

# Authoring Together in a 3D Spatio-Temporal Space

Mee Young Sung  
Department of Computer  
Science & Engineering  
University of Incheon  
Incheon, Korea  
82 32 770 8496  
mysung@incheon.ac.kr

Do Hyung Lee  
Department of Computer  
Science & Engineering  
University of Incheon  
Incheon, Korea  
82 32 770 8496  
oldkill@hanmail.net

Seung Jin Rho  
Department of Computer  
Science & Engineering  
University of Incheon  
Incheon, Korea  
82 32 770 8496  
sjrho@incheon.ac.kr

Sang-Yong Rhee  
Division of Information &  
Computer Engineering  
Kyungnam University  
Masan, Korea  
82 55 249 2706  
syrhee@kyungnam.ac.kr

## ABSTRACT

In this paper, we propose a collaborative multimedia authoring system. The system integrates the spatial editing interface and the temporal editing interface in one 3D graphical interface and allows users in different places to author together a multimedia presentation simultaneously in a shared spatio-temporal space. This interface allows for simultaneous authoring and manipulation of both the temporal and the spatial aspects of a presentation. Using our system, users can design multimedia presentations collaboratively in the unified spatio-temporal space while freely traversing the spatial domain and the temporal domain without changing the context of authoring. Our authoring system internally represents the edited multimedia presentation using a Temporal Relation Network (TRN), which corresponds exactly to the conceptual temporal structure of the multimedia presentation. We also propose an algorithm for automatically generating a TRN from the graphical representation specified by the author of the presentation. In addition, we suggest some ideas for efficient concurrency control. They are mainly based on user awareness, multiple versions, and access permissions of shared objects.

## Categories and Subject Descriptors

H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces – *Collaborative computing, Computer-supported cooperative work, Synchronous interaction*

## General Terms

Algorithms, Design, Experimentation

## Keywords

Collaborative Authoring, Multimedia, Presentation, 3D Graphical Interface, Spatio-Temporal Relations, Concurrency Control

## 1. INTRODUCTION

A multimedia authoring system must provide an environment where the temporal relations and spatial relations can be edited simultaneously. An interactive multimedia authoring system also needs to support user interactions. Some media (such as video, sound, and animation) require users to specify temporal characteristics, and other media (such as video, animation, images,

and text) require users to specify the spatial relationship between objects. Computer designers are reworking the way humans interact with computers, trying to find a feasible replacement to the two-dimensional (2D) desktop-metaphor. Many researchers are interested in developing an innovative 3D interface for operating systems that transforms the desktop into an interactive 3D environment. If these new systems catch on, it could change the whole concept of how we relate to our machines. Many computer game programs are already using 3D interfaces. It is clear that 3D interfaces are valuable because they provide an intuitive and easy-to-use human computer interaction. The 3D interface is being accepted as an efficient user interface and is becoming a universal interaction method for computing technologies. However, research on authoring systems with 3D interfaces is currently limited to specific fields, such as CAD design or simulations. Our study intends to extend 3D interfaces for use in general multimedia authoring applications.

The key to authoring a presentation lies in the composition of spatial relationships and temporal relationships between objects. The existing authoring tools usually provide an authoring environment where the spatial information and temporal information are edited independently in two different 2D GUIs (Graphical User Interfaces), which can inconvenience the users. One of the GUIs represents the spatial characteristics of the multimedia content and the other represents its temporal characteristics. However, a 2D interface is not sufficient to completely represent multimedia information. Because, the spatial domain itself is two-dimensional, a 2D presentation space cannot accommodate the characteristics of both the spatial and temporal domains at the same time. Addition of the temporal dimension to the two spatial dimensions results in three dimensions. We can represent simultaneously the 2D space and the 1D time in three-dimensional space. Therefore, a multimedia authoring tool benefits greatly from a 3D interface which intuitively represents the multimedia content in one seamless environment. Using a 3D interface, multimedia authors can display and edit the complete spatial and temporal information simultaneously.

Our 3D interface for interactive multimedia authoring requires a 3D engine to analyze the multimedia information and transfer appropriate information into 3D interface. Various editing facilities are also needed for providing an efficient authoring environment. Our system is based on SMIL, an XML-based markup language for integrated streaming media. Our system allows users to compose and edit SMIL content in 3D space.

We developed a collaborative multimedia authoring system based on the SMIL (Synchronized Multimedia Integration Language) [3, 4] 1.0 Recommendation [3] and the SMIL 2.0 Recommendation [4]. The existing SMIL authoring tools provide basic user interfaces such as scaled timeline-based user interfaces (representing media objects as different bars arranged in multiple

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*ITP 2002*, December 6, 2002, Juan-Les-Pins, France.

Copyright 2002 ACM 1-58113-640-4/02/0012...\$5.00.

layers on the scaled timeline) or textual tag editing user interfaces for authoring. What distinguishes our system is that it provides a unified 3D interface that allows for simultaneous authoring and manipulation of both the temporal and the spatial aspects of a presentation.

In this paper, we propose a collaborative authoring system which integrates the spatial editing interface and the temporal editing interface in one 3D graphical interface. Our system contains a 3D spatio-temporal editor, a tag editor, an attribute editor, a text editor, and a Media Object Manager. The Media Object Manager is responsible for visualizing media objects in 3D, maintaining the consistency of the presentation information, and distributing all modification information from each editor to the other editors. We will examine our mechanism for representing spatio-temporal relationships in 3D in the following section. In section 3, we will investigate the Temporal Relation Network (TRN) upon which our model is based. The concurrency control model of our system will be discussed in section 4. Finally, the last section will provide conclusions and some future work.

## 2. REPRESENTATION OF SPATIO-TEMPORAL RELATIONS IN 3D

Our authoring system represents a multimedia presentation in a 3D coordinate system. One axis represents the traditional timeline information (T-zone), and the other two axes represent spatial coordinates (XY-zone) as shown in Figure 1. Our system represents a visual media objects as a 3D parallelepipeds and audio media objects as cylinders. The length of the shape along the time axis corresponds to the duration of the media. A cross section of the parallelepiped corresponds the spatial area of the visual media to be presented.

Figure 1 illustrates a perspective view of a multimedia presentation. This presentation consists of four media objects, an audio clip (in green), a video clip (in red), an image (in blue), and a text object (in pink). Authors can create media objects, place objects at the desired positions, and enlarge or shrink the temporal length of objects by dragging and dropping. Authors can change the perspective from which the objects are viewed in 3D space using the arrow keys.

Figure 2 corresponds to the temporal projection view of the example in Figure 1. Figure 3 corresponds to the spatial projection view of the example in Figure 1. Authors can quickly change to these default views by selecting a corresponding icon.

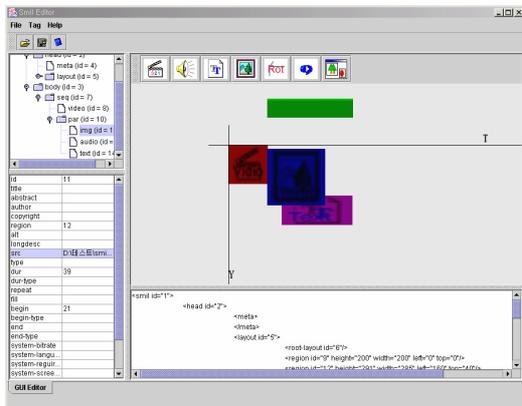


Figure 1. An Example of 3D Representation of a Multimedia Presentation: Perspective View

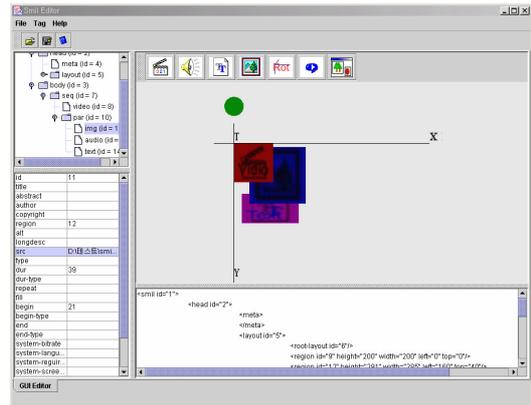


Figure 2. Temporal View of Figure 1

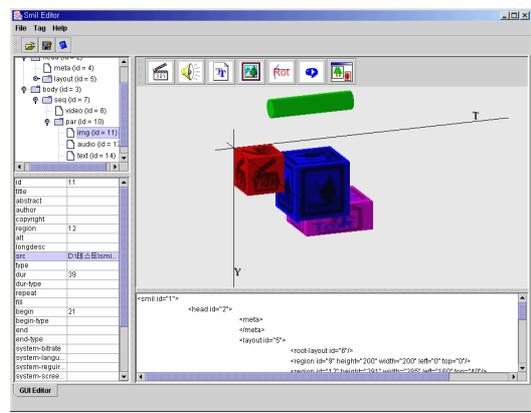


Figure 3. Spatial View of Figure 1

Our system allows users to drag and drop any object of local or network file systems into the 3D spatio-temporal space. For example, users can add a video object to a multimedia presentation by dragging and dropping the corresponding video file from the file system viewer. In addition, every object in our 3D spatio-temporal space can be enlarged, shrunken, or replaced in the spatial aspect as well as in the temporal aspect. As a multimedia presentation is being designed step-by-step, the corresponding SMIL code is dynamically updated in real-time and appears in its window.

## 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 [5, 6, 7, 8]. Some existing studies on the internal representation of multimedia applications include: OCPN (Object Composition Petri Nets) [2], DTPN (Dynamic Timed Petri Nets) [5], XOCPN (eXtended OCPN) [6, 7], 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. In our study, we propose using a TRN (Temporal Relation Network) as the internal representation of a multimedia presentation.

### 3.1 Internal Representation of Temporal Relations Using TRN

Our system's internal representation for a presentation is based on Allen's temporal intervals [1]. Conceptually, every temporal relationship can be described using one of seven relations ('before', 'meets', 'overlaps', 'during', 'starts', 'finishes', and 'equals'). The conceptual representation provides an efficient means for our multimedia authoring system to automatically fill in the necessary timing details.

We propose a graph structure which describes the conceptual temporal relationships among media objects as the internal representation of a multimedia presentation. We named the graph structure TRN (Temporal Relation Network). A TRN is a directed and weighted graph that corresponds exactly to the conceptual temporal structure of a multimedia presentation. The right column of Figure 4 shows the TRN structure of corresponding conceptual temporal relationship.

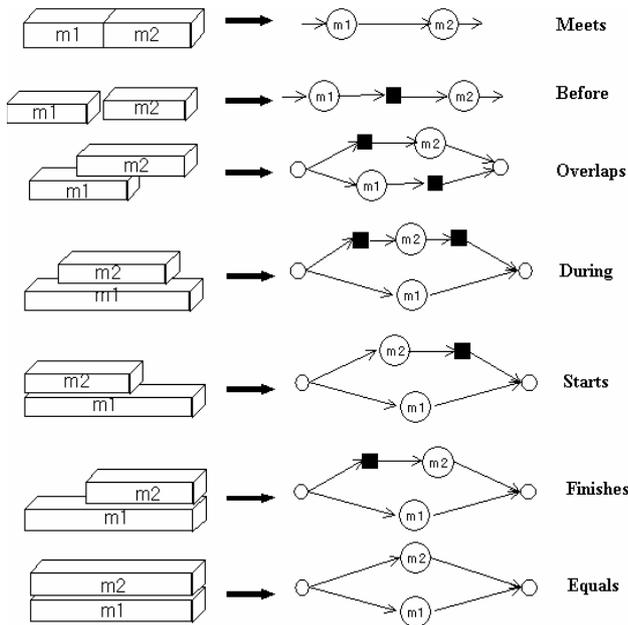


Figure 4. Internal Representation of Temporal Relations

Note that we represent the parallel relationships (such as 'overlap's, 'during', 'start's, and 'finishes') by adding dummy delay objects (represented as small black squares) to the 'equal' relationship as shown in Figure 4. Therefore, all five parallel relations can be generalized as the 'equal' relation by inserting some delay objects when they are needed.

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.

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 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 automatically generates SMIL documents through the interaction between TRN and DOM [10].

### 3.2 Automatic Generation of TRN from the 3D Spatio-Temporal Representation

In this section, we briefly discuss our algorithm for automatically generating a TRN from the graphical representation specified by the author of the presentation. The algorithm 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.

## 4. CONCURRENCY CONTROL

Collaborative systems need concurrency control to resolve conflicts between users' simultaneous operations. There are many concurrency control mechanisms, such as simple locking, transaction mechanism, turn-taking protocols, centralized controller, dependency detection, and reversible execution [9]. Finding a best concurrency control algorithm absolutely depends on the application semantics. Also, it requires us to suffer from the tradeoff between the responsiveness and the performance for keeping the consistency of shared data.

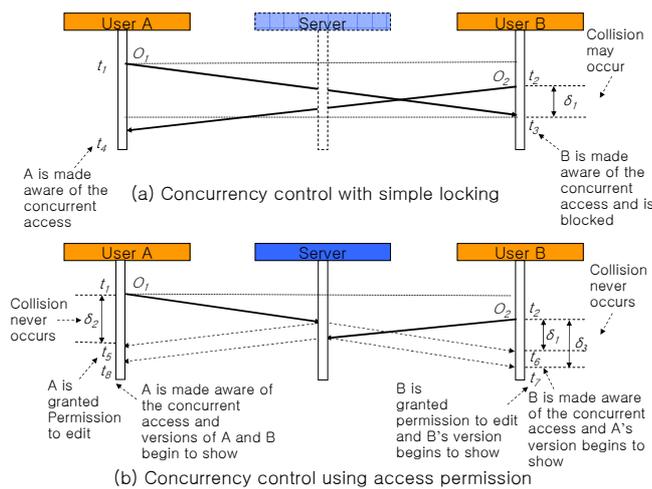
Our experience in the development of various collaborative systems leads us to conclude that a combination of the following methods can produce satisfactory results:

- Make users aware of concurrent operations
- Provide a fine granularity of sharing
- Maximize the responsiveness while minimizing collisions

In our 3D collaborative authoring system, we propose the following ideas for efficient concurrency control:

- Make users aware of every version of ongoing concurrent operations by changing the appearance of objects in concurrent access. One possibility is to give such objects a transparent look and show all concurrent operations. After the concurrent operations are complete, the proper version will be chosen as the final version made visible to all users.
- Maximize the locking granularity by separating the temporal editing operations and the spatial editing operations of an object and applying different concurrency control mechanisms to each.
- Maximize the responsiveness using optimistic concurrency control with versioning, and minimize the collision due to the concurrent operations by requiring users to obtain access permission before editing.

Figure 5 shows some details of concurrency control based on the access permission.



**Figure 5. Concurrency Control with Access Permission**

$O_1$  and  $O_2$  correspond to the concurrent operations of two users on a shared object. Variables of times and time intervals in Figure 5 are as follows:

- $t_1$ :  $O_1$  is started
- $t_2$ :  $O_2$  is started
- $t_3$ : B is made aware of the concurrent access and is blocked
- $t_4$ : A is made aware of the concurrent access
- $t_5$ : A is granted permission to edit
- $t_6$ : B is made aware of the concurrent access and A's version begins to show
- $t_7$ : B is granted permission to edit and B's version begins to show
- $t_8$ : A is made aware of the concurrent access and versions of A and B begin to show
- $\delta_1$ : Time interval between the request and the awareness of concurrent access (Collision may occur)
- $\delta_2$ : Time interval for obtaining A's access permission (A is blocked and collision never occurs)
- $\delta_3$ : Time interval for obtaining B's access permission (B is blocked and collision never occurs)

## 5. CONCLUSION

We created a 3D spatio-temporal collaborative authoring system which allows users in different places to design a multimedia presentation at the same time. The shared 3D interface integrates the 2D spatial editing environment and the 1D temporal editing environment into a single, unified editing environment. Our authoring system automatically converts an authored multimedia presentation to a Temporal Relation Network (TRN) for its internal representation. A TRN corresponds exactly to the conceptual temporal structure of the multimedia presentation. Our system also provides an efficient concurrency control mechanism based on user awareness, multiple versions, and access permissions of shared objects.

One advantage of our system is that multimedia authors need not change the editing environment from the spatial editing environment to the temporal editing environment, and vice-versa. Another advantage is its TRN representation of multimedia presentations. We believe that the TRN representation can provide an efficient means for optimal automatic scheduling mechanisms to guarantee fine-grained synchronization. In addition to their use in a SMIL authoring system, our system components can be used in a 3D virtual environment, for example a virtual collaborative system.

In the future, we must further explore the concurrency control mechanism of shared objects to ensure their consistency and coherence. Also, we plan to study optimal automatic scheduling mechanisms to guarantee fine-grained synchronization using TRN.

## 6. ACKNOWLEDGMENTS

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

## 7. REFERENCES

- [1] James F. Allen, "Maintaining Knowledge about Temporal Intervals", *Communications of the ACM*, pp832-843, November 1983.
- [2] Thomas D. C. Little, and Arif Ghafoor, "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks", *IEEE Computer*, Vol.24, No.10, pp42-50, October 1991.
- [3] W3C, Synchronized Multimedia Integration Language (SMIL) 1.0 Specification: W3C Recommendation 15-June-1998, <http://www.w3.org/TR/REC-smil/>
- [4] W3C, Synchronized Multimedia Integration Language (SMIL 2.0): *W3C Recommendation 07 August 2001*, <http://www.w3.org/TR/smil20/>
- [5] 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, pp110-119, 1994.
- [6] 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, pp157-166, 1993.
- [7] 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, pp147-155, 1993.
- [8] Junewha Song, G. Ramalingam, Raymond Miller, and Byoung-Kee Yi, "Interactive authoring of multimedia documents in a constraint-based authoring system", *Multimedia Systems Vol. 7*, pp424-437, Springer-Verlag, 1999.
- [9] Clarence A. Ellis, Simon J. Gibbs, and Gail L. Rein, *Groupware: Some Issues and Experiences. Communications of the ACM* 34, 1 (Jan. 1991), 38-58.
- [10] S.J. Rho, D.H. Lee, and M.Y. Sung, "Designing Multimedia Presentations in 3D Spatio-Temporal Space", *Proceedings on VSMM 2002*, Gyeongju, Korea, September 25-27, 2002.