A Collaborative Multimedia Authoring System Based on the Conceptual Temporal Relations

Mee Young Sung

Department of Computer Science & Engineering, University of Incheon
177 Dohwadong, Namgu, 402-749 Incheon, South Korea
mysung@incheon.ac.kr

Abstract. We developed a SMIL-based collaborative multimedia authoring tool supporting a mechanism for conceptually representing the temporal relations between different media. Among the many editors that make up our system, the temporal relation editor provides users with an intuitive mechanism for representing the conceptual flow of a presentation by simple and direct graphical manipulations. Our system proposes TRN (Temporal Relation Network) as its internal multimedia presentation representation. The TRN corresponds exactly to the conceptual temporal structure of the multimedia presentation. A TRN is composed of media objects, delay objects and a set of temporal relationships among objects. A media object is associated with a duration. A parallel relationship found in a TRN can be collapsed into a single par (parallel) synchronization block. This collapsible synchronization block facilitates the determination of the playing time of each component and can be the basic unit for reusability of already prepared blocks of presentation code. In addition, our system allows users in different places to design together a multimedia presentation collaboratively in reviewing the same presentation at the same time.

Keywords: Collaborative authoring, Multimedia authoring, SMIL (Synchronized Multimedia Integration Language), Temporal relation representation, Synchronization.

1 Introduction

The key to authoring a presentation lies in the composition of temporal relationships between objects. Conceptually, the temporal relationships between two media can be classified into one of seven possibilities. They are 'before', 'meets', 'overlap', 'during', 'starts', 'finishes', and 'equals' [1][2]. Every temporal relationship can be described using one of these seven relations. Spatial relationships can be described by specifying sub-regions within the total presentation region that correspond to each object.

The goal of this study is to develop an easy and efficient multimedia authoring environment where users can create a multimedia presentation in a simple and intuitive manner. Toward this goal, we provide users with the capability to

K. Aizawa, Y. Nakamura, and S. Satoh (Eds.): PCM 2004, LNCS 3332, pp. 649–656, 2004.

edit temporal relationships between media objects at the conceptual level: for example, presenting object A before B, presenting object A during B, etc. We also want to allow users to create multimedia content without manually specifying the playing time (e.g. a specific start time and duration) for each media. Instead, our authoring system automatically calculates the playing time and then generates proper start times and durations for each object. In the traditional scaled timeline approach, users can directly view and control the structure of the content; however, the representation is fixed, and the operations are manual. Our goal was to develop a good tool for generating the presentation schedules conceptually without considering the details, and the system can automatically detail the properties of the media. Using our system, users can focus on the creative aspects of their design, and not worry about manual specification of timing details for each object.

We developed a multimedia authoring system based on the SMIL (Synchronized Multimedia Integration Language) [3][4][5][6] 1.0 Recommendation[3] and SMIL 2.0 Recommendation [4][7][8]. The existing SMIL authoring tools provide basic user interfaces such as the scaled timeline-based user interfaces (representing media objects as different bars arranged in multiple layers on the scaled timeline) or textual tag editing user interfaces for authoring. What distinguishes our system is that it provides a simple and intuitive editing mechanism for creating conceptual flows of a presentation, in addition to the basic timeline-based interface.

In this paper, we present the design and implementation of our multimedia presentation authoring system which provides a mechanism for conceptually representing the temporal relations of different media. We will examine our mechanism for representing conceptual temporal relationships in the following section. In section 3, we will investigate the Temporal Relation Network (TRN) upon which our model is based. We will discuss our algorithm for automatically generating a TRN from the DOM in section 4. In section 5, the implementation of our collaborative authoring is presented. Finally, the last section will provide conclusions and some future work.

2 Representation of Conceptual Temporal Relations

A main focus in authoring a multimedia presentation is the design of the temporal behaviors for the components that make up the presentation. Our system is designed to allow users to specify temporal behaviors of media objects at the conceptual level. This section describes our model for representing conceptual temporal relations.

Our system's multimedia representation is based on Allen's temporal intervals [1]. Allen distinguished thirteen different time intervals between two objects. They can be reduced into seven temporal relationships such as 'before', 'meets', 'overlap', 'during', 'starts', 'finishes', and 'equals' by removing the relationships in inverse order. The graphical representations of the seven conceptual temporal relations of our system are summarized in Figure 1. The graphical representations shown in Figure 1 correspond exactly with the internal representation of each corresponding temporal relationship. Note that we represent the parallel relationships such as overlaps, during, starts, and finishes) by adding dummy delay objects to make the 'equal' relationship. As shown in Figure 1, all five parallel relations can be generalized as the 'equal' relation by inserting some delay objects when they are needed. As a consequence, any parallel relation can be collapsed into a single object. We simplify a group of networked icons as a synchronization block object. This block object can be opened to how the details or collapsed to a single block icon as shown in Figure 2. This mechanism is proposed to simplify the representation of complicated parallel relationships within SMIL content.

3 TRN (Temporal Relation Network)

The authoring process is composed of a series of user interactions for editing a multimedia presentation. An interactive authoring system should process each user interaction immediately and return appropriate feedback. Supporting an interactive authoring environment requires consistent internal maintenance of the state of the presentation [9][10][11][12]. Some existing studies on the internal representation of multimedia applications include: OCPN (Object Composition Petri Nets) [2], DTPN (Dynamic Timed Petri Nets) [9], XOCPN (eXtended OCPN) [10][11], etc. DTPN and XOCPN are variants of OCPN. In these systems, the internal multimedia representations are based on the Petri Net, and the interface is a scaled timeline-based UI. Our system uses TRN (Temporal Relation Network) as its internal representation of a multimedia presentation in order to unify the internal representation and the external user interface with respect to temporal relationships. TRN is a directed and weighted graph. Details are described in the reference [13].

As mentioned before, our system is based on SMIL, whose grammatical structure is the same as that of XML (eXtensible Markup Language). SMIL documents, like any other XML-based document, can be described as a Document Object Model (DOM). The objective of DOM is to provide a standardized interface for accessing XML-based documents (such as XML, SMIL, WML; Wireless Markup Language, SVG; Scalable Vector Graphic, etc.) in diverse computing environments. DOM specifies how to describe the logical structure of XML documents and the details of the components that they contain. DOM describes the logical relationships of document components in a tree structure; however, DOM cannot effectively describe the temporal relationships of components in its simple tree structure. Therefore, we need another mechanism to describe the temporal relationships among different media.

Our system represents internally, as well as graphically, the temporal relationships of a presentation in the TRN graph structure. TRN represents temporal relationships among objects and playing times of objects using information from all objects included in document. If the presentation is new, a new TRN is created. If the presentation already exists, the corresponding DOM structure of the

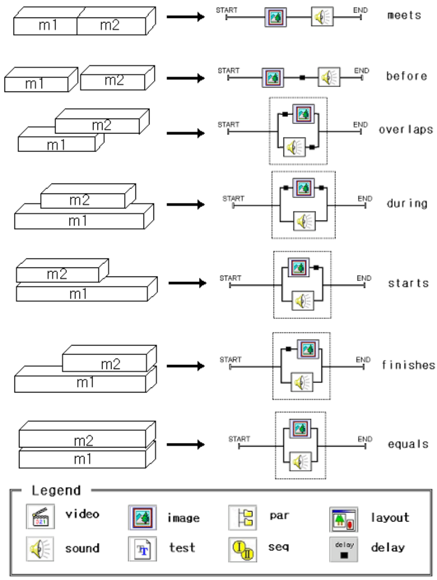


Fig. 1. Representation of temporal relations

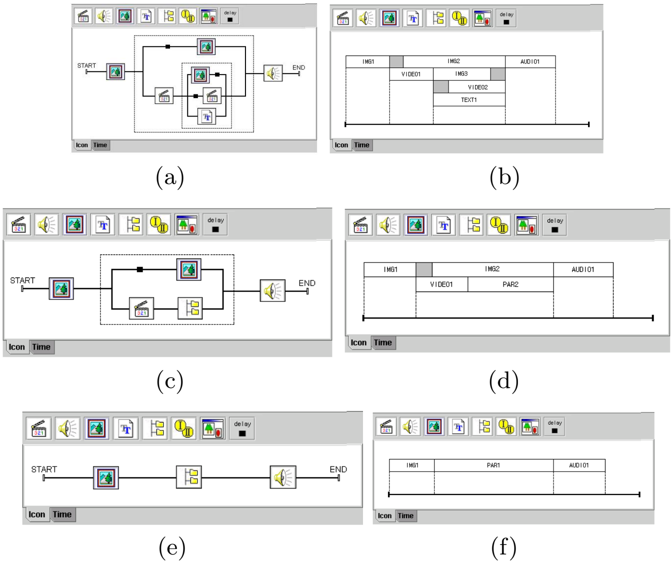
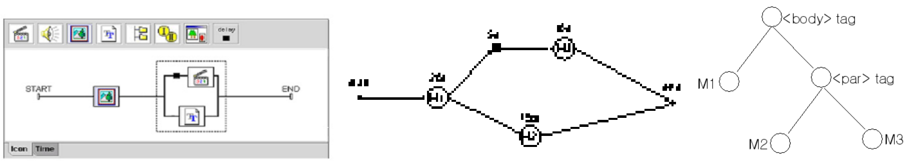


Fig. 2. An example of collapsing some nested parallel synchronization blocks and the corresponding timeline representation: (a), (c), (e) are conceptual representations and (b), (d), (f) are the corresponding timeline representations



(a) (b) (c)

Fig. 3. (a) An example of graphical representation of a multimedia presentation, (b) TRN generated from the DOM information, (c) DOM structure generated from the corresponding TRN.

presentation is reconstructed from its SMIL codes and automatically transformed into the internal TRN structure. As authoring is performed, the underlying TRN must be dynamically changed. After the authoring is finished, a DOM structure can be generated from the internal TRN structure. Our system generates SMIL documents through the interaction between TRN and DOM.

Figure 3(b) illustrates an example TRN that is created as a user authors the presentation shown in Figure 3(a). A node contains more than one outgoing arrow to indicate that the following objects will be played synchronously.

Our system generates the DOM structure in Figure 3(c) and the following SMIL codes from the information of TRN in Figure 3(b).

```
<seq>
    <video id=M1 src=media1.mpg dur=20s/>
    <par>
    <audio id=M3 src=audio.wav begin=5s dur=10s/>
    <img id=M2 src=image.jpg dur=15s/>
    </par>
</seq>
```

Note that there is a direct correspondence between the internal TRN representation and the graphical representation used for authoring a presentation. Therefore, for efficiency, we can collapse parallel relationships found in a TRN into a single node just as we collapsed parallel objects into a single synchronization block as described in section 2.2. This simplification is made through maximized use of the temporal relation 'equals'. It is important to note that this simplification impacts the algorithms used to calculate the actual values of 'dur', 'begin', and 'end' tags when the SMIL code is generated. Using collapsible parallel objects, our system can easily determine the playing time of each component by first considering the group of parallel relations as a single object. This simplification also impacts the algorithms used to implement a SMIL player, because the SMIL player uses the same TRN when it actually schedules the presentation. In addition, this synchronization block representation will be the basic unit for reusability of SMIL code.

4 Automatic Generation of a TRN from the DOM

The algorithm for automatically generating a TRN from the DOM primarily consists of three modules. They are *build_TRN*(), *insertSeqNode*(), and *insertPar*().

The *build_TRN*() function actually takes charge of traversing over all the nodes of the document structure. Each component module in this algorithm includes all of the methods required to allow direct or sequential traversal of the document structure, e.g. *getNextSibling*(), *getChildNode*(), *getChildNodes*(), *getParentNode*(), etc. The *insertSeqNode*() routine creates a media node and inserts it into the TRN using the attributes specified as arguments. An additional delay object is automatically created and inserted into the TRN if it is needed. In the *insertSeqNode*() module, the temporal relation ‘meets’ or ‘before’ can be determined by whether or not a delay object exists between the current object and the preceding object. The module *insertPar*() for handling parallel relationships (such as ‘equals’, ‘starts’, ‘finishes’, ‘during’, and ‘overlaps’) of objects. Any parallel relation can be collapsed into a single object. We call a group of networked objects in parallel relationships as a parallel block. The *insertPar*() performs the required tasks as follows:

1. Determine the number of child nodes of the parallel block.
2. Calculate the total playing time of a parallel block.
3. Determine the temporal relationships between each child object and the parallel block.
4. If there are only two child nodes,insert these objects and determine the temporal relationship from the total playing time and the attributes of each child object. Then the routine is terminated.

Note that *insertPar*() is a recursive algorithm for inserting inner parallel blocks inside a parallel block. We also note that Our algorithm should take time $O(MN)$, where M is the number of attributes and N is the number of nodes in the TRN. The algorithm traverses the document tree in $O(N)$ time and each iteration of traversal invokes *insertObject*() which takes time $O(M)$. In practice, the best algorithm for traversing a tree takes time $O(N)$. The data structure itself should store in $O(N)$ space.

5 Implementation of Collaborative Authoring

Our authoring system allows a group of users working at different machines to work on the same multimedia presentation and to communicate in real time. The collaboration manager of our system takes charge of the communications of all events generated by users. Each authoring system at different places can be a server as well as a client of a collaboration group at the same time. A server generates itself as the first client of the collaboration group. Any client can connect to the server using TCP (Transmission Control Protocol) and generates packets corresponding to the content that is created as users edit

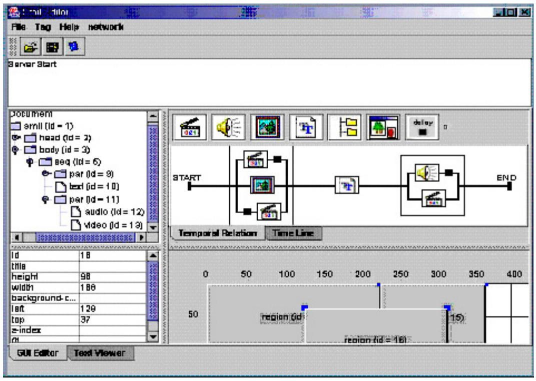


Fig. 4. A screen capture of our authoring system

the presentation. It also receives packets from the server, analyses the packets, and invokes appropriate events or modules. Once a client connects to a server, the server updates the list of groups and initializes the new client by sending a group of objects that have been authored up until that time to the new client. After then, the server multicasts any messages passed to it and the client processes and visualizes any received messages. This mechanism is a variation of a client-server mechanism which can provide better network performance and better portability of the system.

In any collaborative computing environment, multiple users or processes can access a shared object concurrently. In this situation, an inconsistency of shared data might occur therefore a concurrency control is required. We implemented some ideas for efficient concurrency control in our system. They are mainly based on user awareness, multiple versions, and access permissions of shared objects. Details of our concurrency control mechanism are described in the reference [13].

Figure 4 presents a screen capture of our system. Exactly the same images are shown at each user's screen.

6 Conclusion

We developed a SMIL-based collaborative authoring system which allows users to edit the temporal relations among media conceptually by simple and intuitive graphical manipulations. The system editors exchange information through the SOM (SMIL Object Manager) and together form an easy and efficient editing environment. Our authoring system creates and modifies a multimedia presentation using a Temporal Relation Network (TRN) which corresponds exactly to the structure seen in the graphical representation of the presentation. One advantage of our system is that multimedia authors need not specify all of the painstaking details about start times and duration information when creating a presentation. The TRN representation provides an efficient means for the system to automatically fill in the necessary timing details. This frees multimedia authors to focus instead on the creative aspects of the presentation. Another advantage is the use of collapsible synchronization blocks to efficiently represent and specify simultaneous presentation of parallel media. The collapsible synchronization blocks used by our system also provide a means for portable and reusable SMIL code blocks. Note that in addition to their use in SMIL authoring systems, our system editor modules can be used separately in various kinds of multimedia presentation authoring systems, such as XMT (eXtensible MPEG-4 Textual format)-based authoring tool. Because our system makes use of the standard DOM structure, its components can easily be applied by any multimedia system which also uses the DOM structure for its internal document representation.

Acknowledgement. This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Multimedia Research Center at the University of Incheon.

References

1. James F., Allen: Maintaining Knowledge about Temporal Intervals. Communications of the ACM (November 1983) 832–843
2. Thomas D. C. Little, Arif Ghafoor: Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks. IEEE Computer **24**(10) (October 1991) 42–50
3. W3C, Synchronized Multimedia Integration Language (SMIL) 1.0 Specification: W3C Recommendation 15-June-1998, http://www.w3.org/TR/REC-smil/ (1998)
4. W3C,Synchronized Multimedia Integration Language (SMIL 2.0): W3C Recommendation 07 August 2001, http://www.w3.org/TR/smil20/ (2001)
5. W3C, W3C Issues SMIL as a Proposed Recommendation, http://www.w3.org/Press/1998/SMIL-PR (1998)
6. W3C, Synchronized Multimedia, http://www.w3c.org/AudioVideo
7. Dick C.A. Bulterman: SMIL 2.0 Part1: Overview, Concepts and Structure. IEEE Multimedia **8**(4) (October-December 2002) 82–88
8. Dick C.A. Bulterman: SMIL 2.0 Part2: Examples and Comparisons. IEEE Multimedia **9**(1) (January-March 2002) 74–84
9. Mitsutoshi Iino, Young Francis Day, and Arif Ghafoor: An Object-Oriented Model for Spatio-Temporal Synchronization of Multimedia Information. IEEE International Conference on Multimedia Computing and Systems, May 14-19, 1994, Boston, Massachusetts (1994) 110–119
10. B. Prabhakaran and S. V. Raghavan: Synchronization Models For Multimedia Presentation With User Participation. Proceedings on ACM Multimedia 93, 1-6 August 1993, Anaheim, California (1993) 157–166
11. Naveed U. Qazi, Miae Woo, and Arif Ghafoor, "A Synchronization and Communication Model for Distributed Multimedia Objects," Proceedings on ACM Multimedia 93, 1-6 August 1993, Anaheim, California (1993) 147–155
12. Junewha Song, G. Ramalingam, Raymond Miller, Byoung-Kee Yi, Interactive authoring of multimedia documents in a constraint-based authoring system. Multimedia Systems **7** Springer-Verlag (1999) 424–437
13. M.Y. Sung, D.H. Lee: A Collaborative Authoring System for Multimedia Presentation. Proceedings on The 2004 International Conference on Communications (ICC 2004), Paris, France, June 20-24, 2004, MM04-2 (2004) 1396–1400