A Java-Based Collaborative Authoring System for Multimedia Presentation

Mee Young Sung and Do Hyung Lee

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

Abstract. In this paper, we propose a Java-based collaborative authoring system for multimedia presentation. 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. We also 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. Our work on the development of a Java-based collaborative authoring system leads us to conclude that the Java technology is a satisfactory tool for developing a collaborative authoring system including a 3D interface. However, the Java3D technology needs to be ameliorated in the aspect of performance.

1 Introduction

Various editing facilities are needed for providing an efficient authoring environment. Among other things, a multimedia authoring system must provide an environment where 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.

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). 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.

K. Aizawa, Y. Nakamura, and S. Satoh (Eds.): PCM 2004, LNCS 3332, pp. 96–103, 2004.

Moreover, 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. Adding 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.

We developed a collaborative multimedia authoring system based on the SMIL (Synchronized Multimedia Integration Language). SMIL is an XML-based markup language for integrated streaming media. 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 from others is that it provides a unified 3D interface that allows for simultaneous authoring and manipulation of both the temporal and spatial aspects of a presentation.

In this paper, we propose a Java-based collaborative authoring system for multimedia presentation that is efficient through providing various editing facilities including concurrency control mechanisms. We will describe three major components of our system in the following section. The representation of temporal relations will be discussed in section 3. In section 4, we will examine the concurrency control mechanism. The implementation and the experiment of our system will be presented in Section 5 and 6 individually. Finally, the last section will provide conclusions and outline plans for future work.

2 System Structure

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

The Editing System consists of a 3D Spatio-Temporal Eeditor, a Temporal Relation Network (TRN) Editor, a Timeline Editor, a Tag Editor, an Attribute Editor, and a Text Editor. These editors exchange their information through the Media Object Manager and together form an easy and efficient editing environment. Among many editors, the 3D Spatio-Temporal Editor and the TRN Editor are special. The 3D Spatio-Temporal Editor represents media in 3D space. The Spatio-Temporal 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. TRN Editor vi-

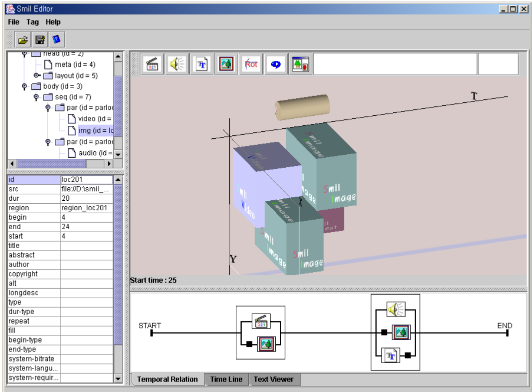

Fig. 1. An Example of 3D Representation of a Multimedia Presentation: Perspective View

sualizes the internal representation of the presentation and allows us to represent the conceptual flow of the presentation.

The Media Object Manager of our system is an essential part of our system. It 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 consists of three modules: a 3D Engine, a Control Module, and a Parser & DOM Generator.

The Collaboration Manager allows a group of users working at different machines to work on the same multimedia presentation, and to communicate in real time. It processes the events and eliminates the collisions (the inconsistency of shared data which might occur when they are accessed concurrently). Our Collaboration Manager is composed of two modules: a Communication Module and a Concurrency Module.

3 Representation of Temporal Relations

Our 3D Spatio-Temporal Editor allows users in different places to simultaneously author and manipulate both the temporal and the spatial aspects of a presentation. 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 visual media object as 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.

Figure 1 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 in 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 Figure 1. Authors can change the perspective from which the objects are viewed in 3D space using the arrow keys. Also, authors can quickly change to default views by selecting a corresponding icon. Details of our 3D Spatio-Temporal editor are described in the references [1].

Our system uses TRN (Temporal Relation Network) [1]as its internal representation of a multimedia presentation. TRN is based on Allen's temporal intervals [2]. 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. Our TRN supports these seven relationships. TRN corresponds exactly to the conceptual temporal structure of the multimedia presentation. TRN is a directed and weighted graph. Figure 1 also illustrates an example graphical representation of conceptual temporal relation of the presentation that is created as a user authors the presentation. After the authoring is finished, a DOM structure is generated from the graphical representation. The internal TRN and the graphical TRN are generated automatically from the 3D graphical representation specified by the author of the presentation. Our system finally generates SMIL documents through the interaction between the graphical representation and the DOM structure.

4 Concurrency Control

Collaborative systems need concurrency control to resolve conflicts between users' simultaneous operations [3]. There are many concurrency control mechanisms are studied in the field of databases [4]. They are simple locking [5], transaction mechanism [6], turn-taking protocols [7], centralized controller [8], dependency detection [9], and reversible execution [10][11], operation transformation [12], multiple versions [13], etc. Finding a best concurrency control algorithm absolutely depends on the application semantics [14]. 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 concurrency control methods can produce satisfactory results [1]. 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. 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 is 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.

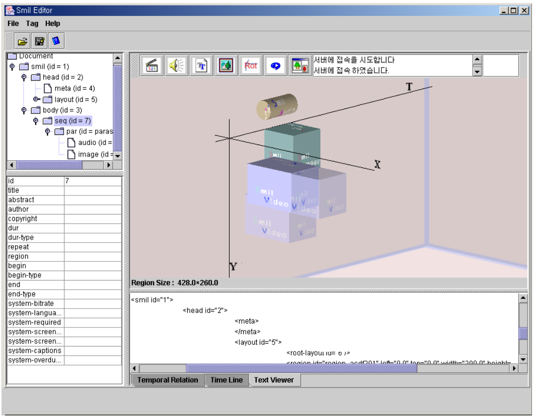

Fig. 2. An Example of Viewing Multiple Versions of Concurrent Editing

• 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 2 presents an example of concurrent editing of an object. Details of concurrency control based on the access permission can be found in the reference [1].

5 Implementation

Our system is implemented using J2SDK (Java 2 Standard Development Kit) 1.4.1 and Java3D 1.3 library. The execution of our system requires an XML parser, since the grammar of the SMIL code is based on XML. Our system uses the JAXP (Java API for XML Processing) 1.1 for XML parsing. The collaboration module is implemented using Java sockets.

Figure 3 shows the structure of the main Java classes of our system. The role of each class is as follows:

• SmilEditor: This 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.

• Inter3D (means the 3D Interface): It 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 generates the transparent classes and collaborates directly with the Network class in the case of collaborative authoring.

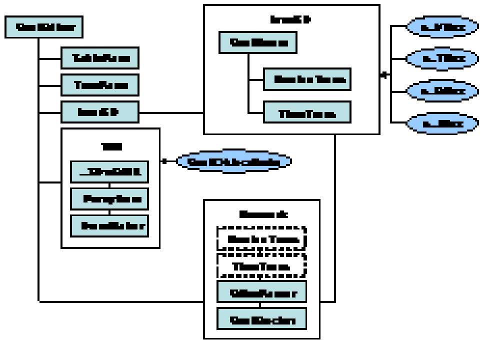

Fig. 3. Structure of Main Java Classes of Our System

- TreePane: It generates the classes for managing the Tag editor.
- TablePane: It generates the classes for managing the Attribute editor.
- TRN: This 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.
- Network: This class is generated when a collaborative authoring operation begins. It creates the shared media objects (transparently viewed) and works for the concurrency control in collaboration with the Inter3D class and the TRN class.

6 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 system) 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.

The results obtained through these experiments are as follows:

- The 3D authoring interface allows users to author faster than the two 2D authoring interfaces.
- 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.

7 Conclusion

The objective of this study was to develop an easy and efficient collaborative authoring environment for multimedia presentation using Java technology. Toward this end, we created a shared authoring system which is composed of several editors such as a 3D Spatio-Temporal editor, a Temporal Relation Network (TRN) editor, a Timeline editor, a Tag editor, an Attribute editor, and a Text editor. Among these editors, the 3D Spatio-Temporal editor and the TRN editor are special.

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, and access permissions of shared objects. Our system allows users to compose and edit SMIL content in 3D space. 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.

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. The TRN representation allows the system to automatically fill in the necessary timing details. This frees multimedia authors to focus instead on the creative aspects of the presentation. Also, we believe that the TRN representation can provide an efficient means for optimal automatic scheduling mechanisms to guarantee fine-grained synchronization.

We implemented our system using Java technology. Our work on the development of a Java-based collaborative multimedia authoring system leads us to conclude that the Java technology is a satisfactory tool for developing a collaborative multimedia authoring system. 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.

In the future, we must further explore the concurrency control mechanism of shared objects to ensure their consistency and coherence. We want also to examine the future extension of the SMIL standard for supporting authoring and the playback of 3D multimedia presentations in a 3D virtual environment.

Acknowlegement. 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. Sung, D.H. Lee: A Collaborative Authoring System for Multimedia Presentation. Proceedings on The 2004 International Conference on Communications (ICC 2004) MM04–2 (2004) 1396–1400
2. James F., Allen: Maintaining Knowledge about Temporal Intervals. Communications of the ACM (November 1983) 832–843
3. C. A. Ellis, S. J. Gibbs, G. L. Rein: Groupware: Some Issues and Experiences. Communications of the ACM, **34**(1) (January 1991) 38–58
4. J. Gray, A. Reuter: Transaction Processing: Concepts and Techniques Morgan Kaufmann. (1993) 1070 pages
5. M. Stefik, D. G. Bobrow, G. Foster, S. Lanning, D. Tatar: WYSIWIS Revised: Early Experience with Multiuser Interfaces. ACM Transactions of office Information Systems **5**(2) (1987) 147–167
6. Dick C.A. Bulterman: SMIL 2.0 Part2: Examples and Comparisons. IEEE Multimedia **9**(1) (2002) 74–84
7. 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 (1994) 110–119
8. B. Prabhakaran and S. V. Raghavan: Synchronization Models For Multimedia Presentation With User Participation. Proceedings on ACM Multimedia 93 (1993) 157–166
9. Naveed U. Qazi, Miae Woo, and Arif Ghafoor: A Synchronization and Communication Model for Distributed Multimedia Objects. Proceedings on ACM Multimedia 93 (1993) 147–155
10. 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
11. S. Sarin, I. Grief: Computer-Based Real-Time Conferences. IEEE Computer **18**(10) (1985) 33–45
12. C. A. Ellis, S. J. Gibbs: Concurrency Control in Groupware Systems. Proceedings of ACM SIGMOD International Conference on Management of Data (1989) 399-407
13. C. Sun, D. Chen: Consistency Maintenance in Real-Time Collaborative Graphics Editing Systems. ACM Transactions on Computer-Human Interaction **9**(1) (2002) 1–41
14. J. Munson, P. Dewan: A Concurrency Control Framework for Collaborative Systems. Proceedings of ACM CSCW '96 (1996) 278–287