

LAPCHAT: A Contents-Sharable Management System for Computer Supported Collaborative Learning

Mizue Kayama and Toshio Okamoto

Senshu University, School of Network and Information

Graduate School of Information System, The University of Electro-Communication

Abstract: The purpose of this study is to explore the architecture for a collaborative learning environment, in which individual learning and collaborative learning are smoothly connected. We proposed a composition model of a collaborative workplace, and a management model for action in the collaborative learning space and the state of learning context. Based on the ideas, in this paper we describe a Contents-Sharable mechanism between a private workplace and a collaborative workplace.

Keywords: Collaborative learning, learning contents sharing, contents-sharable.

1. INTRODUCTION

In this study, we explore the design of a collaborative learning environment which is realized by the seamless linkage of individual learning and the collaborative learning. A composition model of the collaborative workplace and a management model for learning activity and learning condition have been proposed.

In this paper, the concept of “Contents-Sharable” and a learning system for managing Contents-Sharable are proposed. The sophisticated mechanism for contents sharing, which realize a smooth linkage between learning contents in the private/collaborative workplace, is described. To support these linkages a learning tool, the “Lapchat”, which supports learning in the ICT environment, is described. In this tool, the user controls various kind of information by unification of a type of image. Preparation of the electronic notebook in the ICT environment for individual learning/collaborative or learning/lecture is realized in an OS independent application.

Please use the following format when citing this chapter:

Kayama, M. and Okamoto, T., 2007, in IFIP International Federation for Information Processing, Volume 230, Knowledge Management for Educational Innovation, eds. Tatnall, A., Okamoto, T., Visscher, A., (Boston: Springer), pp. 33–42.

2. THE ARCHITECTURE MODEL OF THE COLLABORATIVE LEARNING FRAMEWORK

In this paper a classification for the architecture of a collaborative learning framework is studied. The MVC (Model–View–Controller) pattern is utilized as a method of the classification. In an attempt at classification, comparisons from the following viewpoints are tried.

- A realization of the synchronous / asynchronous collaborative learning workplace (a private workplace and a collaborative workplace).
- The convenience of adaptation to the educational situation.
- The network traffic load.
- The simplicity of the packaging.

The collaborative learning framework is classified as 4 types based on the management form of the MVC pattern.

2.1.1 Centralized Architecture

The first type is a structure of centralized architecture. In the Centralized Architecture type, only one original application is run on the specific client terminal. All original part (Model, View and Controller) will exist in that specific terminal. The operation event for the original application is transmitted to other terminals. Then, the original GUI (= the View and the Controller) is copied to the specific terminal, and delivered to other terminals. The frameworks that have adopted this architecture are synchronous collaborative learning support systems. Many of these systems have a desktop sharing function. The main feature of this architecture is to work the terminals which run the same OS. The screen image and the operation for any application can be shared between these terminals. However, broadband networks are required in the utilization. The system cannot work when there is a network failure in order to control the whole management by only one terminal.

2.1.2 Replicated Architecture

The second type is a structure of replicated architecture. In this type, all modules related to an application are installed in each client terminal. The original MVC exists in all terminals. Application of this architecture is implemented using the API (Application Programming Interface) for collaboratively utilizing the application. By utilizing the peculiar API, the synchronous function between applications on the each client is secured. The examples of this type are Habanero (Jackson 1999), MatchMaker (Zhao & Hoppe 1994), SAILE (Goodman, Geier, Haverty, Linton & McCready 2001), SIMPLE (Plaisant, Rose, Rubloff, Salter & Shneiderman 1999), MediaFusion. In the Replicated Architecture type, the necessary network bandwidth decreases. In addition, the client software can be use as a stand-alone application.

2.1.3 Distributed Architecture

The third type is a structure of distributed architecture. In the Distributed Architecture type, MVC is dispersed to multiple hosts. A Model is put to a server. Each View and Controller is allocated to each client: the View/Controller of the client will manipulate the Model on the server. A concrete system of this type is a web site with a database, for example CSILE/Knowledge Forum (Bereiter 1997). The web browser works as the View and the Controller. The server with Model manages data from the Controller and the data for the View. In an application of this type, the View and the Controller are actually constructed on the server. Then, they are transmitted to the client in the execution. This type has common features with the Centralized Architecture type. Additionally, as well as the Replicated Architecture type, the necessary bandwidth is also narrow. The consistency of the Model is also easy to keep. However, the system cannot be utilized as well as the Centralized Architecture type, when there is a failure in the network.

2.1.4 Hybrid Architecture

The fourth type is hybrid architecture. This type is combination of the other forms. For example, a form that combined the Replicated Architecture and the Distributed Architecture was considered. In this type, the synchronization between clients is realized by utilizing the eternal Model on the server (Constantino-gonzalez, Suthers & Santos 2003; Kayama & Okamoto 2002). The application is formed by the MVC on the each user terminal. The condition of the application is preserved in the local file system, when this system works by using each Model as stand-alone. The external Model on the server is utilized for the case of collaborative learning activity. The renewal of operation result of each GUI is guaranteed between clients who login to the server. This type can be utilized even in the situation of failure in the network. However, in this type, the multiple clients are able to renew the common Model. It is necessary to consider data loss consistency in the application implementation.

3. CONTENTS-SHARABLE MANAGEMENT SYSTEM FOR COLLABORATIVE LEARNING

We have explored the design and implementation of a collaborative learning framework. As a result, a learning framework which supports synchronous/ asynchronous collaborative learning using a Hybrid Architecture was proposed. Features of this framework were the integrated mechanism for learning resources sharing, and an event data management mechanism for the record/replay/reference of the collaborative learning situation (Kayama & Okamoto 2002). In order to develop learning resources

for this framework, the engineer was required to change his software code by using the specific API. In this framework, the operation history and maintenance of the application's internal state (= the learning condition) were guaranteed by the learning management server. Applications which were rebuilt by using the specific API become the learning resources for our framework. This work is an obstacle for development.

Then a more sophisticated collaborative learning tool was proposed. The features of this tool are to guarantee natural individual learning and to realize the easy linkage and easy management of the group/individual learning result. In this tool, the sharing of learning contents in the private workplace and the collaborative workplace is realized in the form of a contents "image". In this study, this function is named "Contents Sharable".

3.1 The Structure of the Contents-Sharable System

The proposed system is implemented as a Hybrid Architecture that combined the Replicated Architecture and the Distributed Architecture. The outline is shown in figure 1. To each learner terminal, each MVC element is offered. A Model as a eternal object is implemented on the server. The operation by the learner terminal (the Controller data) broadcasts to other learner's terminals via the server. Transmitted information is a set of the Controller data. The Model data is not transmitted. Therefore, in this type, the network load does not rise in comparison with the other Hybrid Architecture. The function of the Contents-Sharable does not depend on the platform of the private workplace and the collaborative workplace. The Model on each learner terminal individually reproduces the condition of the learning situation through the JNI (Java Native Interface) layer. This function works in any user terminal platform, which is able to run the JVM (Java Virtual Machine).

The object structure of the Lapchat is as follows. The client communicates with the server through the connector object. These communications have been realized by distributed object programming. The server involves the Web server. The communication between clients and

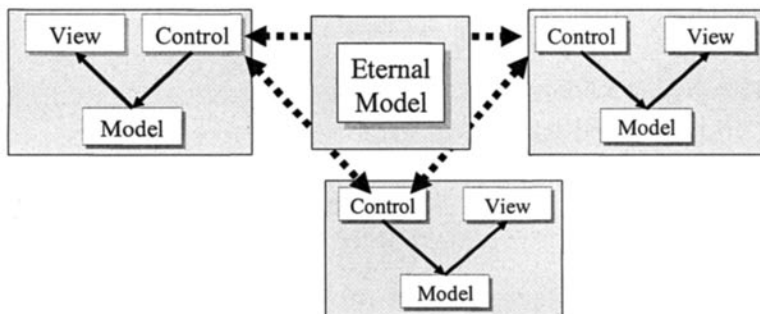


Figure 1. MVC Architecture of the Proposed System

download of the shared contents from the server are done through Horb or HTTP. The user accounts are managed in the server. The user belongs to a group. The learning contents are shared between group members. The user freely accesses the shared contents of the group to which he belongs. The user makes his learning contents in his private workplace. This user is able to upload his learning contents to the collaborative workplace. In addition, the shared contents of group A are loaded in a learner's private workplace, and then this content is able to be loaded in the workplace of group B by that learner. The Share object manages the shared contents. The group has multiple Share objects. The user who belongs to the group creates a Share object. Contents controlled in a Share object are stored in the Repository object. The Repository object controls not only the shared contents but also the relation information such as their attributes and the access permissions to them.

3.2 Design Concept

The Lapchat is a leaning tool to support preparation (information collection/arrangement) /open / shared / exchange of an electronic notebook in the ICT environment. This tool does not disturb the ICT environment which a learner daily utilizes for learning. Simultaneously, this tool offers a sophisticated mechanism which realizes uniform information management. Features of this tool are shown in the following:

- OS independence in of the client environment.
- The independent of the application utilized by the client terminal.
- Independent in the type of contents of a management object.

In the Lapchat, the method and process of information generation / discovery, and operation of the application which an individual learner utilizes are not shared. Especially, the following are important:

- Security/privacy by the learner terminal,
- Guarantee of the ownership for each learning contents.

Sharing of application operations at the Meta level has been realized. The individual who runs an application carries out the operation to that application. Then, these learners make open this result. In the Lapchat, the learner shares only the changed result.

3.3 Contents-Sharable Management

The Lapchat works in a client environment with a window interface (such as Windows, MacOS, Linux and so on). The image in the window that the learner designates is uniformly controlled as image information. In addition, postscript data (drawing and character, etc.) for the image information is controlled as the other image layer. By this, only the addition and deletion of the postscript data are enabled, in the condition that the image information was ensured in the lower layer. The management object by the Lapchat is shown in the following:

- a) Contents image in a window interface on client desktop,
- b) Image data in the client terminal,
- c) A URL reachable image data,
- d) Drawing and character string added to the data of a) ... c),
- e) Postscript /annotation of a) ... d). (generation day, creator, comments, title, etc. are contained) and
- f) Discussion information for the a) ... e).

This information management is done by a hierarchical log data management method (Kayama & Okamoto 2002) which was proposed as a function of a collaborative learning framework (Kayama & Okamoto 2004).

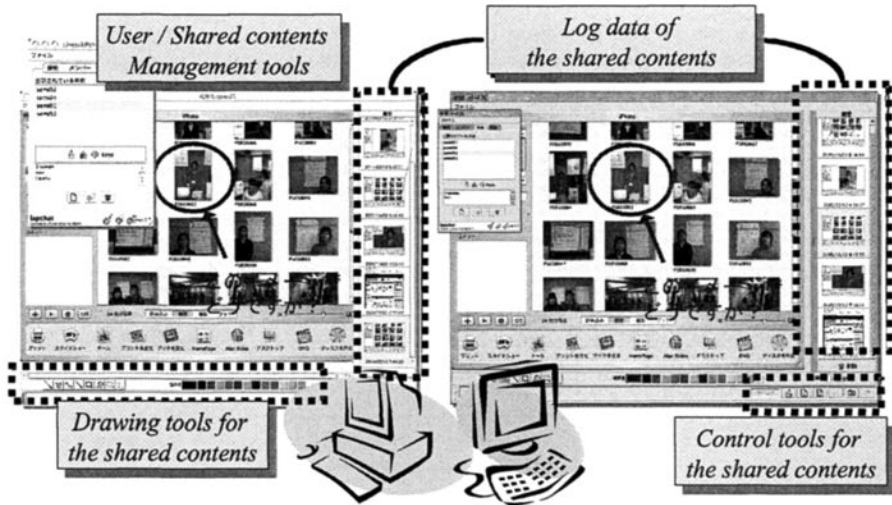


Figure 2: Interfaces of the Lapchat

3.4 Functions

The interface image of the proposed tool is shown in figure 2. For any type of learning (individual learning, collaborative learning or lecture) the following five functions are provided.

1. Persistence attribute of the collaborative workplace.
2. Linkage of the individual learning contents to the group learning contents.
3. Drawing for the shared contents,
4. Maintenance for the history log of the shared contents
5. Operation permission for the shared contents history.

3.4.1 Persistence attribute of the collaborative workplace

In this function, a user can set the externalization/ deletion permissions of shared contents which are formed in the collaborative workplace. These permissions are set by a dialog shown on the left top part of figure 2. The externalization of contents means leaving the learning result on the server after the learning activity is finished. By making contents into externalization, the reference and the additional change to the shared contents become possible, even if after the learner logouts from the collaborative workplace. When a user does not set the contents as externalized, the shared contents are automatically discarded at the end of the learning. The deletion permission means to decide the propriety of the shared contents deletion by the group member.

3.4.2 Linkage of individual contents to groups

The individual learning contents which are able to be uploaded to the collaborative workplace are as follows.

- Window image on the learner's desktop (includes the steal image for the dynamic application),
- URL reachable information,
- Image file in the learner's terminal,
- The previous shared contents.

To upload the individual contents to the collaborative workplace, the learner uses the right bottom part of the figure 2.

3.4.3 Drawing for shared contents

The drawing of the shared contents is carried out by the left bottom part of figure 2. In the Model, the shared contents and the drawing data for it are distinguished. Each of them separately managed. Therefore, it is possible to realize renewal only of the shared contents or renewal only of the drawing data.

3.4.4 Shared contents log management

The log data of the shared contents with their drawing data is retained in the Model. The history data is made to be the shared contents again. Then loading the current/past shared contents by individual learner to his private workplace is possible.

3.4.5 Operation permission for the shared contents history

Operation permission for the shared contents history is as follows:

- Addition of shared contents to the history list,
- Uploading of the other history data to the collaborative workplace,

- Modification of uploaded data,
 - Deletion of the history data.
- Group management is controlled by accessibility to the shared contents on the server.

3.5 LEARNING with the Lapchat

An outline of learning by Lapchat is shown in figure 3.

3.5.1 Individual Learning

The Lapchat client is started as a local connection (local://), and it is made to be the individual learning mode. In this mode, the 1st learner constructs his electronic notebook. Five kinds of information which are shown in (a), b), ... e) are controlled by repository for the individual (PR:Personal Repository). PR is constructed in the local directory (Removable Disk, Hard Disk, Floppy Disk, etc.), which the learner designated. PR is transportable. Then, with the Lapchat client, a learner can continue his learning without choosing the learning environment.

3.5.2 Collaborative Learning

It becomes the collaborative learning mode, when the Lapchat client is started in server connection (laph://). Synchronous collaborative learning is possible when other learners login to the Lapchat server at the same time. Equivalent permission to access the shared contents is given to all learners. Each operation in the identical timing has been realized. Asynchronous

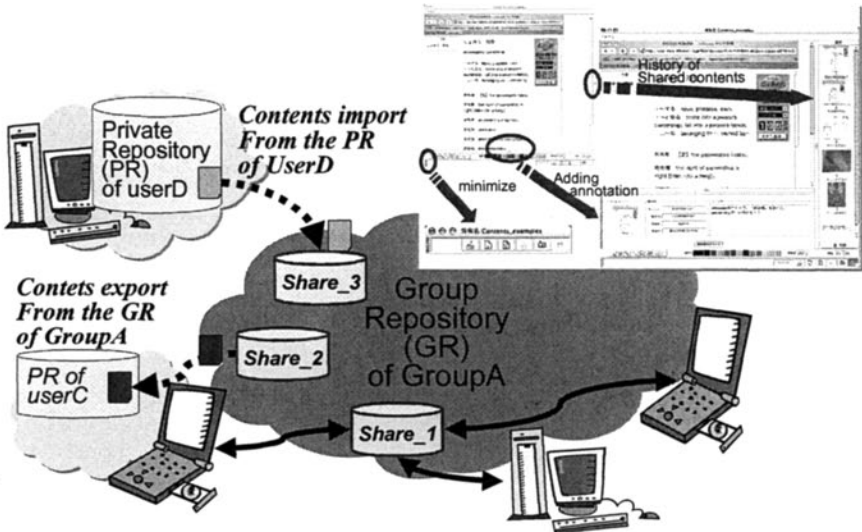


Figure 3 Outline of the learning with the Lapchat

learning will begin when another learner does not exist at the server in the same timing. In collaborative learning mode, six kinds of information which are shown in (a), b), ... f) are controlled by a repository on the Lapchat server for the group (GR:Group Repository). In this mode, the following actions become possible, whether synchronous/asynchronous learning type.

- Construction of the electronic notebook as a collaborative work in multiple learners (GR),
- Opening of the content of individual RP of the group member,
- Saving of information of GR to individual RP.

3.5.3 Lecture

A lecture is realized by controlling the shared contents in GR in the collaborative learning mode. The restriction for the shared contents means the following five kinds of setting.

1. Drawing permission to the base image of the lowest layer,
2. Change permission of base image,
3. Addition permission of the history of the shared contents,
4. Deletion permission of the history of the shared contents,
5. Correction/change permission of the history data attributes.

The learner, who has a teacher role, makes the electronic notebook in his PR at the individual learning mode. Then, the learner opens to public his note from his PR to the GR. In the collaborative learning mode, the learner makes all shared restrictions OFF. When his group members participate in learning at the collaborative learning mode, the lecture is started. In the lecture situation it becomes possible that the learner preserves the shared contents, which the teacher presents in his PR. By writing annotations additionally to the preserved contents in his RP electronically note-taking is realized.

4. CONCLUSION

In this paper, the Lapchat, which supported learning in the ICT environment, was proposed. The Lapchat realizes the construction / open / shared of the electronic notebook. It is independent of OS and applications for learning in the client environment. Also, the concept of Contents-Sharable is proposed. A seamless linkage between contents in the private workplace and contents in the collaborative workplace is offered. Exploration of a group model based on the analysis of the postscript/annotation information will be tried in future.

5. REFERENCE

- Kayama, M. & Okamoto, T. (2002) Collaborative learning in the internet learning space, *International Journal of Industry and Higher Education*, IEEE, August, pp.1-11.
- Kayama, M. & Okamoto, T (2004). PLATFORM AND FUNCTIONAL MODEL FOR COLLABORATIVE LEARNING, *Advanced Technology for Learning*, Vol.1, No.3, pp.139-146.
- Suthers, D. (2001). Architectures for computer supported collaborative learning., *IEEE International Conference on Advanced Learning Technologies*, pp.6-8.
- Jackson, L. (1999). Concurrent Engineering in Construction -Challenges for the New Millennium-, *Proc. of the 2nd Int. Conf. on Concurrent Engineering in Construction*, pp 37-46.
- Zhao, J. & Hoppe, H.U. (1994). Supporting flexible communication in heterogeneous multi-user environments, *Proc. of the 14th IEEE Int. Conf. on Distributed Computing Systems*, pp.422-229.
- Goodman, B., Geier, M., Haverty, L., Linton, F. & McCreedy, R. (2001). A Framework for Asynchronous Collaborative Learning and Problem Solving, <http://downloads.openchannelsoftware.org/SAILE/SAILE2001.pdf>.
- Plaisant, Rose, A., Rubloff, Salter & Shneiderman (1999). The Design of History Mechanisms and Their Use in Collaborative Educational Simulations, *Proc. of the 3rd Int. Conf. on Computer Support for Collaborative Learning*, pp.348-359.
- Bereiter (1997). Situated cognition and how to overcome it, In D. Kirshner & J.A. Whitson (Eds.) *Situated cognition: Social, semiotic, and psychological perspectives*, pp.281-300.
- M.A.Constantino-gonzalez, Suthers & Santos (2003). Coaching Web-based Collaborative Learning based on Problem Solution Differences and Participation, *Int. J. of AIED*, Vol.13, pp. 263 - 299.