# A Collaborative Authoring System for Multimedia Presentation

Mee Young Sung, Do Hyung Lee
Department of Computer Science & Engineering
University of Incheon
402-749 Incheon, South Korea
{mysung, oldkill}@incheon.ac.kr

***Abstract*— We propose a collaborative authoring system for multimedia presentation. Our multimedia authoring system benefits greatly from a 3D spatio-temporal interface which intuitively represents the multimedia presentation in one seamless environment. Our system is composed of an Editing System, a Media Object Manager and a Collaboration Manager. The Editing System contains two unique editors; a 3D Spatio-Temporal Editor and a Temporal Relation Network (TRN) Editor, in addition to the traditional editors such as Timeline Editor, Tag Editor, Attribute Editor, and Text Editor. These editors are all shared over the Internet and together they form an easy and efficient multimedia authoring environment. Using our system, users in different places can author multimedia presentations in a unified spatio-temporal space while freely traversing the spatial domain and the temporal domain without changing the context of authoring. Our system also provides users with an intuitive mechanism for representing the conceptual flow of a presentation. We implemented some ideas for efficient concurrency control in our system. They are mainly based on user awareness, multiple versions, fine granularity of locking, and access permissions of shared objects. Some our experiments lead us to conclude that our system is more advantageous than traditional multimedia authoring systems in terms of authoring time and ease of interaction.**

***Keywords-*** *collaboraative authoring; multimedia presentation; 3D Spatio-Ttemporal Editor; Temporal Relation Network Editor; SMIL (Synchronized Multimedia Integration Language)*

## I. INTRODUCTION

Collaborative authoring is one of the important collaborative works. We feel keenly the necessity of a good collaborative authoring tool for multimedia presentation. In authoring a multimedia presentation, 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. Various editing facilities are needed for providing an efficient multimedia authoring environment. Among other things, a multimedia authoring system must provide an environment where the temporal relations and spatial relations can be edited simultaneously [1, 2, 3, 4, 5]. An interactive multimedia authoring system also needs to support user interactions.

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 two-dimensional (2D) GUIs (Graphical User Interfaces). One of the GUIs represents the spatial characteristics of the multimedia content and the other represents its temporal characteristics. Traversing two different GUIs can inconvenience the users. Moreover, a 2D presentation space cannot accommodate the characteristics of both the spatial and temporal domains at the same time. Adding the temporal dimension to the two spatial dimensions results in three dimensions. We can represent simultaneously the 2D space and the one-dimensional (1D) time in three-dimensional (3D) space. Using a 3D interface, multimedia authors can display and edit the complete spatial and temporal information simultaneously. 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.

We developed a collaborative multimedia authoring system based on the SMIL (Synchronized Multimedia Integration Language). 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 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 present our collaborative authoring system that is efficient through providing various editing facilities in addition to the concurrency control mechanisms. We will briefly describe the system components of our system in the following section. The details of our editors will be discussed in section 3. In section 4, we will examine the concurrency control mechanism of our system. In section 5 and 6, we will investigate the implementation and the experiment of our system individually. Finally, the last section will provide conclusions.

 

## II. SYSTM COMPONENTS

Our system represents images, videos, sounds, texts, and animations as 3D objects, and provides various editing functionalities for temporal compositions and spatial compositions. Our system is composed of three main components: Editing System, Media Object Manager, and Collaboration Manager. Fig. 1 illustrates the overall structure of our system.

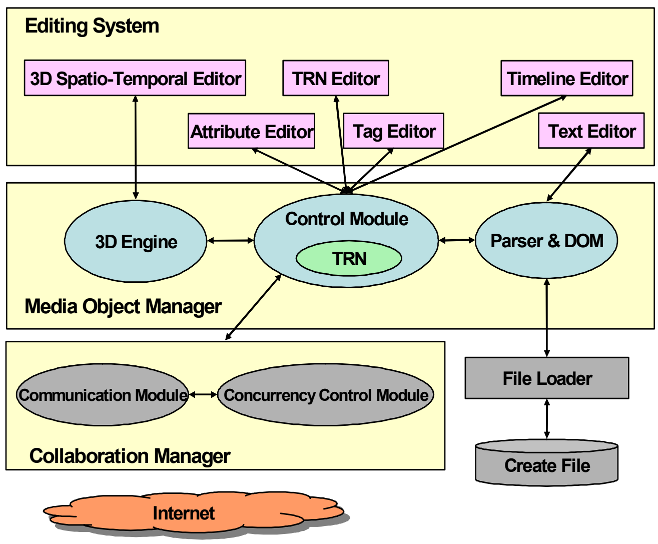


Figure 1. System Overview

• Editing System

The Editing System contains a 3D Spatio-Temporal Editor and a Temporal Relation Network (TRN) [6, 7] Editor, a Timeline Editor, a Tag Editor, an Attribute Editor, and a Text Editor. The 3D Spatio-Temporal Editor represents media in 3D space. This editor is responsible for integrating and analyzing the detailed information about the spatio-temporal relationships among media. It allows users to edit the temporal relations between media via simple and direct graphical manipulations in 3D space in a drag and drop manner. The TRN Editor visualizes the internal representation of the presentation and allows us to represent the conceptual flow of the presentation. The Timeline Editor is based on a timeline where the start time and the duration of the presentation of each media are represented. The Tag Editor can be used to directly insert and delete SMIL tags. This represents the structure of a document as a tree structure composed of tags. This allows users to view and edit the overall structure of the document. The structure of the Tag Editor corresponds exactly to the DOM (Document Object Model) structure of the presentation. The Attribute Editor allows for editing attributes of SMIL objects in detail which are difficult to represent in 3D, such as clip-begin, clip-end, system-bitrate, system-screen-size, system-language, etc. The Text Editor can be used to directly edit SMIL files in text form. This represents the actual SMIL code being generated or loaded from the underlying SMIL file. This allows users to verify the generated codes in real-time. These editors are all shared over the Internet and together they form an easy and efficient multimedia authoring environment. Using our Editing System, users in different places can author together a multimedia presentation simultaneously in a single unified spatio-temporal space. These editors exchange their information through the Media Object Manager and together form an easy and efficient editing environment.

• Media Object Manager

The Media Object Manager of our system is an essential part of our system. 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. The Media Object Manager is also responsible for visualizing temporal relations of media objects in graphical Temporal Relation Network. The Media Object Manager consists of a 3D Engine, a Control Module, and a Parser & DOM Generator. The 3D engine is the core part of our system. This engine transforms the user input into 3D objects and passes the user input and the corresponding 3D objects to the 3D Spatio-Temporal Editor for visualization. The visualization of objects in 3D coordinates depends on the user-selected basis plane for viewing, such as a temporal view, a spatial view, a perspective view, etc. The 3D Engine maintains the coordinates and the location information depending on the basis plane. The Control Module exchanges information between each editor and dynamically updates the status of all editors to ensure consistency. The control Module also creates and maintains the internal TRN structure, automatically inserting delay objects when necessary. Control Module also communicates with the DOM generator to read and store presentation information. The DOM generator works with information provided by the Control Module to generate a standardized DOM structure and the corresponding SMIL code. The parser verifies the syntax of the SMIL code that is automatically generated by the DOM Generator. The parser also displays the SMIL code on the Text Editor.

• Collaboration Manager

The Collaboration Manager takes in charge of real-time communications over the Internet and concurrency control of objects in simultaneous editing. The Collaboration Manager supports users in distance to communicate in real-time and author together a multimedia presentation. This manager processes the events and eliminates the collisions (the inconsistency of shared data which might be occurred when they are accessed concurrently). It consists of a Communication Module and a Concurrency Control Module. The Communication Module takes charge of the communications of all events generated by users. This connects to the server using TCP (Transmission Control Protocol) and generates packets corresponding to the content that is created as users edit the presentation. It also receives packets from the server, analyses the packets, and invokes appropriate events or modules. The Concurrency Control Module is connected to the Communication Module for concurrency control. It asks the server for access permission. After getting access permission, it controls and manages the shared objects that are accessed concurrently without collisions.

## III. EDITORS

Among the many editors that make up our system, the 3D Spatio-Temporal Editor and the TRN Editor are unique.

Our 3D Spatio-Temporal Editor allows users in different places for simultaneous authoring and manipulation of both the temporal and the spatial aspects of a presentation via simple and direct graphical manipulations in 3D space. 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 Fig. 2. 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 to the spatial area of the visual media to be presented. Authors can change the perspective from which the objects are viewed in 3D space using the arrow keys. Also, authors can quickly change to these default views by selecting a corresponding icon.

Fig. 2 illustrates a perspective view of a multimedia presentation. This presentation consists of five media objects, an audio clip, a video clip, two images, and a text object. Authors can create media objects, place objects at the desired positions, and enlarge or shorten the temporal length of objects by dragging and dropping. A structural view of SMIL tags of the presentation (corresponds exactly to the DOM structure of the presentation) and a view of a SMIL object's attributes are also presented on the left of Fig. 2.

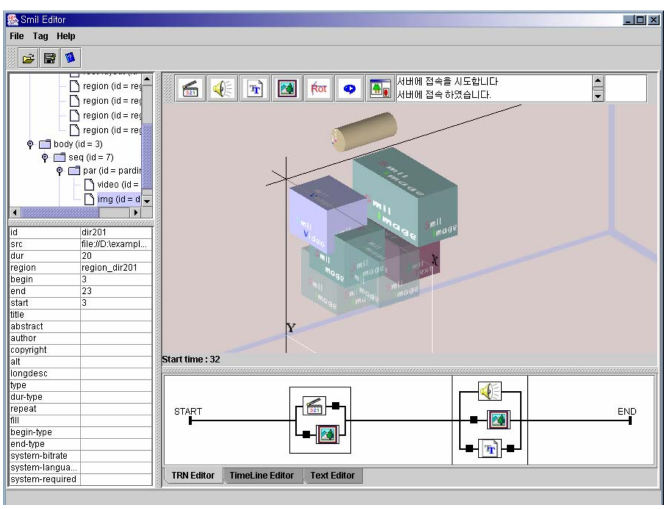


Figure 2. Example of 3D Representation of a Multimedia Presentation (viewing multiple versions of concurrent editing): Perspective View

Our authoring system internally represents the edited multimedia presentation using a Temporal Relation Network (TRN). TRN corresponds exactly to the conceptual temporal structure of the multimedia presentation. TRN is a directed and weighted graph. TRN representation of presentation is based on Allen's temporal intervals [8]. Conceptually, every temporal relationship can be described using one of seven relations ('before', 'meets', 'overlap', 'during', 'starts', 'finishes', and 'equals'). The conceptual representation provides users an efficient means for designing temporal scenario of the presentation.

Our TRN Editor provides users with an intuitive mechanism for representing the conceptual flow of a presentation by simple and direct graphical manipulations. The bottom right bottom of Fig. 2 illustrates an example graphical representation of TRN of the presentation that is created as a user authors the presentation. The internal TRN and the graphical TRN can be generated automatically from the 3D graphical representation specified by the author of the presentation. After the authoring is finished, a DOM structure can be generated from the 3D graphical representation. Our system generates SMIL documents through the interaction between the TRN and the DOM structure [6].

The author can choose to view between the TRN Editor, the Timeline Editor, or the Text Editor using the tabs at the bottom of the 3D TRN Editor panel shown in the Fig. 2. Details of our 3D Spatio-Temporal Editor and our TRN Editor are described in [6] and [7] individually.

## IV. CONCURRENCY CONTROL

Collaborative systems need concurrency control to resolve conflicts between users' simultaneous operations [9]. There are many concurrency control mechanisms are studied in the field of databases [10]. They are simple locking [11], transaction mechanism [12], turn-taking protocols [13], centralized controller [14], dependency detection [15], and reversible execution [16, 17], operation transformation [18], multiple versions [19], etc. Finding a best concurrency control algorithm absolutely depends on the application semantics. 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 different concurrency control methods can produce satisfactory results. In our 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. Our system gives such objects a transparent look and shows all concurrent operations as shown in Fig. 2.
- Maximize the responsiveness using optimistic concurrency control with multiple versioning of concurrent operations. After the concurrent operations are complete, the proper version will be chosen as the final version made visible to all users.
- Fine 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.
- Minimize the collisions due to the concurrent operations by requiring users to obtain access permission before editing [6].

Fig. 3 summarizes our algorithm for processing events relevant to the concurrency control.

```
mediaEditStart(Object mediaObject)
    // Invoked at the beginning of editing
begin
    // Gets the access permission and
    // creates a copy of the shared object  (cp_mediaObject)
    cp_mediaObject ::= getPermission(mediaObject);
end

mediaEditing(Object cp_mediaObject)
    // Invoked in the middle of editing
begin
    // Applies the user's edition to the copy of the shared object
    // Transmits the modifications to other users
    … editing …
end

mediaEditEnd(Object cp_mediaObject)
    // Invoked at the end of editng
begin
    // Confirms the status of the target shared object
    if (getMediaStatus(cp_mediaObject))
        // Applies the user's edition: substitutes the target
         // shared object with the copied transparent object
        mediaEditApply(cp_mediaObject);
    else
        // Undoes the user's edition : deletes the copied object
        mediaEditUndo(cp_mediaObject);
    endif
end

getMediaStatus(Object cp_mediaObject)
   // Examines whether the target object is accessed concurrently
Begin
   // Returns false(undo) when the number of users in access
    // is greater than 1
   if (cp_mediaObject.orginal_mediaObject.cp_count != 1)
        return false;
   // Returns true(apply the edition) when the number of users
    // in access is equal to 1
   else if (cp_mediaObject.orginal_mediaObject.cp_count == 1)
        return true;
end
```

Figure 3. Algorithm for Processing Events Relevant to the Concurrency Control

## V. IMPLEMENTATION

Our system is implemented using Java technology, such as J2SDK (Java 2 Standard Development Kit) 1.4.1 (including Java sockets), Java3D 1.3 library, JAXP (Java API for XML Processing) 1.1. Fig. 4 shows the structure of the main Java classes of our system.

The SmilEditor class manages and executes the overall system. It creates three classes for generating the interfaces. They are Inter3D, TreePane, and TablePane. The SmilEditor class also generates the classes for creating and manipulating the TRN (the internal representation of a multimedia presentation) and the classes for managing the network communications. The Inter3D (means the 3D Interface) class generates the classes for managing the 3D Spatio-Temporal Editor, such as SmilRoom. The class SmilRoom creates a 3D spatio-temporal space and also generates the RegionTrans class and MouseTrans class for the edition in the 3D spatio-temporal space. The RegionTrans class and the TimeTrans class take charge of transferring every editing result to the TRN structure. The Inter3D class also generates the transparent objects of a concurrently accessed object and collaborates directly with the Network class in the case of collaborative authoring. The TreePane class generates the classes for managing the Tag Editor. The TablePane class generates the classes for managing the Attribute Editor. The TRN class creates a template of the SMIL document structure and modifies the values of the tag structure in real-time as the users author the presentation. It also generates a DOM structure and a SMIL file corresponding to the presentation using the _3DtoSMIL class, the PrettyDOM class, and the DomMaker class. The Network class is generated when a collaborative authoring operation begins. It uses the shared media objects (transparently viewed) and works for the concurrency control in collaboration with the Inter3D class and the TRN class.

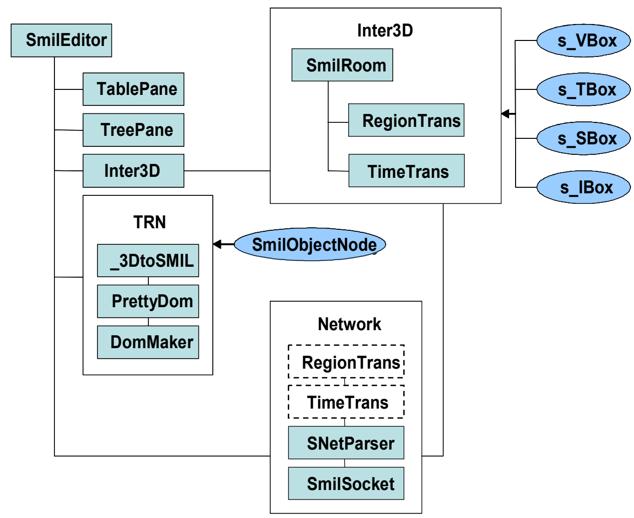


Figure 4. Structure of Main Java Classes of Our System

The experience on the development of our system using Java technology leads us to conclude that the Java technology is a satisfactory tool for developing a collaborative authoring system which is one of the distributed applications. Java's network capability is especially excellent. However, the Java3D technology needs to be ameliorated in terms of performance, even the programming with Java3D is easier than with OpenGL or DirectX directly.

## VI. EXPERIMENT

We performed some experiments to validate the usability of our authoring system. The experiment was set with 10 university students who are good at computing. In the first experiment, the students proposed to author a multimedia content comprised of several media objects. We let five students to use the 3D editor (our 3D Spatio-Temporal Editor) first, then the 2D editor (TagFree 2000 SMIL Editor of Dasan Technology) second. We let another five students use the 2D editor first, then the 3D editor second. In the second experiment, we gave the students 6 different scenarios of presentations whose number of objects increases by 2 (such as 1, 3, 5, 7, 9, and 11) and let them author those presentations by crossing the editors in the same manner as the first experiment. Table1 presents the average editing time for authoring 11 objects.

TABLE I. AVERAGE EDITING TIME FOR AUTHORING 11 OBJECTS

| | 3D Interface | 2D Interface |
|---|---|---|
| 3D Interface First | 6 min 06 sec | 8 min 32 sec |
| 2D Interface First | 5 min 35 sec | 8 min 48 sec |

The results obtained through these experiments explain that the 3D authoring interface allows users to author faster than the two 2D authoring interfaces. We also noticed that the difference between the average authoring time for 2D and that of 3D increases as the complexity of media objects increases. The first reason for latency comes from the time consuming adjustment of media objects on the scaled time-line for representing the temporal relationships such as 'parallel' or 'sequential'. The second reason originates from the change of the authoring environment from the temporal interface to the spatial interface, or vice-versa. The last reason is caused by the separate editing of layout regions in the spatial interface followed by the manual linkages of the layout objects to the objects in the temporal interface. Our 3D authoring interface allows users to recognize the spatio-temporal characteristics at a glance. Moreover, it is obsolete to link the temporal objects to the spatial objects in our 3D authoring environment. The students who participated in this experiment concluded that our 3D authoring interface is an intuitive and faster authoring tool. Generally, our system was favored above the others. However, the students highlighted one inconvenience of our system. That is the difficulty for editing the overlapped objects in 3D, since the grasp of the sense of distance and pointing out the appropriate 3D distance with the mouse in the 3D perspective view is not always evident.

## VII. CONCLUSION

The objective of this study was to develop an easy and efficient collaborative authoring environment for multimedia presentation. Toward this end, we created a collaborative authoring system providing two special editors; 3D Spatio-Temporal Editor and TRN Editor. In our 3D Spatio-Temporal Editor, the spatial aspects and the temporal aspects of a multimedia presentation are represented in an integrated environment, so that users can create a multimedia presentation in a simple and intuitive manner. Our authoring system automatically converts the 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. The internal TRN is visualized in the TRN Editor. We also provide a concurrency control mechanism that is a combination of user awareness, multiple versions, fine granularity of locking, 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. The TRN representation provides an intuitive means for designing a temporal overview of a presentation at the conceptual level. This allows authors to focus on the creative aspects of the presentation. Also, we believe that the TRN representation can provide an efficient method for optimal automatic scheduling mechanisms to guarantee fine-grained synchronization.

We performed some experiments to validate the usability of our authoring system. The experiments lead us to conclude that our system is more advantageous than traditional multimedia authoring systems in terms of authoring time and ease of interaction. 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. In addition to their use in a collaborative SMIL authoring system, our system components can be used in a 3D networked virtual environment, for example a virtual collaborative system.

## ACKNOWLEDGMENT

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

## REFERENCES

[1] M. Y. Kim, J. Song, "Multimedia Documents with Elastic Time," *Proceedings of ACM Multimedia '95*, November 5-9, 1995, San Francisco, California, USA, pp143-154, 1995.

[2] J. Song, G. Ramalingam, R. Miller, B. K. Yi, "Interactive authoring of multimedia documents in a constraint-based authoring system," *Multimedia Systems*, Vol. 7, pp424-437, Springer-Verlag, 1999.

[3] M. Vazirgiannis, I. Kostalas, T. Sellis, "Specifying and Authoring Multimedia Scenarios," *IEEE Multimedia*, Vol. 6, No. 3, pp24-37, July-September 1999.

[4] A. Karmouch, J. Emery, "A Playback Schedule Model for Multimedia Documents," *IEEE Multimedia*, Vol. 3, No. 1, pp50-61, Spring 1996.

[5] T. D. C. Little, A. Ghafoor, "Spatio-Temporal Composition of Distributed Multimedia Objects for Value-Added Networks," *IEEE Computer*, Vol. 24, No. 10, pp42-50, October 1991.

[6] M. Y. Sung, D. H. Lee, S. J. Rho, S. Y. Rhee "Authoring Together in a 3D Spatio-Temporal Space," Proceedings of ACM Multimedia 2002 Workshop ITP (International Conference on Immersive Telepresence), December 6, 2002, Juan-les-Pins, France, 4 pages, 2002

[7] M. Y. Sung, S. J. Rho, J. H. Jang, "A SMIL-based Multimedia Presentation Authoring System and Some Remarks on Future Extension of SMIL," *Proceedings of Packet Video 2002*, Pittsburgh, Pennsylvania, USA, 11 pages, April 24-26, 2002, http://www.pv02.org.

[8] J. F. Allen, "Maintaining Knowledge about Temporal Intervals," *Communications of the ACM*, Vol. 26, No. 11, pp832-843, November 1983.

[9] C. A. Ellis, S. J. Gibbs, G. L. Rein, "Groupware: Some Issues and Experiences," *Communications of the ACM*, Vol.34, No.1, pp.38-58, January 1991.

[10] J. Gray, A. Reuter, Transaction Processing: Concepts and Techniques Morgan Kaufmann, 1070 pages, 1993.

[11] M. Stefik, D. G. Bobrow, G. Foster, S. Lanning, D. Tatar, "WYSIWIS Revised: Early Experience with Multiuser Interfaces," *ACM Transactions of office Information Systems*, Vol. 5, No. 2, pp147-167, April, 1987

[12] S. Greenberg, D. Marwood, "Real Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface," *Proceedings of ACM CSCW'94*, October 22-26, 1994, Chapel Hill, North Carolina, USA, pp207-217, 1994.

[13] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System," *Communications of the ACM*, Vol. 21, No. 7, pp558-565, July 1978.

[14] D. R. Jefferson, "Virtual Times," ACM Transactions on Programming Language and Systems, Vol. 7, No. 3, pp404-425, July 1985.

[15] K. A. Lants, "An Experiment in Integrated Multimedia Conferencing," *Proceedings of ACM CSCW '86*, pp267-275, 1986.

[16] A. Prakash, M. J. Knister, "A Framework for Undoing Actions in Collaborative Systems," *ACM Transactions on Computer-Human Interaction,* Vol. 1, No. 4, pp295-330, December 1994.

[17] S. Sarin, I. Grief, "Computer-Based Real-Time Conferences," *IEEE Computer*, Vol. 18, No. 10, pp33-45, October 1985.

[18] C. A. Ellis, S. J. Gibbs, "Concurrency Control in Groupware Systems," *Proceedings of ACM SIGMOD International Conference on Management of Data,* Seattle, Washington, USA, pp399-407, 1989.

[19] C. Sun, D. Chen, "Consistency Maintenance in Real-Time Collaborative Graphics Editing Systems," *ACM Transactions on Computer-Human Interaction*, Vo. 9, No. 1, pp1-41, March 2002.