DETECTION OF THE TIME CONFLICTS FOR SMIL-BASED MULTIMEDIA PRESENTATIONS

Chun-Chuan Yang

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

In this paper, the mechanism for detecting the time conflicts for SMIL-based multimedia presentations is proposed. The time conflict for SMIL, which is not the syntax error but the semantic error, is defined as the conflict of the attribute values in the SMIL synchronization elements. In addition to the examination of the syntax error, it will be helpful to provide the mechanism of detecting the time conflict in a SMIL presentation while authoring the multimedia presentation. In order to deal with the temporal relationship among the elements efficiently, the SMIL script is first converted to a synchronization model named RTSM (Real-time Synchronization Model). By traversing the RTSM, the firing time (active time) of each transition in the RTSM model could be determined. The case of inconsistency that the firing time of a transition is later than that of its following transition results in a time conflict. The corresponding elements for the pair of conflicting transitions that result in the time conflict are therefore identified.

1. INTRODUCTION

With the development of multimedia technologies over *Internet* and *WWW*, *distributed multimedia presentations* [1-3] are getting popular for web users. Distributed multimedia presentation is concerning with viewing a presentation via the network. Some or all the media objects could be somewhere else in the network instead of the local site. Multimedia systems for distance learning [4-6] also belong to the class of distributed multimedia presentations. The format of the presentation plays a significant role for multimedia systems, since it affects the compatibility of the presentation. Unfortunately, different multimedia presentation systems always have different formats for the presentation. However, it is better to adopt a popular language as the format of multimedia presentations.

From the popularity point of view, HTML seems to be the best candidate. However, the lack of the ability in integrating synchronized multimedia for HTML makes it improper to be the language of multimedia presentations. *Synchronized Multimedia Integration Language (SMIL)* [10-13] was developed by the *WWW Consortium* (*W3C*) to address the lack of HTML for multimedia over WWW. It provides an easy way to compose multimedia presentations. With the efforts of W3C, SMIL is becoming the most popular language in authoring multimedia presentations and it is going to be supported by the newest versions of the browsers. This paper thus focuses on the SMIL-based presentations.

SMIL allows the authors to use a text editor to write the script for a multimedia presentation. In authoring a multimedia presentation, the author always wants to assure the correctness of the SMIL script, not only in the *syntax level* but also in the *semantic level*. Therefore, the mechanisms for detecting the errors in an SMIL script are necessary. The syntax error for the script of a multimedia presentation could be easily detected according to the *DTD (Document Type Definition)* of standard SMIL specification [10]. However, there is no easy way to detect the semantic error since the semantic error is concerning with the contents (physical meanings) of the SMIL elements with associated attribute values instead of the grammar. If there is a mechanism to detect the semantic error for an SMIL script, it will be helpful for authoring a multimedia presentation by either a text editor or an authoring system [7-9] with the graphical user interface.

One type of the semantic error named the *time conflict* is defined in this paper. The author could describe the temporal relationship among the objects of a presentation by adopting the synchronization elements in SMIL such as <*par*>, <*seq*> and media object elements with proper values of attributes. The time conflict is defined as the case of conflicting values of attributes in the SMIL script. A simple example of the SMIL script with the time conflict is shown in Figure 1, in which only the temporal information is displayed. The <par> element in the figure requires that "*audio1.wav*" and "*video1.mpg*" be played out concurrently and sets a limit of 10 seconds on the playback duration by the "dur" (duration) attribute. However, the values of the "dur" attribute for the audio element and the

```
<par  dur=10s>
    <audio  src= audio1.wav      dur=20s>
    <video  src=video1.mpg     dur=20s>
</par>
```

Figure 1. A sample SMIL script with the time conflict.

video element require the two elements be played out for 20 seconds, which is actually a time conflict with the requirement of their parent <par> element. The example implies that the time conflicts exist in a script without any syntax error. In this paper, the mechanism for detecting the time conflict is proposed.

The time conflict is concerning with the conflict of the *synchronization relationship* (*temporal relationship*) among the elements in SMIL. In order to deal with the synchronization relationship efficiently, a synchronization model [14-19], which provides a systematic view of the synchronization relationship specified in the SMIL script, is necessary for detecting the time conflict. A Petri-net based model namely *OCPN (Object Composition Petri Net)* was proposed to model the synchronization relationship among media objects [14]. However, we had shown that OCPN is not suitable for real-time network applications, and instead the *Real-time Synchronization Model (RTSM)* was proposed [19]. Moreover, some synchronization behaviors that could be specified in SMIL make RTSM more suitable than OCPN. For example, SMIL allows the author to set the explicit beginning time, duration, or end time for a media object. RTSM could easily model the synchronization behaviors as will be explained in section III, but OCPN could not. After modeling the synchronization relationship of the SMIL script by RTSM, the RTSM model is then traversed to detect the time conflicts.

The paper is organized as follows. First of all, we make a survey of SMIL and RTSM in section II. The conversion algorithm for SMIL to RTSM is presented in section III. The method for detecting the time conflict in a SMIL script is explained in section IV. Finally, section V concludes this paper.

2. SURVEY OF SMIL AND RTSM

2.1 Synchronized Multimedia Integration Language

The *Synchronized Multimedia Integration Language (SMIL)* could be used to describe both the spatial relationship and temporal relationship of a multimedia presentation. The spatial relationship is concerning with the visual layout of media objects in the presentation, while the temporal relationship is concerning with the timing control of media objects. The elements for spatial relationship in SMIL include the <*layout*> element and the <*region*> element. The <layout> element determines how the elements in the document's body are positioned. The <region> element controls the position, the size, and scaling of media object elements.

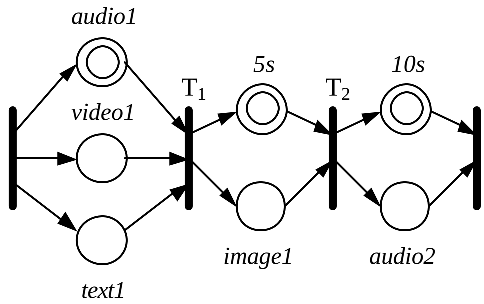


Figure 2. An example of RTSM

The synchronization elements in SMIL for temporal relationship include the <*seq*> element, the <*par*> element, and the class of media object elements such as <*img*>, <*video*>, <*audio*> and <*text*>, etc. The <seq> element defines a sequence of elements in which elements play one after the other. The <par> element defines a simple parallel time grouping in which multiple elements can play back at the same time. Both <seq> and <par> allow the nested structure that means the children element of them could be any of the synchronization elements. The media object elements allow the inclusion of media objects into an SMIL presentation. Media objects are included by reference (using a *URI: Uniform Resource Identifiers*).

Besides, some synchronization related attributes such as "*begin*", "*dur*", and "*end*" could be associated with these synchronization elements. The "begin" attribute specifies the time for the explicit begin of an element. The "end" attribute specifies the explicit end of an element. The "dur" attribute specifies the explicit duration of an element. Since any temporal relationship among multimedia objects could be represented by the combination of the parallel element and the sequential element with proper attribute values, it is easy to see that SMIL could be used to present all synchronization relationships.

2.2 Real-time Synchronization Model

The elements in RTSM include *place*, *token*, and *transition* as in *Object Composition Petri Net* (*OCPN*) [14]. However, there are two kinds of places in RTSM, *regular places* and *enforced places*. A different firing rule for enforced places is defined. The rule specifies 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.

An example of RTSM is shown in Figure 2 in which a single circle is for the regular place, a double circle is for the enforced place, and a bar is drawn for the transition. The RTSM in the figure requires that the audio segment *audio1*, the video clip *video1* and the text data *text1* be played simultaneously. Since *audio1* is an enforced place, transition T_1 is fired right after *audio1* is finished, no matter

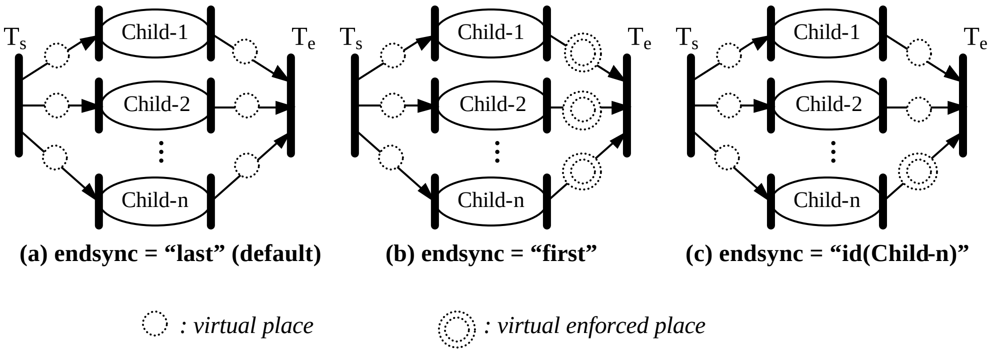


Figure 4. Convert the <par> element to RTSM

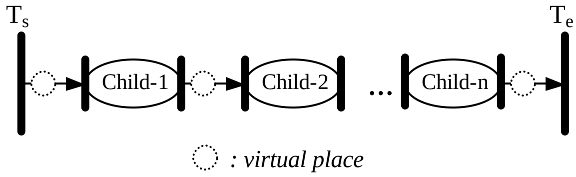


Figure 3. Convert the <seq> element to RTSM

video1 or *text1* is finished or not. After firing T_1, *image1* is displayed for 5 seconds then transition T_2 is fired. Finally, *audio2* is played for 10 seconds after T_2 fires. Note that the enforced place of “5s” in the figure is not a media object but a virtual medium that is called *Time Medium* [19]. The time medium is used to represent time duration.

3. CONVERTING SMIL TO RTSM

In this section, the synchronization elements in SMIL are examined, and the algorithm that converts the elements to RTSM is presented. There are three kinds of synchronization elements in SMIL to be converted: the <seq> element, the <par> element, and the class of media object elements such as <img>, <video>, <audio> and <text>, etc. Besides, some synchronization related attributes such as “begin”, “dur”, and “end” could be associated with these synchronization elements. We assume that the syntax of the input SMIL script has been checked, so the algorithm proposed in this section merely performs necessary conversion.

3.1 Converting the <seq> element

The <seq> element defines a sequence of elements in which elements play one after the other. The children elements of the <seq> element could be any of the synchronization elements such as <seq>, <par>, or the media object elements, so the conversion is actually a recursive procedure. Since the children of a <seq> element form a temporal sequence, we concatenate each child of <seq> one by one in RTSM as illustrated in Figure 3. Note that there are *virtual places* (denoted by the dashed circle) in the figure, which is used to maintain the consistency of RTSM since the arc could only be the link between a transition and a place. In fact, the virtual place is a regular place that maps

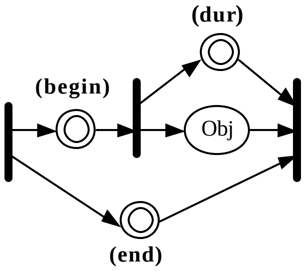


Figure 5. Effect of the attributes on RTSM

to the time medium with zero duration.

3.2 Converting the <par> element

The <par> element defines a simple parallel time grouping in which multiple elements can play back at the same time. Thus, all children of <par> should be within the same pair of transition (T_s, T_e) as illustrated in Figure 4. There are three variations for <par> since a special attribute, “endsync”, could be associated with <par>. The “endsync” attribute controls the end of the <par> element, as a function of children. Legal values for the attribute are “last”, “first”, and “id-ref”.

The value of “last” requires <par> to end with the last end of all the child elements, and the corresponding RTSM is shown in Figure 4-(a), in which transition T_e could not be fired unless all the children end. The value of “first” requires <par> to end with the earliest end of all the child elements. Therefore, we should change the places between each child element and transition T_e to *virtual enforced places* as illustrated in Figure 4-(b) so that the child that ends first will fire transition T_e. A virtual enforced place is an enforced place that maps to the time medium with zero duration. The value of “id-ref” requires <par> to end with the specified child. So we change the place between the specified child and transition T_e to the virtual enforced place as shown in Figure 4-(c).

Other synchronization attributes, such as “begin”, “end’, and “dur”, could also be associated with <seq> and <par>, but the conversion is similar to that in the media object elements that we present in the next sub-section.

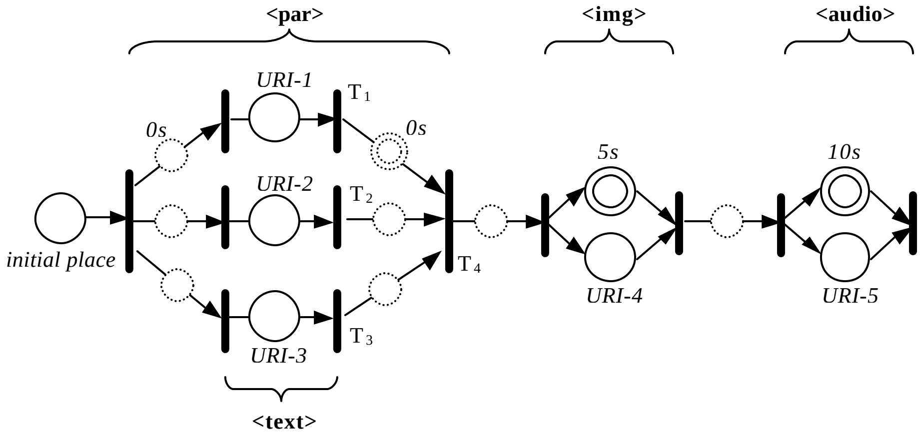


Figure 7. RTSM for the sample SMIL document

```
<seq>
    <par   endsync = id(URI-1)>
           <audio    src=URI-1 />
           <video    src=URI-2 />
           <text     src=URI-3 />
    </par>
    <img    src=URI-4    dur= "5s" />
    <audio  src=URI-5    dur= "10s" />
</seq>
```

Figure 6. Sample SMIL Document

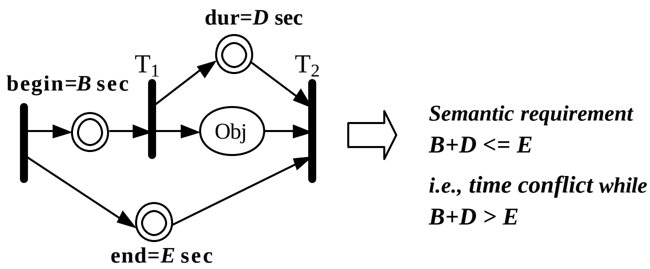


Figure 8. Detecting the intra-element time conflict

3.3 Convert the media object elements

The media object elements allow the inclusion of media objects into a SMIL presentation. Media objects are included by reference (using a *URI*). One media object element naturally represents a regular place in RTSM. However, the attributes associated with the element require some more conversion. Since the "*begin*" attribute specifies the time for the explicit begin of an element, one enforced place representing the begin time with the specified duration is added in front of the element. The "*end*" attribute specifies the explicit end of an element, so one enforced place with the clock value is added between the original start transition and the end transition. The "*dur*" attribute specifies the explicit duration of an element, thus an enforced place between the actual start transition and the end transition is added. We illustrate the effect of these attributes on the element in Figure 5.

3.4 An example for the conversion

Figure 6 shows a sample SMIL script, in which only temporal information is displayed. The SMIL script in Figure 6 requires the player to play the audio object *URI-1*, the video object *URI-2* and text object *URI-3* synchronously since these three objects are contained in a <par> element. The value of the "*endsync*" attribute in the <par> element requires <par> to end with the end of the audio object *URI-1*. In other words, once the audio object *URI-1* finishes playing, the video object *URI-2* and the text object *URI-3* must also stop playing at the same time. After the <par> element, the player has to display the image object *URI-4* for 5 seconds, and then play the audio object *URI-5* for 10 seconds. It is easy to be aware that the synchronization relationship of the SMIL script is similar to that of the RTSM in Figure 2. The obtained RTSM for the sample SMIL document after the converting process is shown in Figure 7.

4. DETECTION OF TIME CONFLICTS

There are two types of time conflicts for SMIL presentations, the *intra-element time conflict* and the *inter-element time conflict*. The intra-element time conflict is the case of conflicting attributes associated with a single element. The inter-element time conflict is the case of conflicting attributes among different elements. Therefore, to detect the intra-element conflict, we only need to examine the values of attributes associated with a single element. Considering the RTSM model for a single element in Figure 8, the values of "*begin*", "*dur*", and "*end*" for the element are "*B* seconds", "*D* seconds", and "*E* seconds" respectively. It is easy to see that the case of $B+D > E$ results in the unreasonable physical meaning, and it is

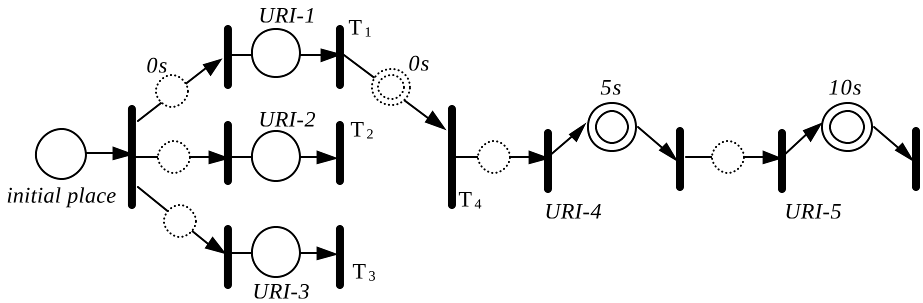
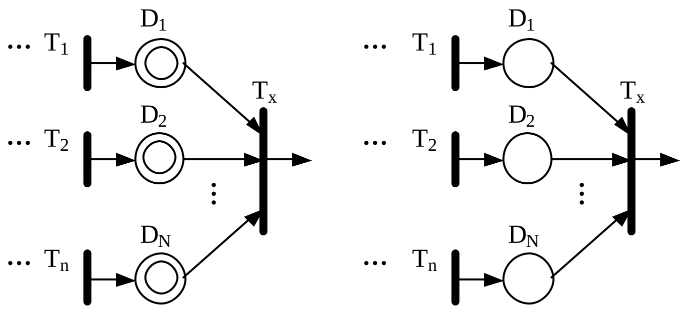


Figure 9. Reduced RTSM for the example in Figure 7



(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 10. Determine the firing time for transition T_x

identified as the intra-element time conflict.

On the other hand, there is no easy way to detect the inter-element time conflict, since it involves the attributes of more than one element. Reconsidering the RTSM in Figure 8 from the transition point of view, the firing time of transition T_1 should be earlier than that of T_2, since T_2 follows T_1. Thus, the case of B > E results in a time conflict. It implies that if the pre-assigned values of attributes by the author make the firing time of a transition later than the firing time of some following transition, it results in the inter-element time conflict. According to the firing rules of RTSM, the inter-element time conflict is solved by the backtracking rule of RTSM when the transition with the enforced place is fired at the run-time [19]. However, the backtracking rule could not provide any help in the authoring time of the presentation. Therefore, we need a mechanism to detect the inter-element time conflict before the run-time.

Since conflicting firing times of transitions result in the inter-element time conflict as mentioned above, the RTSM for the SMIL script is traversed and the firing time of each transition is computed. The inter-element time conflict could be detected by comparing the computed firing time of transitions. The algorithm to calculate the firing time of each transition in RTSM is presented in the following.

4.1 Determining the firing time of transition in RTSM

To compute the firing time for each transition, we have to follow the progress of RTSM. Since there is usually more than one place that feeds to a transition, the behavior of the transition depends on the type of places that feed into it. If a transition is fed by some enforced places, the enforced places will dominate the behavior of the transition. In other words, if a transition is fed by some enforced places, other regular places cannot affect the firing time of the transition at all. Thus, we reduce the RTSM model by removing the regular places that feed to a transition with enforced places. The reduced RTSM for the example in Figure 7 is shown in Figure 9.

The firing time for each transition is computed by traversing the reduced RTSM transition by transition from the initial place (i.e. the start of the presentation). There are only two possibilities for one transition in the reduced RTSM: (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 10-(a). Figure 10-(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 implicit duration of a continuous object. Assuming that the implicit duration for *URI-1* in Figure 9 is 10 seconds, the playback time for each object in the example of Figure 9 is shown in Figure 11.

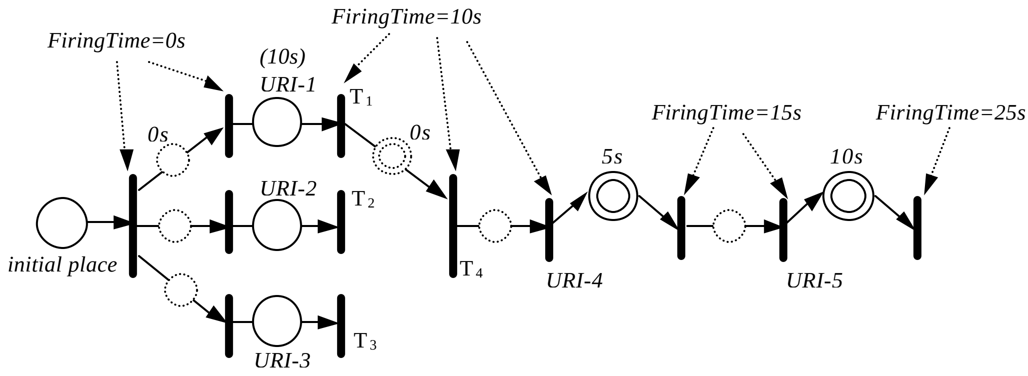


Figure 11. Computing the firing time for each transition

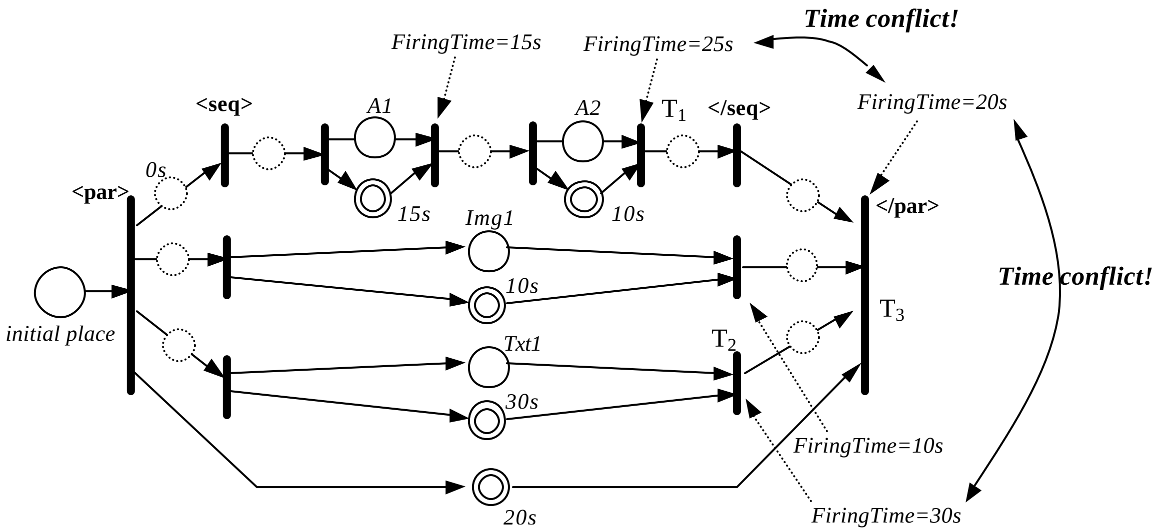


Figure 13. The time conflict of the sample SMIL scrip in Figure 12

```
<par   dur="20s" >
    <seq>
        <audio    src= A1    dur="15s" />
        <audio    src= A2    dur="10s" />
    </seq>
    <img     src=Img1     dur= "10s" />
    <text    src=Txt1     dur= "30s" />
</par>
```

Figure 12. Sample SMIL Document with the inter-element time conflict

4.2 Detecting the inter-element time conflict

As mentioned above, the inter-element time conflict could be identified as the case that the firing time of a transition later than the firing time of the following transition. Therefore, when the firing time of each transition is computed, we traverse the RTSM again from the initial place and located the pair of transitions at which the conflicting firing times occur. The corresponding pair of the elements with the conflicting transitions is therefore the source of the inter-element time conflict.

Figure 12 shows an example of the SMIL script with the inter-element time conflict, and RTSM for the example is shown in Figure 13. As illustrated in Figure 13, the firing time for T_1 and T_2 are 25 seconds and 30 seconds respectively. However, the firing time of the following transition T_3 is 20 seconds. It results two pairs of conflict transitions, (T_1, T_3) and (T_2, T_3). It implies that the corresponding pairs of elements, i.e. (A2, <par>) and (Txt1, <par>), result in the inter-element time conflict.

5. CONCLUSION

The *time conflict*, a semantic-level error of SMIL-based multimedia presentations, is defined in this paper. The reason of the time conflict is the conflicting values of attributes in the elements of a SMIL script, and it results in an unreasonable physical meaning. There are two kinds of time conflict in an SMIL file: the *intra-element time* *conflict* and the *inter-element time conflict*. The intra-element time conflict is the case of conflicting attributes associated with a single element. The inter-element time conflict is the case of conflicting attributes among different elements. The intra-element time conflict is easy to detect since only the attributes of a single element are examined.

On the other hands, detection of the inter-element time conflict requires more processing on the temporal relationship of elements than that of the intra-element time conflict. In order to manipulate the synchronization relationship of the elements in an SMIL script efficiently, the synchronization relationship of the SMIL script is first converted to RTSM. The firing time of each transition is then computed by traversing the obtained RTSM. The inter-element time conflict is identified as the case of conflicting firing times of the transitions, in which the firing time of a transition later than the firing time of the following transition. Therefore, the pair of elements correspond to the pair of conflicting transitions is identified as the source of the inter-element time conflict.

Reference

[1] Taeil Jeong, Jae Wook Ham, and Sung Jo Kim, "A pre-scheduling mechanism for multimedia presentation synchronization," Proceedings., IEEE International Conference on Multimedia Computing and Systems '97, 1997, pp. 379-386.

[2] R. Paul, M. F. Khan, S. Baqai, and A. Ghafoor, "Real-time scheduling for synchronized presentation of multimedia information in distributed multimedia systems," Proceedings., Third International Workshop on Object-Oriented Real-Time Dependable Systems, 1997, pp. 177-184.

[3] De Lima, R. M. et al., "SAMM: An integrated Environment to Support Multimedia Synchronization of Pre-orchestrated Data," Proceedings., IEEE International Conference on Multimedia Computing and Systems, 1999, pp. 929-933.

[4] M. P. Hui and S. L. Hung, "Using distributed hypermedia educational systems for distance education," Proc. IEEE International Conference on Multimedia Engineering Education, pp. 135-140, 1996.

[5] H.Y. Chen and G.Y. Chen, "Design of a Web-based Synchronized Multimedia Lecture System," Proc. IEEE International Conference on Multimedia Computing and Systems, Vol. 2, pp. 887-891, 1999.

[6] H.Y. Chen, Y.T. Chia, G.Y. Chen, and J.S. Hong, "RTP-based Synchronized Hypermedia Live Lecture System for Distance Education," Proc. ACM Multimedia Conference (Multimedia'99), pp. 91-100, 1999.

[7] J. Kelner, D. Hadj Sadok, F. Marques, and A.Neves, "The role of parametrization in the multimedia authoring process," Proceedings., IEEE Conference on Protocols for Multimedia Systems - Multimedia Networking, 1997, pp. 142 –149.

[8] M. Jourdan, N. Layaïda, C. Roisin, L. S.-Ismaïl, and L. Tardif, "Madeus, and authoring environment for interactive multimedia documents," Proceedings., 6th ACM international conference on Multimedia, 1998, pp. 267 – 272.

[9] M. Jourdan, C. Roisin, and L. Tardif, "Multiviews interfaces for multimedia authoring environments," Proceedings., Multimedia Modeling, 1998 (MMM '98), pp. 72 –79.

[10] *Synchronized Multimedia Integration Language (SMIL) 1.0 Specification*, W3C Recommendation, June 1998, http://www.w3c.org/TR/REC-smil.

[11] *Synchronized Multimedia Integration Language (SMIL) Boston Specification*, W3C Working Draft 20-August-1999, http://www.w3.org/TR/smil-boston.

[12] Larry Bouthillier, "Synchronized Multimedia on the Web- A New W3C Format is All Smiles," Web Techniques Magazine, September 1998, Vol. 3 Issue 9.

[13] Philipp Hoschka, "An introduction to the Synchronized Multimedia Integration Language," IEEE Multimedia, Oct.-Dec. 1998, pp. 84-88.

[14] T. D. C. Little, and A. Ghafoor, "Synchronization and storage models for multimedia objects," IEEE Journal of Selected Area in Communications, vol. 8, no. 3, pp. 413-427, Apr. 1989.

[15] Chung-Ming Huang and Chian Wang, "Synchronization for interactive multimedia presentations," IEEE Multimedia, Oct.-Dec. 1998, pp. 44–62.

[16] InHo Lin, Chwan-Chia Wu, and Bih-Hwang Lee, "A synchronization model for multimedia presentation with critical overlap avoidance," Proceedings. International Workshop on Multimedia Software Engineering, 1998, pp. 36-43.

[17] I. F. Cruz and P. S. Mahalley, "Temporal Synchronization in Multimedia Presentations," Proceedings., IEEE International Conference on Multimedia Computing and Systems, 1999, pp. 851-856.

[18] Z. Yang, et al., "A New Look at Multimedia Synchronization in Distributed Environments," Proceedings., 4th International Symposium on Parallel Architectures, Algorithms, and Networks, (I-SPAN '99), 1999, pp. 322 –327.

[19] C. C. Yang and J. H. Huang, "A Multimedia Synchronization Model and its Implementation in Transport protocols," IEEE Journal of Selected Area in Communications, vol. 14, No. 1, pp. 212-225, Jan. 1996.