Implementing a Distributed Lecture-on-Demand Multimedia Presentation System

Lawrence Y. Deng, Timothy K. Shih, Sheng-Hua Shiau, Wen-Chih Chang and Yi-Jen Liu
Dept. of Computer Science and Information Engineering, Tamkang University, Tamsui, Taipei Hsien, Taiwan 251, R.O.C. (BIRD@CS.TKU.EDU.TW and TSHIH@CS.TKU.EDU.TW)

Abstract: *Lecture-on-Demand (LOD) multimedia presentation technologies among the network are most often used in many communication services. Examples of those applications include video-on demand, interactive TV and the communication tools on a distance learning system and so on. In this paper, we describe how to present different multimedia objects on a web-based presentation system. The distributed approach is based on an extended timed Petri net model. Using characterization of extended media streaming technologies, we developed a Web-based Multimedia Presentation System. For a real-world example, suppose a well-known teacher is giving a lecture/presentation to his student. Because of time constraints and other commitments, many students cannot attend the presentation. The main goal of our system is to provide a feasible method to record and represent a lecture/presentation in the air. Using the browser with the windows media services allows those students to view live video of the teacher giving his speech, along with synchronized images of his presentation slides and all the annotations/comments. In our experience, this very approach is sufficient to the use of distance learning environment.*

Keywords: *Lecture-on-Demand, Distributed Multimedia Presentation, Distance Learning, Media Streaming*

1. Introduction

To control and demonstrate different types of multimedia objects is one of important functions in distributed multimedia presentation system. The concept of our model is based on the Petri net [1-3]. Petri net is a graphical and mathematical modeling tool applicable to many systems. Its features can be used with both practice and theory. Thus, it provides a powerful medium of communication between them. Additional extensions have been proposed, and this has led to the following types of Petri nets: the timed Petri net, the stochastic Petri net, colored Petri net, and object-related Petri net [6-12]. The "Object Composition Petri Net"(OCPN) and the "extended Object Composition Petri Net"(XOCPN) are two graphic-based models that propose synchronous theoretical for multimedia. The OCPN is a comprehensive model for specifying timing relations among multimedia data. The XOCPN can specify temporal relationships for the presentation of pre-orchestrated multimedia data, and to set up channels according to the required Qos of the data [4, 5]. These two models lack methods to describe the details of synchronization across distributed platforms and do not deal with the schedule change caused by user interactions in interactive multimedia systems [13]. However, when considering the network transport issue of multimedia and the floor control with multiple users, OCPN/XOCPN model are not sufficient to deal with those problem. In this paper, we use the extended timed Petri net to construct the web operations on a distance learning system. When multimedia objects are represented on the system, we have to consider different situations of multimedia objects such as asynchronous operations, time scheduling, and flow-control. In addition to system operations, dynamical operations of users are important issues. Thus, we can apply characteristic of Petri net to implement our mechanism and study the theory.

Before we design our system, we have surveyed both academic researches and industrial software packages. Sony Corporation developed a hypermedia prototype system (SAL) for multimedia authoring, which is based on a link and node model used in most authoring systems. The Layered Multimedia Data Model (LMDM) allows the reuse of presentation templates, which is important for improving the efficiency of multimedia presentation designs. LMDM has a number of strengths including the support of a general model of media synchronization, limited system dependencies, and the generalization of a traditional animation model. The work discussed in proposes a framework, which provides a homogeneous strategy to access, process, and exchange multimedia documents generated by different authoring and presentation systems. Diamond is a multimedia message system built on a distributed architecture for creating, editing, transmitting, and managing multimedia documents. Multimedia documents are stored in folders in a supports heterogeneous multimedia data types and allows new types to be added. A platform independent multimedia presentation composition system is discussed in [17].

Proceedings of the 22 nd International Conference on Distributed Computing Systems Workshops (ICDCSW'02)
0-7695-1588-6/02 $17.00 © 2002 **IEEE**

We also looked at the following commercial products related to multimedia authoring or presentation designs: (1) Authorware by Macromedia, Inc. (2) Multimedia Viewer by Microsoft (3) Multimedia Toolbook by Asymetrix Corporation (4) Hypermedia Authoring and Playback System by ITRI (5) Action! By Macromedia, Inc. (6) Audio visual connection by IBM (7) Astound by Gold Disk Inc.

Authorware uses an event control flow diagram allowing the presenter to specify presentation objects and control, which can be decomposed into several levels in a hierarchy structure. The system also provides a simple script language for calculation and data manipulation. Other systems also provide script languages and API functions. Hypermedia System and Action! Use a time line table allowing actions or objects be dropped in particular time slot. Most systems allow users to cut and paste presentation objects or actions via button click and drawing. Multimedia Viewer also provides a set of medium editing tools. Presentation objects produced by these tools can be linked together by a script language supporting functions, data, structure, and commands.

None of the above system, however, allows dynamic presentations. That is, a presentation generated by the above systems will stay as the form it was created. Different audience watches the same presentation over and over again. Not many systems focus on the web-based presentation designs. Our system helps the presentation designers to design his/her presentation step by step until he/she published the presentation in the air.

This paper is organized as follows. Section 2 proposed the architecture, operations and the transferring messages of our system. Section 3 demonstrated our implementation. Section 4 gives the conclusions.

2. System Development

2.1 Technical Support

In our system, we used the Windows Media Codecs [16] to be the media streaming solution. Windows Media Codecs for creating advance stream format (ASF) content use compression/decompression algorithms (codecs) to compress audio and/or video media, either from live sources or other media formats, to fit on a network's available bandwidth. The ASF is a data format for streaming audio and video content, images, and script commands in packets over a network. ASF content can be an .asf file or a live stream generated by Windows Media Encoder. ASF content that is in the process of being delivered over a network is called an ASF stream. Some features that ASF supported were described as follows:

Audio: mandatory in authoring and rendering, using uncompressed files or files compressed with the Windows Media Audio, Sipro Labs ACELP,MPEG-3, or another codec.

Video: mandatory in authoring and rendering, using uncompressed files or files compressed with the MPEG-4, TrueMotion RT, or ClearVideo codec.

Digital Rights Management (DRM): is the technology for securing content and managing the rights for its access. It is optional in authoring and mandatory for rendering.

Script Command: Script commands instruct Microsoft Windows Media Player to perform additional tasks, called Script, along with rendering the ASF stream. Script commands can be used with both live ASF streams and stored .asf files. Script commands can be added to live streams through Windows Media Encoder and added to stored files through either Windows Media ASF Indexer or the command-line utilities.

2.2 A Multiple Level Content Tree for Abstraction

Given a web-based multimedia presentation, the corresponding multiple level content tree can be constructed, as shown in Figure 1. A teaching material can be taken as a multimedia presentation (e.g. collection of text, video, audio, image…etc.) with some kinds of sequence fashion. The multiple level content tree approach may be used to arrive at an efficient summarizing method.

A content tree is a finite set of one or more nodes such that there is a particularly designated node called the root. The *level* of a node is defined by initially letting the root be at level 0. If a node is at level q, then its children are at level $q+1$. Since a node is composed of a presentation *segment*, the *siblings* with the order from left to right represent a presentation with some sequence fashion (as illustrated by Figure 2). The higher level gives the longer presentation. Consequently, this approach gives flexible teaching material; accordingly, it is very fit for the web-based multimedia presentation.

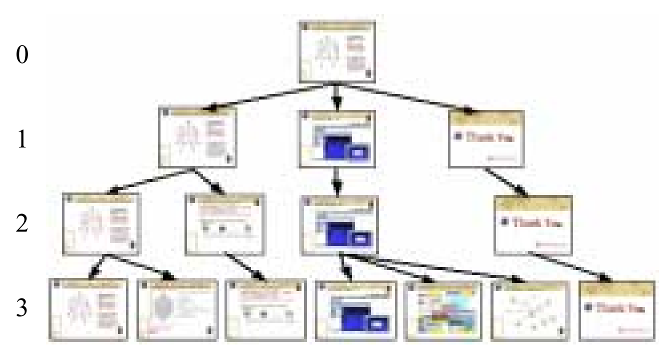


Figure 1. An example of multiple level content tree.

Segment	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9
Level	0	1	1	2	3	3	2	3	1	1

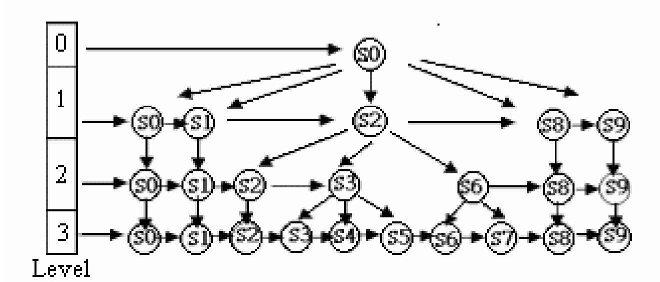


Figure 2. A well-defined multiple level content tree.

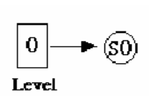
The Abstractor utilizes the content tree to organize the information, as shown in Figure 1. There are a number of primitive operations that can be applied to the content tree. Algorithms to initialize a content tree, attach a node, detach a node, and calculate the presentation time at a given level, have been developed. The content tree serves as an internal data structure for all these operations.

2.3 An example of building a tree as below parameter:

Segment	S0	S1	S2	S3	S4
Level	0	1	2	2	1

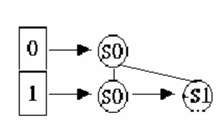
Consider the above user specified parameter of certain presentation segment and its level. After we perform the following four steps operations, we will get the user specified content tree.

Step1:add S0



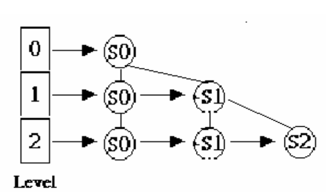
highestLevel = 0;
LevelNodes[0]-> value = 20;

Step 2: add S1



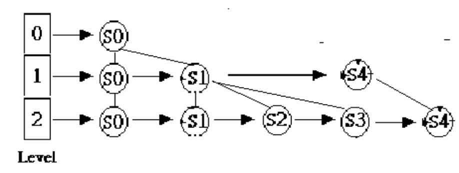
highestLevel = 1;
LevelNodes[1]-> value = 40;

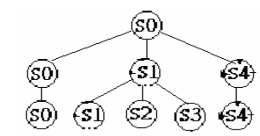
Step 3: add S2



highestLevel = 2;
LevelNodes[2]-> value = 60;

Step 4: add S4

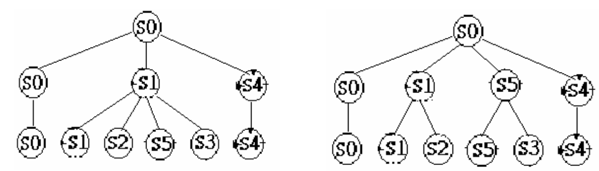




highestLevel = 2;
LevelNodes[1]-> value = 60;
LevelNodes[2]-> value = 100;

2.4 An example of inserts a node to the content tree:

Segment	S0	S1	S2	S5	S3	S4
Level	0	1	2	1	2	1

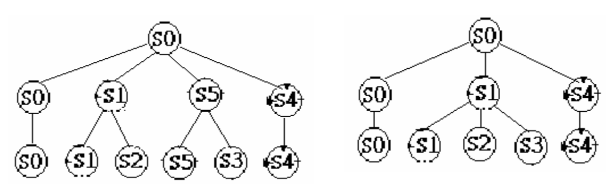


(a) (b)

Figure 3: Action of insert a node S5 (level 1) into the content tree.

highestLevel = 2;
LevelNodes[0]-> value = 20;
LevelNodes[1]-> value = 60;
LevelNodes[2]-> value = 120;

Figure 4 illustrates the deleting a node S5. Figure 4(a) is the original content tree. When we perform the operation with "deleting the S5 node", the S5's children will be adopted by S5's *siblings* S1, as shown in Figure 6(b).

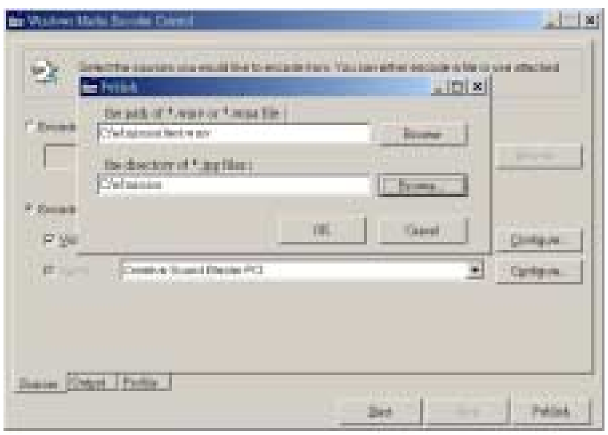


(a) (b)

Figure 4. Action of deleting a node S5 (level 1).

2.5 System User Interface

The major of the system user interface consists of the configuration module. The configuration module provides the user with the facilities to select the sources/devices who would like to encode from and to select how you want to output your encoded content. User can either encode a media file (video/audio) or use attached devices (video camera or microphone) to produce the orchestrated media contents. Another, user can select either broadcast their encoded content in real time after finished configuring the server HTTP port and the URL for Internet/LAN connections. User can select the profile that best describes the content you are encoding. This profile means the different bandwidth will be configured. The more high bit rate means the content will be encoded to a more high-resolution content. The different bandwidth profile selection window was also allowed.



(a) Fill the path in the form for publishing

(b) replay the representation

Figure 5. A web publishing manager

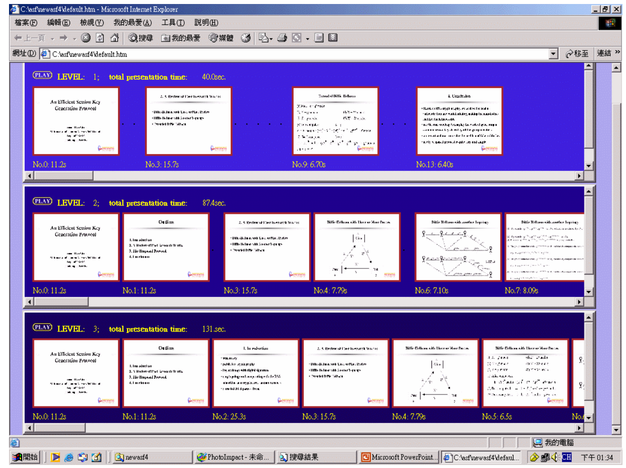


Figure 6. An Multi-level content tree of the web-based multimedia presentation

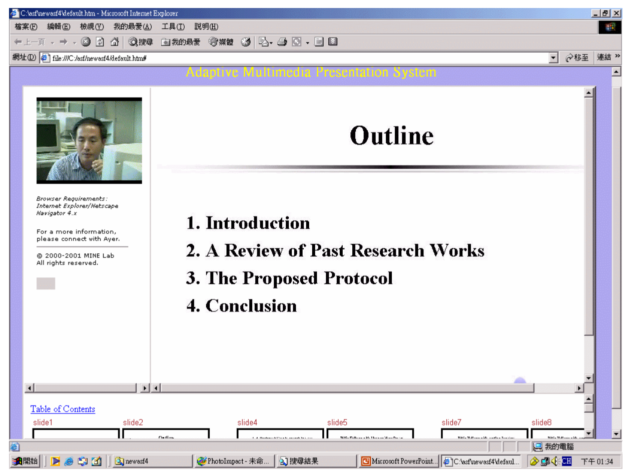


Figure 7. An example of Presentations

Figure 5, 6, and 7 showed the publishing operations. User must fill the path of video file (MPEG4) and the directory of the presented slides. Our system could make the video and presented slides synchronized with the temporal script commands as an advanced stream format (ASF) file automatically. When user replayed the presentation by media player, the orchestrated ASF file will show the video and the presented slides.

4. Conclusion

We not only proposed a framework of web-based multimedia presentation system (WMPS) but also implemented this system. We use the extended timed Petri Net as the medias synchronization model. With the easy to use interfaces, the configuration and the

Proceedings of the 22 nd International Conference on Distributed Computing Systems Workshops (ICDCSW'02)
0-7695-1588-6/02 $17.00 © 2002 **IEEE**

operation steps of the WMPS are clear and definite. The main goal of our system is to provide a feasible method to record and represent a lecture/presentation in the distributed environment. Using the browser with the windows media services allows users to view live video of the teacher giving his speech, along with synchronized images of his presentation slides and all the annotations/comments. The system can be used for general purposed presentations as well as distance learning, advertisement, and others.

References

[1] Tadao Murata, "Petri Nets: Properties, Analysis and Applications," Proceedings of The IEEE Vol. 77, No. 4, 1989.

[2] J. L. Peterson, "Petri Net Theory and the Modeling of Systems," Englewood Cliffs, NJ: Prentice-Hall, Inc. 1981.

[3] Richard Zurawsi and MengChu Zhou, "Petri Nets and Industrial Application: A Tutorial", IEEE Transactions on Industrisl Electronics, Vol. 41, No. 6, Dec. 1994.

[4] T. D. C. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects," IEEE Journal on Selected Areas in Communications, pp.413-427, Apr. 1990.

[5] M. Woo, N. U. Qazi and A. Ghafoor, "A Synchronous Framework for Communication of Pre-orchestrated Multimedia Information," IEEE Network, pp. 52-61, Jan./Feb. 1994.

[6] S. Crespi-Reghizzi and D. Mandrioli, "Petri nets and szilard languages," Information and Control, Vol. 33, No. 2, pp. 177-192, Feb. 1977.

[7] E. W. Mayr, "An algorithm for the general Petri net reachability problem," SIAM, J. Comput. Vol. 13, No. 3, pp. 441-460, Aug. 1984.

[8] Toshinori Suzuki and Sol. M. Shatz, "A Protocol Modeling and Verification Approach Based on a Specification Language and Petri Nets," IEEE Transactions on Software Engineering. Vol. 16, No. 5, May 1990.

[9] M. A. Holiday and Mary K. Venon, "A generalized timed Petri net model for performance analysis," IEEE Trans. Software Eng., vol. SE-13, no. 12, pp.1297-1310, Dec. 1987.

[10] J. L. Peterson, "Petri nets," ACM Computing Surveys, vol. 9, no. 3, pp. 223- 252, Sept. 1977.

[11] G. A. Schloss and M. J. Wynblatt, "Presentation Layer Primitives for the Layered Multimedia Data Model," In Proceedings of the IEEE 1995 International Conference on Multimedia Computing and Systems, May 15-18,Washington DC, pages 231-238, 1995.

[12] R. H. Thomas and e. a. Harry C. Forsdick, "Diamond: A Multimedia Message System Built on a Distributed Architecture," IEEE Computer, December: 65-78, 1985.

[13] Sheng-Uei Guan, Hsiao-Yen Yu and Jen-Shun Yang, "A Prioritized Petri Net Model and its Application in Distributed Multimedia Systems," IEEE Transaction on Computer, Vol. 47 No. 4, April 1998.

[14] T. D. C. Little and A. Ghafoor, " Multimedia Synchronization Protocols for Broadband Integrated Services," IEEE Journal on Selected Areas in Communications, pp1368-1382, Dec. 1991.

[15] T. D. C. Little, "Synchronization for Distributed Multimedia Database Systems," Ph. D. Dissertation, Syracuse University, Aug. 1991.

[16] www.microsoft.com "Windows Medias SDK", 1999.

[17] M. Vazirgiannis and C. Mourlas. "An Object-Oriented Model for Interactive Multimedia Presentations". The Computer Journal, 36(1), 1993.