is zero. For continuous media objects, such as <audio> and <video>, the duration of the place is the implicit duration of the object that is provided by the data server. Since the objects stored in a data server are all pre-orchestrated, it is easy for the data server to obtain the

E-RTSM-1: 0s

Object ID	Beginning time	Ending time
A1	4s	12s
A2	16s	22s
V1	4s + *Btn1.Click*	4s + MIN (“*Btn1.Click+Btn2.Click*”, 12s)

E-RTSM-2: *Btn3.Click*

A3	0s	20s
V2	*Btn4.Click*	MIN (“*Btn4.Click+Btn5.Click*”, 12s)
V3	END (V2) + *Btn6.Click*	END (V2) + MIN (“*Btn6.Click+Btn7.Click*”, 12s)

Figure 9. Beginning and Ending time for each object in Figure 6

implicit duration of a continuous object. However, for a non-deterministic event, the duration of the enforced place is non-deterministic and is represented by a variable in the firing time calculation.

Figure 9 shows the beginning time and ending time of each object in the two sub-E-RTSM models in Figure 7. For those objects (*A1*, *A2*, *A3*) with deterministic beginning time and ending time, a timeline segment is displayed at corresponding position on the time axis in the timeline table as shown in Figure 10. Dynamic sections are used to represent those objects that do not have deterministic beginning time and ending time.

As mentioned in section 2, DS is used to define an object with a beginning event and an ending event. The general mathematical expression for the beginning time of the object in a DS (Figure 2) is T_0+*BeginEvent*, and the ending time of the object, T_0+MIN *(BeginEvent+EndEvent*, *Length of the DS)*. (MIN is the function that returns the smaller one from two given variables/values) Therefore, for those objects with beginning time and ending time in the form of above expressions can be represented by DS. For example, objects *V1* and *V2* are converted to dynamic sections in their respective timeline tables as displayed in Figure 10.

For an object whose playback depends on others is converted to a dependent DS. A dependent DS is connected to a former DS by a DS connector, therefore, the general form of the beginning time and ending time for the object in a dependent is as follows:

Beginning: END (the former DS) + *BeginEvent*

Ending: END (the former DS) +
MIN (*BeginEvent*+*EndEvent*, *Length of the DS*)

For example, object *V3* in Figure 9 has the form of dependent DS (and its former DS is *V2*'s DS). Therefore, a DS connector is created to connect *V3*'s DS (DS3) to its former DS (DS2) as shown in Figure 10.

5. CONCLUSION

This paper presents the mechanisms for the reuse of SMIL2.0 scripts in authoring a new presentation in the timeline-based manner. In order to support temporal non-determinism in SMIL2.0, an extension of timeline scheme namely *Dividable Dynamic Timeline (DDTL)* was proposed in our previous work. To reuse a SMIL2.0 script in DDTL-based editing, the script must be converted to elements in DDTL, which include original timeline segment, dividable timeline, and dynamic section. Two steps are involved in converting a SMIL2.0 script to DDTL: (1) converting the script to our previously proposed model called E-RTSM, and (2) Extracting DDTL elements from E-RTSM. By the reuse of existing SMIL2.0 scripts and the features of DDTL, an efficient intuitive and friendly SMIL2.0 authoring environment can be provided. Based on the SMIL1.0 authoring system [10], implementation of a DDTL-based authoring system for SMIL2.0 is under way.

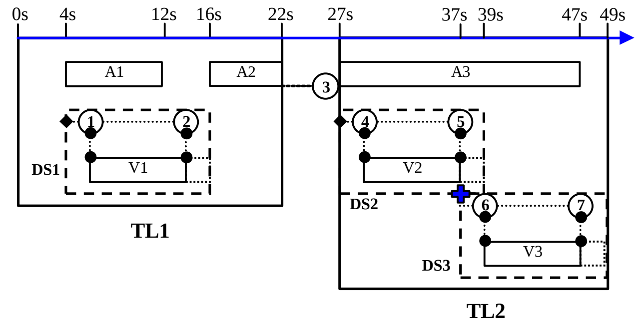


Figure 10. DDTL for the script in Figure 5

Lastly, it's worth mentioning that since DDTL apparently cannot cover the whole set of temporal non-determinism supported by SMIL2.0, there may be some objects that cannot be represented by DDTL elements. Thus, those objects that cannot be DDTL-ized cannot be reused in DDTL-based authoring process.

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

[5] LimSee2, http://wam.inrialpes.fr/software/limsee2/

[6] C. C. Yang, C. K. Chu, and Y. C. Wang, “Dividable Dynamic Timeline-based Authoring for SMIL2.0 presentations,” also in Proceedings of IEEE International Conference on Multimedia and Expo (ICME), 2004.

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

[8] 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.

[9] 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.

[10] C. C. Yang and Y. Z. Yang, “SMILAuthor: and 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