Regular Paper

A Unified Media Synchronization Methods for Dynamic Hypermedia System

Michiaki Katsumoto,† Naoya Seta†† and Yoshitaka Shibata†††

Hypermedia systems are very important as platforms for the multimedia information network service spaces in the future. In order to provide more flexible and sophisticated multimedia presentations, it is desirable to simply and efficiently transmit various media data from databases distributed over a network and present them on users' stations depending on a customized presentation scenario so as to semantically integrate them into multimedia information objects. In this paper, we introduce a new hypermedia system that utilizes an object-oriented structure to perform two different temporal synchronization methods, **lip synchronization** and **scene synchronization**. The lip synchronization method performs fine-grained temporal synchronization between continuous media, such as audio and video. In the scene synchronization, a coarse-grained temporal relationship between continuous media and discrete media, such as image, graphics or text is maintained depending on the presentation scenario. Furthermore, several media which are semantically related to each other are unified into a Media Group (MG) to simplify their media treatment. The MGs and their involved processing functions are encapsulated into a Media Object (MO) and are easily controlled by a message passing procedure. In this paper, we describe in detail implementation of a hypermedia model which performs scene synchronization functions between discrete and continuous media in the MOs, and context switching functions between each media presentation. Finally the Electric Museum as a prototype application of hypermedia systems is introduced to evaluate the suggested synchronization mechanism.

1. Introduction

In general, hypertext structures or systems are very useful to organize integrated context-based information because all information units are directly connected to each other and users can simply navigate over the linked information space. So far, we have developed several prototype systems for multimedia information networks to establish common and unified system architectures and verify their network protocol hierarchies, human interfaces, database management systems, possible applications and end-to-end performance[1)~3)] for our suggested platform. In these prototype systems, we introduced a Dynamic Hypermedia System (DHS) in which the links from the current reference points to the suitable objects among the multimedia database distributed over the network are dynamically determined depending on the user's background, interests and preferences. Eventually, we achieved a more flexible and simpler user's access.

However, in order to realize more realistic and general multimedia information systems, a more sophisticated media structure for object nodes in hypermedia which takes account of several different degrees of temporal relations and context switching functions is required. In this case, **multimedia structure** and **hypermedia structure** must be clearly defined. According to previous papers[4),5)], **hypertext structure** is organized by node and links, whereas, **multimedia structure** is organized by a combination of both continuous and discrete media. Furthermore, the **hypermedia structure** is organized by an extended hypertext structure model in which each node has multimedia structure. However, the definition of multimedia structure in the paper[4),5)] is not sufficient because there is no clear definition of temporal synchronization between continuous media, such as audio and video, and continuous media and discrete media, such as image, graphics and text with presentation scenario. Media transmission over a network depending on presentation scenario while keeping the temporal relation between them is also not considered.

On the other hand, in another paper[6)], although coarse-gained temporal relation depending on presentation scenario is taken into

† Communication Research Laboratory of Ministry of Posts and Telecommunications
†† Toyo Industrical Machinery Co. Ltd.
††† Faculty of Engineering, Toyo University

consideration with scenario, more fine-grained synchronization for hypermedia system is not considered. Furthermore, since the links between the current reference points to the next are fixed and cannot be easily changed once the hypermedia system has been constructed. As results, a **hypermedia structure** with fixed link leads to a lack of flexible or effective user navigation capabilities and a lack of expandable database construction.

In order to overcome these problems, we introduce a new network oriented DHS which is based on object-oriented structure and takes into consideration of different temporal synchronization methods depending on intramedia or intermedia data distributed over the network. So far, we have developed several different media synchronization methods and their architectures for continuous media, such as video and audio for Video-on-Demand (VOD) application[7),9),11)]. We applied these techniques for the hypermedia structure to perform different temporal synchronization methods, **lip synchronization** and **scene synchronization**. The lip synchronization method performs fine-grained temporal synchronization between continuous media, such as audio and video. On the other hand, in scene synchronization, coarse-grained temporal relation between continuous media and discrete media, such as image, graphics or text is maintained depending on the presentation scenario.

Furthermore, several media which are semantically related to each other are unified into a Media Group (MG) to simplify its media treatment. The MGs and involved processing functions are encapsulated into a Media Object (MO) and easily controlled by message passing. In this paper, implementation of hypermedia model which performs scene synchronization function between discrete and continuous media in the MOs depending on the scenario by message passing, and context switching functions between each media presentation are precisely described. Finally the "Electric Museum" as a prototype application of hypermedia systems is introduced to evaluate the suggested synchronization mechanism.

2. Dynamic Hypermedia System

Dynamic Hypermedia System (DHS) is a network oriented platform for multimedia information networks to provide multimedia information space based on hypermedia structure.

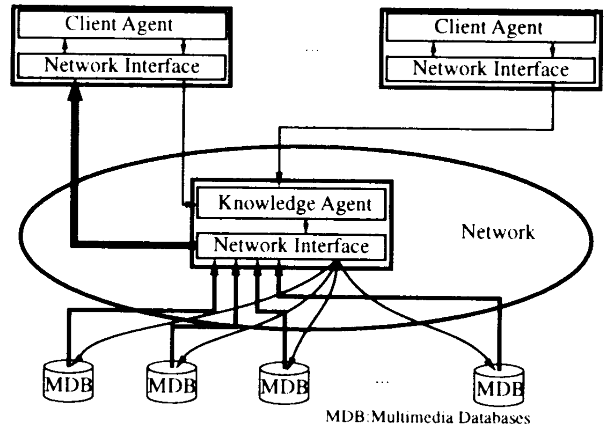


Fig. 1 Architecture of dynamic hypermedia system.

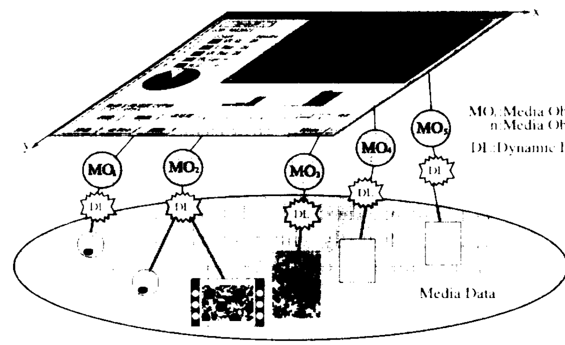


Fig. 2 Example of media objects.

The system architecture of DHS is composed of three components as shown in **Fig. 1**: Client Agents, a Knowledge Agent and a number of Multimedia databases. The Client Agents are located on user stations to provide multimedia presentation capabilities for each user. The Knowledge Agent manages links to information units by dynamic linking method[8)] and generates multimedia objects. The multimedia databases (MDB) manage multiple Media Objects, such as text, images, video and audio data as shown in **Fig. 2**. Since more details of the DHS are explained in[1),2),8)], we will not discuss them in this paper.

3. Synchronization Model

We classify media synchronization into two models; intramedia synchronization and intermedia synchronization.

3.1 Intermedia synchronization

Two different intermedia synchronizations are defined: lip synchronization and scene synchronization. In scene synchronization, the starting points of each media which mean the beginnings of their presentation times are controlled. As an example shown in **Fig. 3** (a),

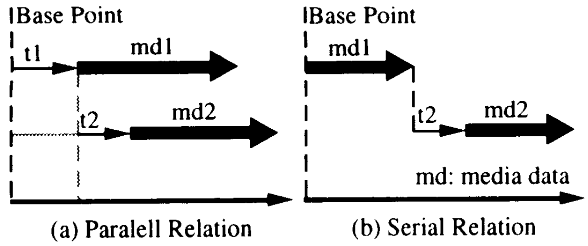


Fig. 3 Illustration of scene synchronization.

when Media1 and Media2 have a parallel relationship to each other, then both are started after $t1$ [sec] and $t2$ [sec] from the original time, respectively. On the other hand, as shown in Fig. 3 (b), when Media1 and Media2 have a serial relationship to each other, then the Media1 is started after $t1$ [sec] from the original time, and Media2 is started after $t2$ [sec] from the end of the Media1. In lip synchronization, a tightly temporal relation between media data, such as video and audio is maintained. This lip synchronization can also be applied in the case where synchronization between more than three videos or audio is required. We will explain more details of lip synchronization in Section 5.

3.2 Intramedia Synchronization

In order to present continuous media, although their frame rates must be maintained at constants, their actual frame rates may be influenced by the random load conditions of the machines or network. In order to resolve this problem, we introduced the following QoS control functions particularly for video presentation: time-priority control and content-priority control. The time-priority control is applied in the case where more precise time interval and more accurate playback speed on a video presentation is required. If the frame rate is incidentally decreased by the increase of network traffic or machine loads, then the video frame is periodically sub-sampled so as not to extend the current video period and to maintain real-time video presentation.

Content-priority control is applied for those cases where significant content meaning is lost when sub-sampling for video frame rate is used. For example, when the video includes an important event which instantaneously finishes, such as a motion video with instantaneous hitting of baseball game. If these video frames are sub-sampled, the most important parts may be lost. In this case, all of the video frames should be presented as slow motion video even though the actual presentation time is expanded compared to the real-time presentation.

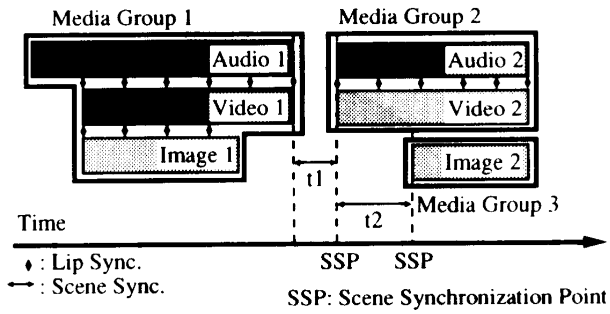


Fig. 4 Illustration of media group.

Frame rate regulation control is applied for continuous media to maintain constant frame rate in video or the constant output rate in audio as set by the QoS controller or application. This rate regulation control is always performed in both the time-priority and the content-priority controls. Since an attainable effective rate is existed for any load condition, the system can always maintain the selected frame rate constant.

4. Multimedia Media Integration

4.1 Media Group

In order to easily deal with more precise temporal synchronization, a concept of media group is introduced. For an example in the scenario in **Fig. 4**, lip synchronizations between Audio1, Video1 and Image1, and between Adio2 and Video2 must be maintained. The pair of Audio2 and Video2 also must be started after $t1$ [sec] from the completion of a pair of Audio1 and Video1. Furthermore, the Image2 also must be displayed with Audio2 and Video2 after $t2$ [sec] from the beginning of both Audio2 and Video2. If these media are independently processed, then the synchronization control for all of the six media has to be simultaneously taken for scene synchronization. Lip synchronization of Audio1 and Video1, or Audio2 and Video2 must be considered as the same level as scene synchronization. Therefore the synchronization of the whole presentation becomes complicated and eventually accurate media synchronization becomes very difficult. On the other hand, if all of these media are controlled only by lip synchronization, then the start times for the media presentation must be individually maintained. Eventually, it is need to redundant lip synchronization control.

Whereas, by grouping a set of related me-

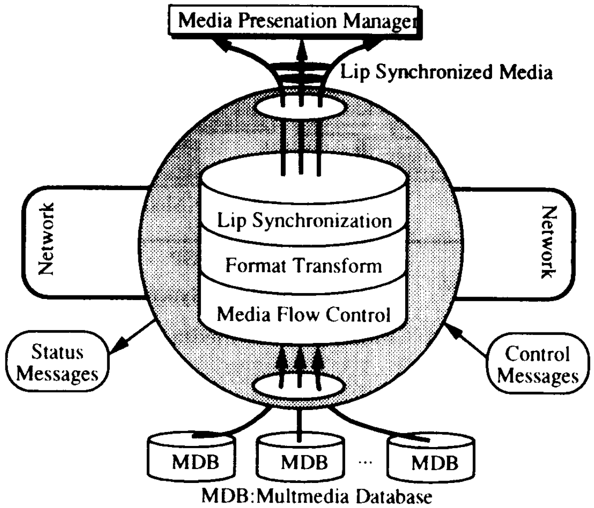


Fig. 5 Illustration of media object.

dia into a media group, this media group can be regarded as a single media. As the result, we only need to consider at most three media (Media Group1–3) in synchronization schedule (Fig. 4) and lip synchronization need only be made within the same media group (Media Group1 and Media Group2). Thus, the number of the media to be synchronized can be reduced and scene synchronization control can be simplified. It is possible to set up any synchronization interval for lip synchronization to each media object. Furthermore, if media objects are implemented by light weight processes or threads on multiple CPU architecture, then the presentation capability of the media group will be more improved. These lip synchronization functions are executed at the **synchronization layer** in the **Media Object** which is explained in the following section.

4.2 Media Object

In order to easily realize scene synchronization by message passing, we defined a Media Object (MO) as a data model which takes account of network transmission. As depicted in **Fig. 5**, a MO includes a Media Group and a set of methods to process the Media Group. These methods can perform media transmission protocols[9] including synchronization function, media format conversion function and media flow control for the streams. In synchronization function performs synchronization between various media. Media format conversion function performs media conversion, compression/decompression of audio and video. Media flow control function perform packet rate control media to avoid overrun on the receiving and sending buffers. Therefore, lip synchronization between the multiple media which organize the MG while keeping effective packet transmission can be attained at the MO's synchronization layer. Thus, By definition of the MO, more unified media transmission and presentation control can be realized whatever these media types may be.

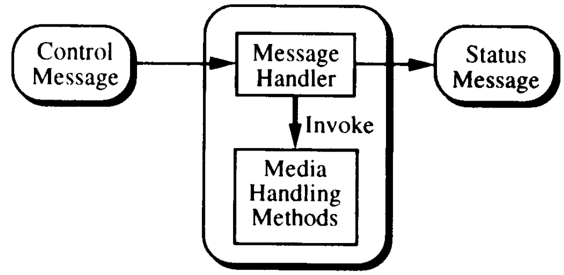


Fig. 6 Implementation model of media object.

4.3 The implementation of MO

The method part of the MO as depicted in **Fig. 6** can be separated into the Message Handler and the Media Handling Methods. The advantage of this separation is that control of the MO can be easily realized by adding the Message Handler to the existing packet audio/video and image transmission and display system function in PAVS[9]. The Message handler module performs as an external interface to the MO, receives the control messages asynchronously, calls the proper method and returns the state message. The Media Handling Methods module includes a set of methods to manipulate media data, such as media stream transmission, media format conversion and lip synchronization processing. It is reminded that the different methods are activated depending on the type of the media data which organize the MO.

5. Realization of Media synchronization

5.1 Lip Synchronization

In order to realize more fine-grained synchronization, we introduced relaxed synchronization[9]. With the relaxed synchronization method, synchronization is executed only once at several video frames and equivalent audio frame points as shown in **Fig. 7**. For example, when the synchronization interval is smaller than 1 [sec], and the video and audio frame rates are defined as N_P [frames/sec] and N_A [segment/sec], respectively, synchronization is executed at every time T which is calculated by N_P and N_A divided by α. Here α is a

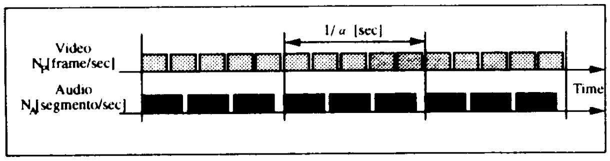


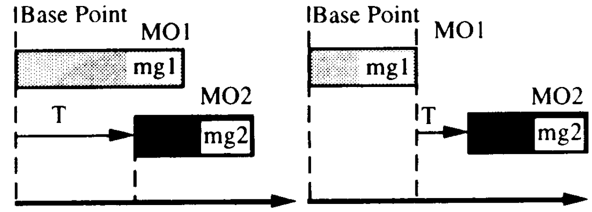
Fig. 7 Illustration of relaxed synchronization.

common divisor between N_P and N_A. In this case, the theoretical synchronization interval is $1/\alpha$ [sec]. When the synchronization interval is larger than 1 [sec], synchronization is executed at every time point which is a multiple integral number of N_P or N_A. In this case, the synchronization interval is α [sec]. If $1/\alpha$ is too strict and more coase-grained synchronization is required, then the synchronization interval $(1/\alpha)n$ can be used where n is an integral number.

The advantage of this method is that a relatively flexible synchronization rate is quickly achieved, and a higher throughput[10),11)] is maintained while the number of synchronization points and the processing load on the client station are reduced.

5.2 Scene Synchronization

Scene synchronization based on the scenario of multimedia presentation is realized by the synchronization mechanism in the multimedia controller as explained in the next section among the MOs. Now we consider the case where the two media groups, mg1 and mg2, which are organized by the MO1 and MO2 as indicated in **Fig. 8** (a), respectively, and have parallel relation to each other. In this case, a start control message is sent to the MO1 to activate the specified method at the beginning of the synchronization point. After T [sec] from the synchronization point, another start control message is sent to the MO2. Thus scene synchronization is realized. On the other hand, in the case where the two media groups mg1 and mg2, which are organized by MO1 and MO2, respectively, and have a serial relation to each other as is indicated in Fig. 8 (b), a start control message is sent to the MO1 for mg1 to activate the specified method at the synchronization point. After T [sec] since a termination state message from MO1 has been received, another start control message is sent to MO2. Thus by confirming the termination of MO1 and starting MO2, the serial relation between both media objects can be maintained. This scene synchronization can be executed by the multimedia controller as discussed in the following section.



(a) Paralell Relation (b) Serial Relation

Fig. 8 Illustaration of media object synchronization.

5.3 Multimedia Controller

A pair of multimedia controllers (MMC) as shown in **Fig. 9** are activated for media presentation at both client and agent and process the scene synchronization based on the presentation scenario. The MMCs perform management/control of all of the MOs which organize the presentation. The client MMC can perform scene synchronization functions using message passing based on the presentation scenario. In Fig. 9, the client MMC receives condition messages from all of the MOs and controls their actions, such as start/stop/termination by taking account of all of the media states. The client MMC also manages the elapsed time of the presentation based on the scenario. The client MMC issues control messages to the MOs when it detected time event activated by the user, and shifts the current time to the user's required presentation point. The process on the client MMC can be summarized in the following steps;

(1) acquisition of the scenario;
(2) activation of the MOs required for the presentation;
(3) scene synchronization control by message passing;
(4) control of the related MOs depending on user's event;
(5) termination of MOs for completion of the presentation.

5.4 Implementation of MMC

The client MMC consists of the three modules: MO Initiator, Scene Synchronization Controller and Message Receiver. The MO Initiator activates the required MOs before starting multimedia presentation. The Scene Synchronization Controller module performs scene synchronization by message passing to the related MOs based on the scenario. In this case, this module refers to the condition tables of the MOs on the shared memory, issues starting control messages to the related MOs after confirming the

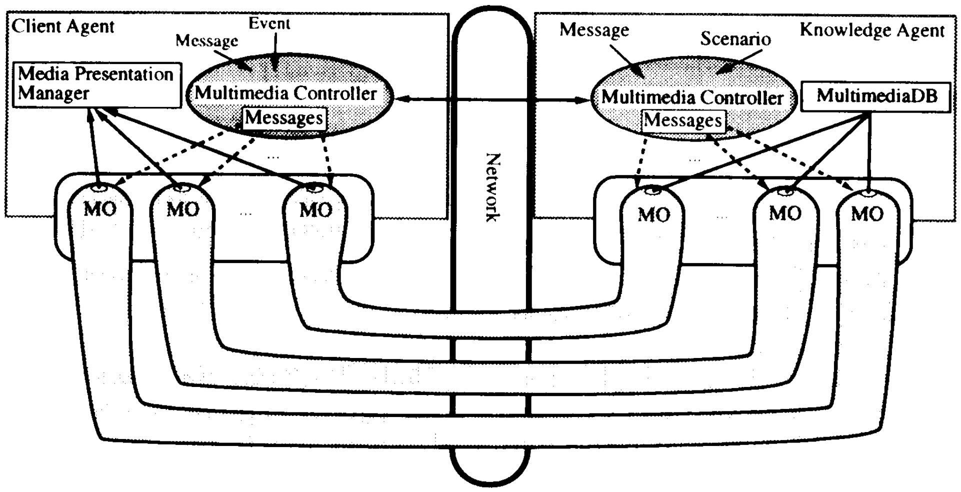


Fig. 9 Architecture of multimedia controller.

related MOs are ready to be processed. This module also confirms the termination of the presentation by maintaining this condition table of the MOs. The Message Receiver module receives various messages which are sent asynchronously from each MO and writes the condition of each MO on the condition table of the shared memory. Thus, the Scene Synchronization Controller can always maintain the state of the MOs.

6. Context Switching

In the hypermedia structure, a number of multimedia presentations are provided on the same output display at the client workstation. Therefore, context switching functions between presentations are required. We introduce three context switching functions; pause, continue and quit in the following:

Pause: as shown in **Fig. 10** (a), Multimedia presentation A is initiated at $t1$. Then Multimedia presentation B is initiated at $t2$ while Multimedia presentation A continues. Then Multimedia presentation A is paused until Multimedia presentation B has been completed. When Multimedia presentation B is finished, Multimedia presentation A is resumed.

Continue: as shown in Fig. 10 (b), while Multimedia presentation A is activating, Multimedia presentation B is initiated at $t2$ without stopping Multimedia presentation A. Thus two presentations are concurrently presented.

Quit: as shown in Fig. 10 (c), while Multimedia presentation A is activating, Multimedia presentation B is initiated at $t2$. Then Multimedia presentation A is stopped and instead the Multimedia presentation B is started.

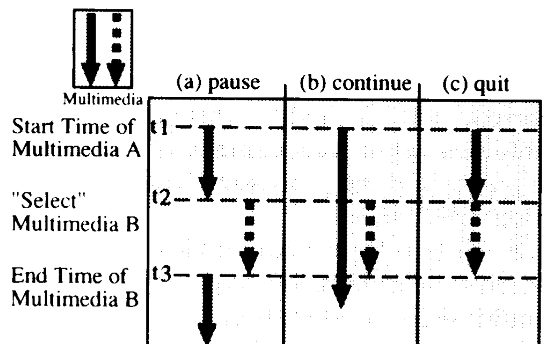


Fig. 10 Illustration of hypermedia contexts swiching.

These context switching parameters are basically embedded in the presentation scenario, as produced by multimedia authors, and managed by dynamic hypermedia space.

In order to realize these context switching functions, the control of MMCs as a unit of multimedia presentation is required. For this reason, we have introduced a hypermedia controller and realized such context switching by message passing for the MMC as described previously.

6.1 Hypermedia Controller

A pair of hypermedia controllers (HMC') for both client and agent are activated to manage the MMC' and to control the context switching functions described in the scenario using message passing. The client HMC receives the state message from the MMCs and executes context switching, such as pause/continue or quit. **Figure 11** indicates a network model with HMC including MMC'. Thus, the process of the client HMC from the beginning of the hypermedia

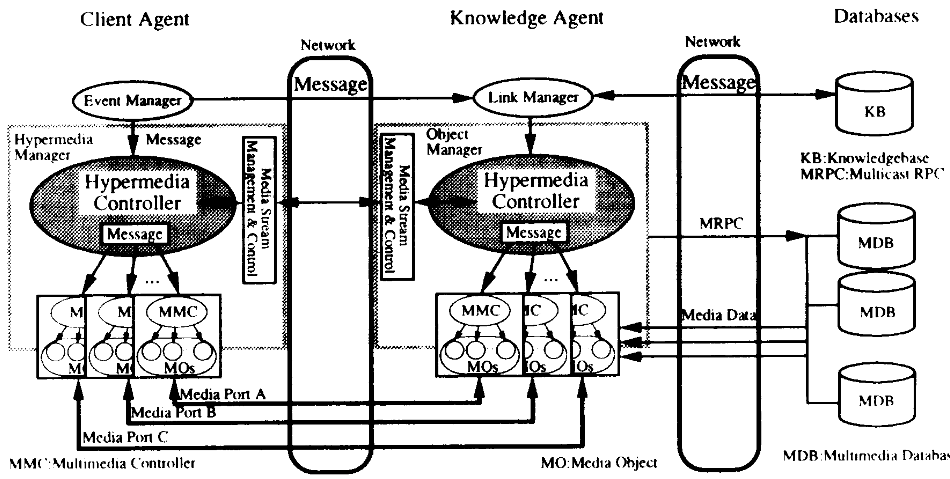


Fig. 11 Structure of hypermedia controller.

presentation to the end is shwon in the following:

(1) acquisition of the name of the scenario from the agent HMC;
(2) notification of the name of the scenario after activating MMC;
(3) control of MMCs depending on user's event;
(4) termination of MMC when the presentation is completed.

6.2 Implementation of HMC

As indicated in Fig. 11, the client HMC consists of three modules; MMC Initiator, Hypermedia Context Controller and Message Receiver. The Message Receiver activates a new MMC when every multimedia presentation is initiated and provides the required scenario for the presentation. The Hypermedia Context Controller executes the required context switching by sending a message based on the time scheduled scenario. At this time, the HMC refers to the condition table on the shared memory, confirms that the objective MMC is ready to be activated at the switching context time and sends the control message. The HMC also confirms the termination of the presentation by checking the condition table. The Message Receiver module receives various messages which are sent asynchronously from each MMC and writes the condition of each of these MMCs on the condition table in the shared memory. Thus, the hypermedia controller can always maintain the state of the MMCs.

7. Prototyping

In order to evaluate the functionalities of the MMC and HMC described so far, we prototyped a dynamic hypermedia system using several RISC-based workstations; one is for an agent and others are for clients over the FDDI network with distributed multimedia database are emulated in the agent. An example of the application, at the Electronic Museum "Kawagoe City" is indicated in **Fig. 12** which introduces the history, tradition, culture of an old Japanese town, Kawagoe City by hypermedia presentation was developed using the X-window system libraries and the C language. **Figure 13** indicates the scenario example of the Electronic Museum Kawagoe City. On this application, users can more interactively and directly retrieve multimedia information based on the hypermedia concept concerned with Kawagoe City using iconic and mouse operations. All of the multimedia information based on the presentation scenario were displayed on the multi-windows based monitor with audio, motion video, image, graphics and text. The contents presented dynamically vary by person-to-person depending on the users' interests, background and preferences.

In Fig. 12, when a user starts client, then a pair of HMC processes are activated on both the client and the agent, and a map with Kawagoe City is displayed in a window. By clicking on the location of the town where the user is interested, a query for the desired information with the objective town is issued to the multimedia database in the agent. Then scenario ID of the location is sent to the HMCs on both the client and the agent. Next, the HMCs obtain a scenario and activate a pair of MMCs on both the client and the agent. Each HMC sends a "start" message to the equivalent MMC. The

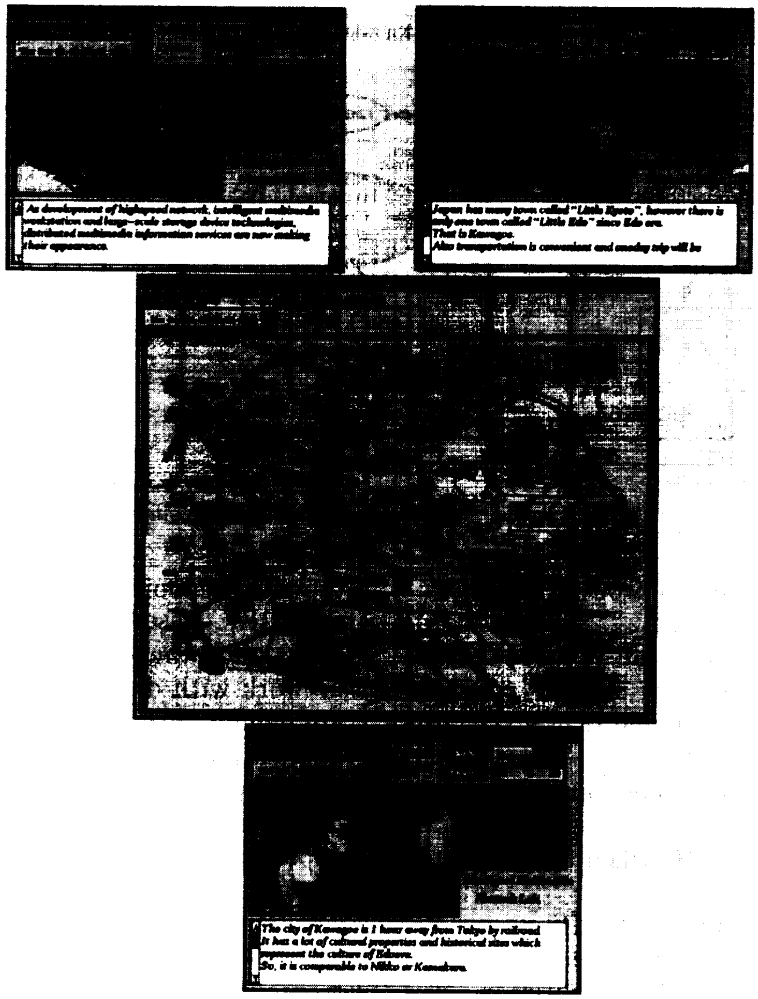


Fig. 12 Presentation example of Electronic Museum "Kawagoe City".

MMC receives the message and starts the multimedia presentation organized by a combination of video and audio, images and text data concerned with the rough shape of old Japanese vehicle in the town as shown in the top left window of Fig. 12. This multimedia presentation is characterized by the scenario structure as shown in the top left of Fig. 13. After that, by clicking the location of the object interest (the old vehicle) in the displayed multimedia presentation, the next multimedia presenation starts with a context switching function. At the same time, each HMC obtains a new scenario and checks the context switching parameter of the first media presentation to identify the type of context. In this case, since the context is pause, each HMC sends a "pause" message to the equivalent MMC for the first multimedia presentation and activates a new multimedia presenation concerned with the more details of the old vehicle and its actual activity at the festival while the first multimedia presenation is paused as shown in the top right window of Fig. 12. This multimedia presentation is characterized by the scenario structure as shown in the top right of Fig. 13. When this presentation comes to an end, the MMC sends a finish message to the equivalent HMC and completes this presentation. Then the HMC received finish message sends a restart message to the newest MMC which is being paused. This newest MMC then resumes its multimedia presentation.

The bottom middle window in Fig. 12 is organized by an image data displayed on top left, video data on top right, text data on bottom and audio data played from the audio speaker. This multimedia presentation is characterized by the scenario structure as shown in the bottom middle in Fig. 13. In this case, lip synchronization is realized by the relaxed synchronization method with a 125 [msec] synchronization interval to maintaine between audio and video based audio data. On the other hand, the im-

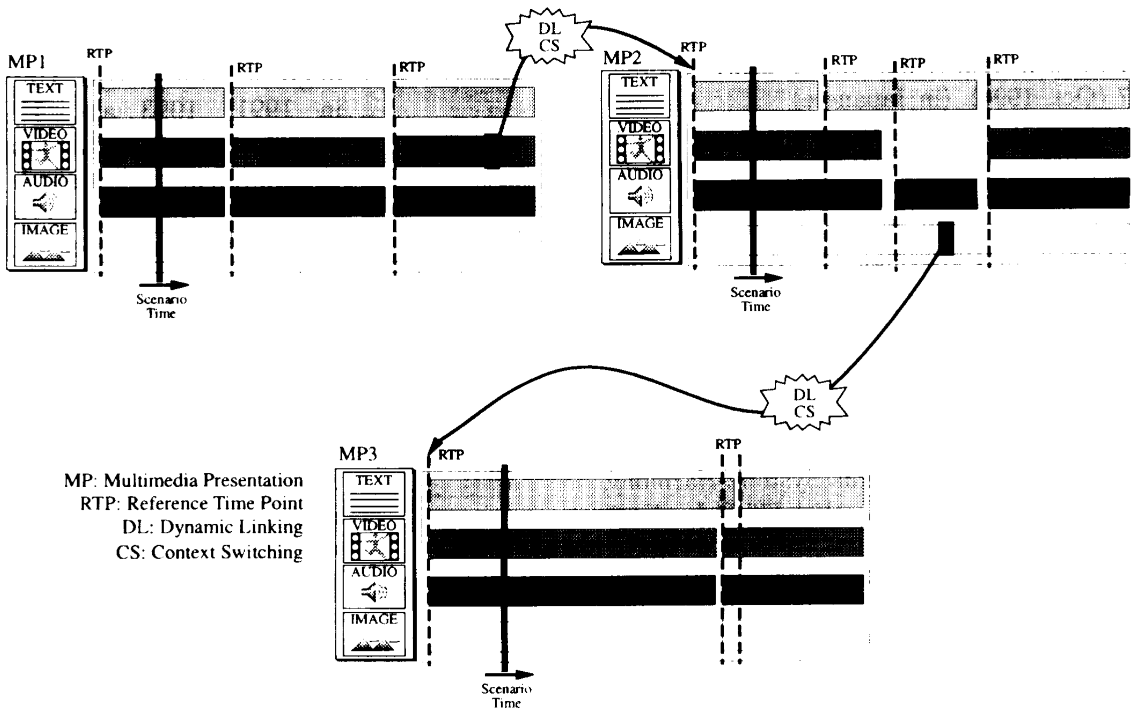


Fig. 13 Scenario example of Electronic Museum "Kawagoe City".

age, the video and the text data are shown according to a scenario by the scene synchronization at the beginning. Next, new image data and video data are presented by the scene synchronization after 15 seconds from the beginning. Furthermore, new text data are displayed by the scene synchronization after 17 seconds from the beginning.

Through this prototype, we could confirm that the functions of the scene synchronization and the lip synchronization could correctly perform and provide satisfactory synchronization accuracy on a multimedia presentation even though multiple windows are simultaneously displayed. However, the response of the context switching was not necessarily satisfactory, because the changing time from one multimedia presentation to another took more than 1 [sec]. From this result, light weight processes or threads are required to implement HMC and MMC to improve the response time of context switching.

8. Conclusions

In this paper, we defined scene and lip synchronizations between intermedia and output control between continuous and discrete intramedia for our DHS. We also introduced the MG and MO to realize more unified media integration and precisely explained how to implement MOs. Furthermore, as control functions of the scene synchronization and the context switching, the implementation of the MMC for MO and HMC for multiple MMCs was described. A prototype of the suggested hypermedia system was built to evaluate the functions introduced in this research using the Electronic Museum application. As a result, both lip and scene synchronization could be accurately realized. Three different context switching functions may also be effectively performed to link the current multimedia structure to the following one with a moderate delayed interval. Now we are extending our hypermedia system by introducing more flexible QoS control functions, such as dynamic resource allocation of CPU for each MMC and HMC, and network bandwidth reservation for continuous media.

References

1) Shibata, Y. and Katsumoto, M.: Dynamic Hypertext and Knowledge Agent Systems for Multimedia Information Networks, *Proc. ACM Hypertext '93*, pp.82–93 (Nov. 1993).
2) Katsumoto, M. and Shibata, Y.: Dynamic Hypermedia System Using Knowledge Agent for Multimedia Information Networks, *Proc. JWCC-8*, pp.C2.2–C2.8 (Dec. 1993).
3) Hirose, N., Katsumoto, M. and Shibata, Y.: Retrieval Method and Performance Analysis of

Distributed Database System, *IPJS Multimedia Communication and Distributed Processing System Workshop*, Vol.94, No.1, pp.289–297 (Oct. 1994). (in Japanese).

4) Hardman, L., Bulterman, D.C.A. and Rossum, G.: Links in Hypermedia: the Requirement for Context, *ACM Hypertext '93*, pp.183–191 (Nov. 1993).
5) Hardman, L., Bulterman, D.C.A. and Rossum, G.: The AMSTERDAM Hypermedia Model: Adding Time and Context to the Dexter Model, *ACM Hypertext '93*, pp.183–191 (Nov. 1993).
6) Fujikawa, K., Shimojo, S.: Synchronization Mechanism among Multimedia Information on Distributed Hypermedia System: Harmony, *Trans. of The IEICE*, Vol.J76-D-I, No.9, pp.437–483 (Sep. 1993).
7) Hashimoto, K., Katsumoto, M., Watanabe, M. and Shibata, Y.: QoS Guarantee Functions for Continuous Media Services, *IPSJ SIG Notes*, Vol.95, No.61, pp.97–102 (July 1995), (in Japanese).
8) Katsumoto, M., Fukuda, M., Irie, N. and Shibata, Y.: Dynamic Hypermedia System based on Perceptional Link Method for Distributed Design Image Database, *Proc. ICOIN-9*, pp.49–54 (Dec. 1994).
9) Shibata, Y., Seta, N. and Shimizu, S.: Media Synchronization Protocols for Packet Audio-Video System on Multimedia Information Networks, *IEEE Proc. 28th HICSS*, pp.594–601 (Jan. 1995).
10) Seta, N., Simizu, S. and Shibata, Y.: Synchronization Methods for Packet Audio-Video System, *IPSJ SIG Note*, Vol.94, No.106, pp.19 24 (Mar. 1994).
11) Akama, T., Watanabe, M., Hashimoto, K. and Shibata, Y.: Dynamic Packet Rate Control for Packet Audio-Video Systems, *IPSJ SIG Note*, Vol.94, No.106 (Dec. 1994). (in Japanese).

(Received October 4, 1995)
(Accepted February 7, 1996)

Michiaki Katsumoto received the B.S., M.S. and Ph.D. degrees from Toyo University in 1991, 1993 and 1996 respectively. He is working now Communication Research Laboratory of Ministry of Posts and Telecommunications. His research interests include hypermedia system and multimedia database. Hi is a member of IPSJ, IEICE, IEEE Computer Society and ACM.

Naoya Seta received the B.S. and M.S. degrees from Toyo University in 1993 and 1995 respectively. Fróm 1995 to 1996, he worked at Hitachi Information Network Inc. Currently he is a technical designer of Toyo Industrical Machinery Co. Ltd., where he designs computer oriented industrial robotics. His research interests include media synchronization and hypermedia system. He is a member of IPSJ and IEICE.

Yoshitaka Shibata received his Ph.D. in computer science from the University of California, Los Angeles (UCLA) in 1985. From 1981 to 1985, he was a doctoral reasearch associate in the Computer Sceince Department, where he engaged in software development of an array processor for high-speed simulation. From 1985 to 1989, he was a research member in Bell Communication Research (former Bell Laboratory), where he was working in the area of higher-layer protocol design and end-to-end performance analysis of multimedia information services. Since 1989, he was an associate professor of Information and Computer Sciences Department in Toyo University, where he conducts an intelligent multimedia network laboratory. His research interests include intelligent human interfaces, hypermedia systems, multimedia databases, high-speed networks and protocols.