User-oriented Operational Interface for Virtual Learning Environment

Tomoko Kojiri,[†] *Yasuki Ito*[††]*, and Toyohide Watanabe*[††]*,*

[†]Information Technology Center, Nagoya University
[††]Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Aichi, 464-8601, JAPAN

Summary
In this paper, we propose the operational interface which changes windows' sizes and locations dynamically according to the student's intention and preference. Students' tasks are executed by manipulating appropriate data through specific windows. Moreover, some tasks are correlated with each other. If windows that students want to use are allocated in an accessible place, students do not have a burden to manipulate the windows. In order to represent student's intention and preference of performing the task, logical window network is generated. Logical window network represents the student's interest for each task and windows that are used for accomplish the task. In the logical window network, nodes correspond to windows/tasks and links show relations among nodes. When a student changes the active window, the system detects its related windows based on the logical window network and rearranges windows in the interface automatically, in order for the student to access the interesting window easily. In this paper, the mechanism of constructing a logical window network is described. Also, the effectiveness of proposed interface is shown based on the experimental results using the prototype system.

Key words:
Human interface, modeling student's intention, logical window network, automatic movement of windows

1. Introduction

With the development of computer and communication network, students tend to study on computers using various learning support systems [1]. The process of achieving the learning goal is decomposed into several small tasks. Students determine appropriate learning tools for realizing each task and change tools dynamically. In the real world, students arrange learning tools, such as dictionaries or textbooks, around them so as to promote each task effectively. On the other hand, in the virtual learning environment, students select appropriate applications for individual tasks and activate them in the right order to achieve their learning objectives. However, since the workspaces in the interface are restricted, students need to rearrange windows every time they change tasks. In addition, because the interface of the application is sometimes composed of many complicated windows, students cannot concentrate on the learning and cannot acquire the knowledge from the learning systems efficiently. Therefore, it is important to provide the effective interface that makes students concentrate on the knowledge or subject that they are studying and not on the computer artifact they are using [2].

There are many researches that provide the real world-like interface. Card, et al. proposed 3 dimensional book-like interfaces for HTML pages, such as WebBook and WebForager [3]. These interfaces introduce book metaphor to represent the structure of a collection of related web pages. They also realize intuitive movement of pages in a book, such as flipping, to help finding interesting pages. On the other hand, Robertson, et al. constructed the 3 dimensional window manager called The Task Gallery [4]. This interface is designed as the virtual art gallery and the user's tasks are managed like the artworks hung on the walls. This interface is designed for providing users familiar perception and cognition for managing the tasks. However, these researches only create the real world-like view in the interface and students still need to manipulate the applications/windows by themselves.

In the real world, students do not always see objects as their physical configurations, but may focus on specific objects in order to understand them deeply. Some researches provide additional information that is not able to be acquired by observing only physical aspects of objects. For example, Carroll, et al. aimed at facilitating the collaborative work by providing various kinds of awareness information [5]. In this research, awareness necessary for collaborative work is classified into 3 levels. Then, at the right timing appropriate students' working states are shown as the awareness information. Yamada, et al. focused on key-stroke-acts of the users in the on-line chat environment [6]. Their system called TangibleChat leads the user to grasp the situation of the companies even they do not share the same physical space. Kojiri, et al. also introduced the personal information in the collaborative learning environment to make student grasp other student's situation [7]. These researches give additional information that is not represented by the physical configuration. However,

Manuscript received June 5, 2006.
Manuscript revised June 25, 2006.

users still need to select and manipulate objects in the interface by themselves in order to execute the task.

The interface corresponds to eyes of students to the virtual learning environment. So if the interface gives the interesting information automatically, they may concentrate on studying rather than manipulating the computer. Kojiri, et al. also focused on the discussion of the collaborative learning and realizes the focusing function of other students [8]. This system determines the student's focusing target from utterances in the discussion and grasps the camera images of the focusing target automatically. However, this system handles only the information of other students. All information in the interface should be controlled according to the student's intention.

In order for students to achieve their learning objectives effectively and acquire the knowledge from the learning support system efficiently, the interface should provide intuitive manipulation that reflects students' intention. Students tend to organize windows in the interface based on the following viewpoints;

- ✧ current task: students want to use windows for achieving the current task, and
- ✧ next or related tasks: students want to see windows of next or related objectives easily.

What is most required for the interface is to make students execute the current task easily. Therefore, windows that satisfy students' requirements need to be organized in order for students to work easily. Our objective is to construct the interface that provides the appropriate information according to the student's intention, such as not only for the current task but also the change of the tasks. The student's intention is modeled as a logical window network based on his manipulation to windows. According to this logical window network, windows for the same task are organized and managed automatically. Then, when students change active window, related windows of the current window are moved to the place where students can manipulate easily.

2. Framework

The complexity of the learning support system sometimes prevents effective learning for the student. If the interface provides information that the student wants to work with, the student can manipulate the system intuitively and may concentrate on the learning itself. The student is conscious of two things while studying with a computer: particular applications that can achieve his task and windows that belong to particular application. When the student wants to attain to his task, he selects appropriate applications that can lead him to achieve the task. If the application consists of plural windows, the student activates one which can realize his task. Therefore, the interface should represent the student's intention in two levels; a task level and a window level.

In addition, the interface should grasp the pattern of student's manipulating windows and provide appropriate windows at a right time. In order to reflect student's pattern of manipulating windows to the organization of the interface, our system consists of two layers: physical layer and logical layer (Figure 1). Physical layer provides the virtual learning space which is like a 3-dimensional seamless room in the real world. The student is able to walk around and arranges windows in the virtual learning space freely. The manipulating windows and relationships among windows are represented based on their locations from the student. The physical layer also provides the task flag which takes the role of the maker in the interface. The task flag functions as separating the area in the virtual learning environment based on the student's intention to tasks globally.

On the other hand, the logical layer constructs the student's mental model based on the student's interaction to the physical layer. Student's mental model represents the student's intention to applications for achieving the objectives and windows for each application. The student wants to arrange windows for the same task around him and windows that may use for the next task in the near place, so the student's mental model represents the relations among tasks and relation among windows, such as sequential relation and simultaneous relation. This model is updated according to the student's

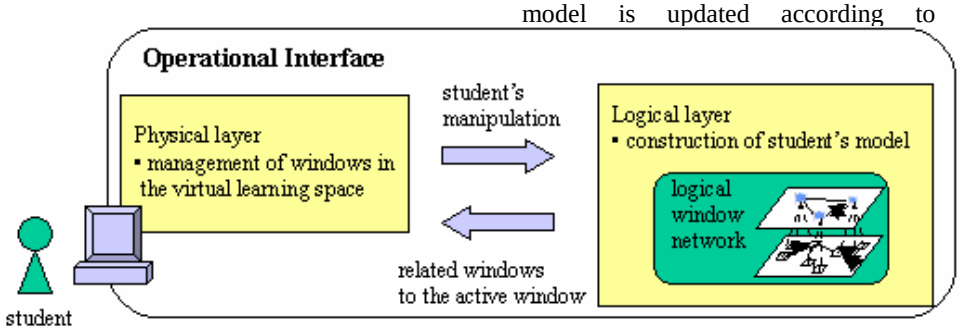


Figure 1; Framework of operational interface

manipulation to the virtual learning environment, and the locations of windows are changed dynamically based on this model.

3. Physical Layer

Physical layer provides the seamless virtual learning space in which plural applications are arranged freely. The student can study with appropriate application by walking in the virtual learning space and finding windows of tools for achieving the task. The virtual learning environment is divided into several areas from viewpoints of the learning tasks by establishing task flags and allocating windows around them.

In the interface, the student's intention or preference to the tasks is grasped from his manipulation to the windows. In the physical layer, the locations of the windows are rearranged dynamically so as to make student study with the computer intuitively. Thus, the student's interesting tasks and windows that are used to achieve the tasks are needed to be grasped. In order to discriminate areas for each task, task flags are provided (Figure 2). A task flag takes the role of a maker for each task. The student separates the virtual learning space using the task flag and allocates related windows that are used for performing the task around it, which creates the working area for each task implicitly. The task flag is also useful for navigating a student to the particular task. Since the virtual learning space is seamless, it is difficult for a student to have a view of the whole learning environment and to find the windows that he wants to use. Therefore, marking the area of the task based on the task flag also promotes a student to handle his own learning space easily.

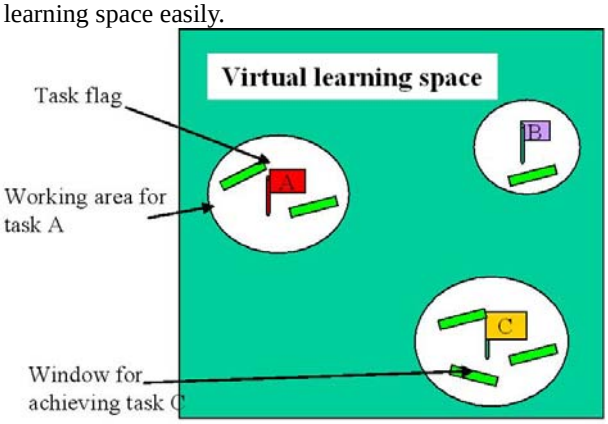


Figure 2: Virtual learning space divided by task flags

A student can walk around in the virtual learning space to find appropriate windows to complete his task. The areas that a student is able to view are defined as two types: a task area and a conscious area (Figure 3). The task area corresponds to the active area in which a student is able to work with the applications. On the other hand, the conscious area is the area whose windows a student can pay attention to, but does not work with immediately. Conscious area is in the back of the task area, so the windows in the conscious area are viewed smaller than those in the task area. Also, a student can only grasp the existence windows in the conscious area, but cannot manipulate the data viewing in the windows. A student moves the windows in the conscious area to the task area to work with it. Table 1 shows the state of windows according to their existence areas.

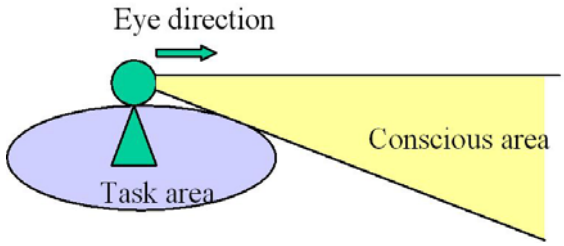


Figure 3: Areas in virtual learning space

In this layer, windows are allocated dynamically according to the student's intention which is grasped in the logical layer. Windows that are related to the current window are moved to the conscious area or the task area according to the relations. If the windows that the student wants to pay attention to are in his view, he does not have to pay extra effort to find the windows of corresponding task and concentrate on the learning itself. Therefore, as the current active window changes, windows in the virtual learning space are rearranged dynamically so as to put windows of interesting tasks near to the student's location.

4. Logical Layer

In the logical layer, student's interest or preference is grasped and modeled as a logical window network based on the student's manipulation of the windows in the physical layer. A student handles windows in the different conscious viewpoints, such as a task and a window. Therefore, the logical window network consists of two layers: task layer and manipulation layer (Figure 4). The task layer represents the student's intention to each task and the manipulation layer corresponds to the student's intention to the individual windows. Since windows are used to achieve the task, these layers are connected by the links that point out the task which windows belong to.

The task layer shows student's intention to the task. It is represented as a network structure in which nodes correspond to the tasks and links show the relations between tasks. Student's tasks are grasped based on the task flags that are created by student. Two kinds of relations are defined as links of the logical window network: concurrent link and sequential link.

Table 1: Windows states and manipulation functions according to existing area

	State	Data manipulation	Change to active state	Change location	Change size
Windows in task area	Active	Yes	---	O	O
Windows in conscious area	Manageable	No	O	O	O
Windows in other area	unmanageable	No	X	O	X

Table 2: Algorithm of creating concurrent link and sequential link

1. Compute T_a, which corresponds to the interval time that window A is activated.
2. If $T_a > T_{th}$, which is a threshold, add 1 to $C_{seq:\ a\text{-}next\ a}$, which is a sequential relation count between window A and the window next to task A. (In Figure 5, add 1 to $C_{seq:\ a\text{-}d}$)
   Else if $T_a \leqq T_{th}$, add 1 to $C_{con:\ a\text{-}between\ a}$, which is a concurrent relation count between window A and the windows in the intervals. (In Figure 5, add 1 to $C_{con:\ a\text{-}b}$ and $C_{con:\ a\text{-}b}$)
3. If $C_{seq} > C_{th}$,which is a threshold, create the sequential link in the logical window network. As the same way, the concurrent link is created when $C_{con} > C_{th.}$

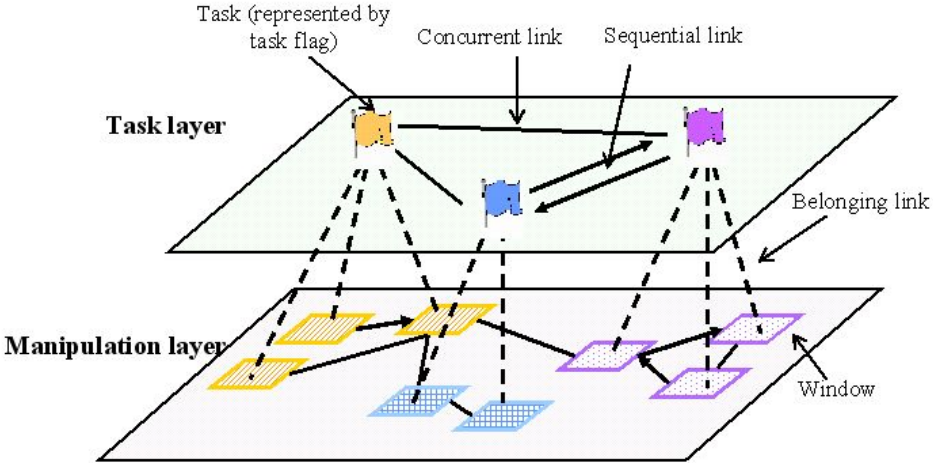


Figure 4: Logical window network

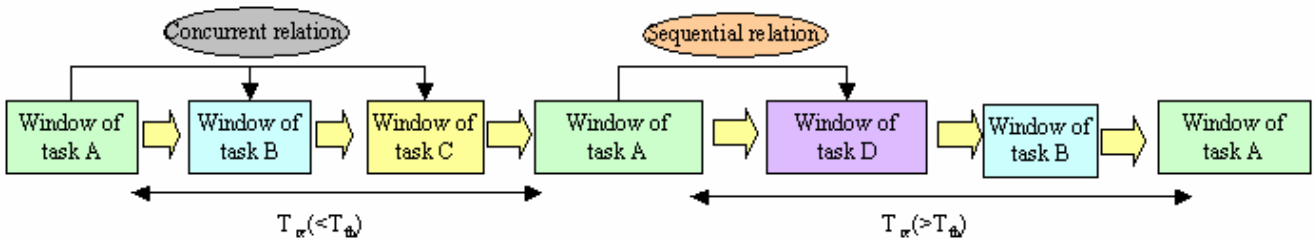


Figure 5: Example of creating concurrent link and sequential link

The concurrent link means that two tasks are executed concurrently and the sequential link indicates that one task may be accomplished next to the other task. These relations are grasped based on the student's manipulation history. Table 2 shows the algorithm of creating links and Figure 5 shows the example of it. Since the student can activate only one window at a time, the relation between tasks is determined by the change of active windows. Sequential relation is defined by the order of activating windows and the concurrent relation is grasped by the interval time of activating the same window.

Manipulation layer is also represented as the network structure. Nodes correspond to windows and meanings of links are the same as those in the task layer. The node in the manipulation layer is linked to the node in the task layer that it belongs to. Since the student defines task areas in the interface based on the task flags, the link is generated according to the distance between windows and task flags that correspond to each task. The window is regarded as the window of the nearest task flag. Equation (1) shows the equation for determining the belonging task of the windows. In this equation, w_a is the

current window and t_b corresponds to the task flag. $|W_aT_b|$ means the distance between the window a and the task flag b. $lw_a\ t_b$ is a belonging weight of the window to each task, which is calculated by dividing the distance between all task flags into the distance between particular task. Then, the window is linked to the task whose $lw_a\ t_b$ is the biggest.

$$l_{w_a t_b} = \frac{1}{\sum_j \frac{1}{|W_a T_j|}} * \frac{1}{|W_a T_B|} \quad \ldots\ldots. (1)$$

This logical window network is updated every time that the student moves the window. Then, based on the logical window network, if the active window has been changed, the windows that relate to current active window are moved to the task area or conscious area in order for the student to work easily.

5. Prototype System

We have implemented the prototype system. The system is implemented by Java3D which is the programming language appropriate for constructing 3 dimensional space. Figure 6 shows the image of our interface. Currently, the screenshots of the windows are arranged in the 3 dimensional virtual learning environment. The student can arrange the windows in the interface, change sizes and locations of them, and select an appropriate window to attain to his task. When the student clicks the screenshot of the window, the active window for the application is emerged and the student is able to manipulate the data of it (see right side of the Figure 6). In the left side of Figure 6, the task flag is shown. Task flag is created by assigning the name of the task and push a create button. Student can move to the task area by pointing out its task flag's name. In addition, the radar view is also provided in order for the student to recognize overview of the virtual learning space easily.

Currently, only the student's intention to the task level is implemented. Namely, all windows for the same tasks are managed equally. Figure 7 shows the movement of the windows that have relations to the current active window. The student wants to use the windows that have the concurrent relation to the current active window simultaneously to the current active window, so they are moved to the student's direction toward the task area. On the other hand, the student wants to use the windows that have the sequential relation to the current active window next to the current window, so they are moved around the student toward the conscious area.

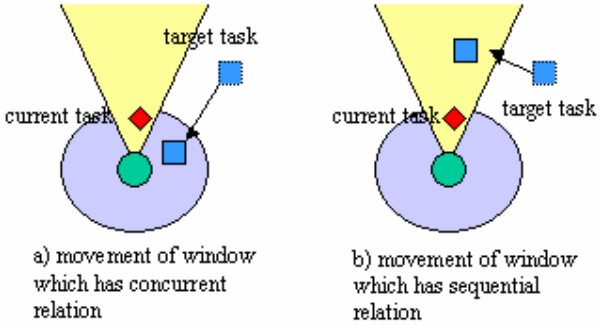


Figure 7: Movement of windows

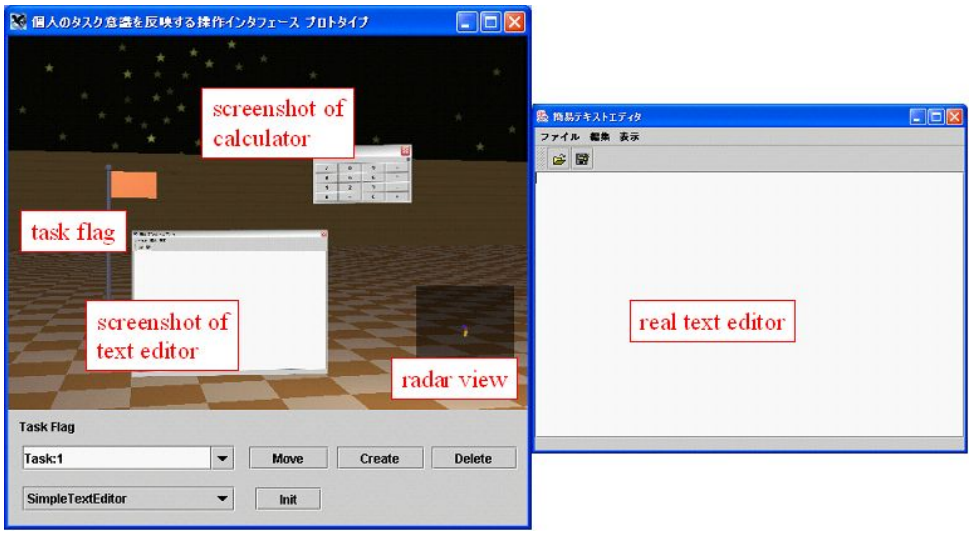


Figure 6: Prototype system

6. Experimental Results

We have asked 8 members in our laboratory to use the prototype system and have evaluated the system. In the experiment, we give members a small task. The task is "to arrange the party". Namely, members are asked to find the nice restaurant through the web, search the way to the restaurant by the railway map, calculate the fare to get to the restaurant, and send the invitation letter by e-mail to the friends. In order for the members to complete the task, we prepared the web browser for searching restaurants, railway map, text editors for creating the e-mail and memos, a list of the friends' e-mail addresses, and a calculator. Firstly, we let members use the interface as a practice, then asked them to work with the task. After the task had been done, the system was evaluated from two viewpoints; whether it could grasp the user's intention correctly, and whether the dynamic movement of windows helps members to use the interface intuitively.

[Experimental result 1]

In order to evaluate the correctness of the relations in the logical window network, such as a sequential relation and a concurrent relation, we have asked members to indicate their tasks, applications that they were used to complete the tasks, and the relations of the tasks for each moment. Then, we compared them to the weights of the links that were calculated by the prototype system.

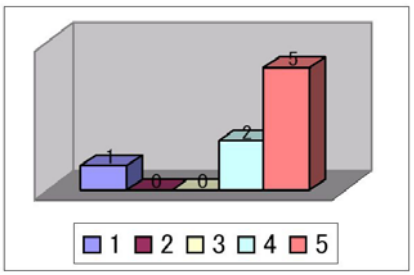
Table 3 shows the result of the experiment. The prototype system could detect the existence of all the relations that the members thought of. In Table 3, each row corresponds to the relations that the prototype system detected and each column represents the relations that the members thought. According to the result, the prototype system could also detect the existence of the relations with more than 90 % accuracy (i.e. (4+1+8+7) /22). However, it could discriminate the types of the relations only with 50 % accuracy (i.e. (4+7) /22). The cause that the prototype system could not grasp the concurrent relation correctly was that the interval time for executing each task was longer than what we had expected. Therefore, if the appropriate threshold T_{th} is set, the prototype system is able to determine the types of each relation correctly.

Table 3: Experimental result 1

		Members		
		Con-current relation	Se-quential relation	no relation
Proto-type system	Concurrent relation	4	1	0
	Sequential relation	8	7	2

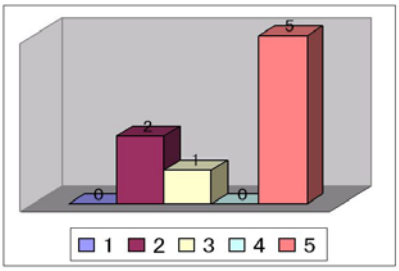
[Experimental result 2]

After the experiment, we asked members about the dynamic movements of windows. The questions are; 1) Did the windows moved to the good position by the system?, and 2) Did the automatic movement of the windows interrupt your current task?. Members are asked to point out the appropriate levels from 1 to 5 for each question. Figure 8 shows the result of the questions. Based on the results of both questions, members thought the movements of the windows by the prototype system were effective and acceptable. However, because currently windows are moved without considering the location of other windows, the sudden emergence of the moved windows sometimes confuses the members. Therefore, to consider the mechanism of arranging windows to appropriate locations is our next issue.



1:inappropriate --- 5: appropriate

(a) Result for question 1



1:interrupted --- 5: not interrupted

(b) Result for question 2

Figure 8: Experimental result 2

7. Conclusion

In this paper, we have proposed the operational interface in which the windows' locations are changed dynamically and automatically according to student's intention and preference. In order to represent the student's intention and preference to the interface, logical window network is generated. Logical window network represents the student's intention to the windows and

tasks to achieve the goal based on the nodes, which correspond to the windows/tasks, and the links, which represents the relations among nodes. The links are generated according to the relations among tasks that are grasped based on the student's manipulation of windows. We have also proposed the methods to detect the relations of windows/tasks based on the intervals of active windows. When the student changes the active window, the system detects the related windows based on the logical window network and reallocates the windows automatically, in order for the student to access the interesting information easily.

In this paper, we also show the effectiveness of proposed interface based on the experimental results using the prototype system. However, our current system could not discriminate the concurrent relation and sequential relation correctly. In order to provide appropriate windows at effective timing, we need to construct the mechanism for finding appropriate threshold of discriminating concurrent relation and sequential relation.

In addition, we should revise our system so as to grasp a student's intention to the individual windows. Moreover, to balance the windows in the virtual learning space needs to be considered when the system moves the related windows dynamically.

References

[1] H. H. Adelsberger, B. Collis, and J. M. Pawlowski (eds.): "Handbook on Information Technologies for Education and Training", Springer-Verlag (2002).

[2] J. E. Bardram and O. W. Bertelsen: "Supporting the Development of Transparent Interaction", Proc. of EWHCI'95, Vol. 1015, pp.79-90 (1995).

[3] S. K. Card, G. G. Robertson, and W. York: "The WebBook and the WebForager: An Information Workspace for the World-Wide Web", Proc. of CHI'96, pp.111-117 (1996).

[4] G. Robertson, M. Dantzich, D. Robbins, M. Czerwinski, K. Hinckley, K. Risden, D. Thiel, and V. Gorokhovsky: "The Task Gallery: A 3D Window Manager", Proc. of CHI 2000, pp.494-501 (2000).

[5] J. M. Carroll, D. C. Neale, P. L. Isenhour, M. B. Rossen, and D. S. McCrickard: "Notification and awareness: synchronizing task-oriented collaborative activity", Journal of Human-Computer Studies, Vol.58 pp.605-632 (2003).

[6] Y. Yamada, T. Hirano, and K. Nishimoto: "TangibleChat: A Chat-system that Conveys Conversation-context-awareness by Transmitting Vibration Produced by Key-stroke-act", Journal of IPSJ, Vol.44, No. 5, pp.1392-1403 (2003) [in Japanese].

[7] T. Kojiri, Y. Ogawa, and T. Watanabe: "Agent-oriented Support Environment in Web-based Collaborative Learning", Journal of Universal Computer Science, Vol.7, Issue 3, pp.226-239 (2001).

[8] T. Kojiri, Y. Ito, and T. Watanabe: "User-oriented Interface for Collaborative Learning Environment", Proc. of ICCE 02, Vol.1, pp.213-214 (2002).

Tomoko Kojiri received the B.E., M.E., and Ph.D. degrees from Nagoya University, Nagoya, Japan, in 1998, 2000, and 2003 respectively. From 2003 to 2004, she was a research associate with Graduate School of Information Science, Nagoya University, Japan. Since 2004, she has been a research associate with Information Technology Center, Nagoya University, Japan. Her research interests include computer-supported collaborative learning, intelligent tutoring system, and human computer interface. She is a member of IPSJ, JSAI, and JSiSE.

Yasuki Ito received the B.E., and M.E degrees from Nagoya University, Nagoya, Japan, in 2002 and 2004 respectively. He currently works for Mitsubishi Electric Corporation.

Toyohide Watanabe received the B.S., M.S., and Ph.D. degrees from Kyoto University, Kyoto, Japan, in 1972, 1974, and 1983, respectively. In 1987, he was an associate professor with Department of Information Engineering, Nagoya University. Currently, Japan. He currently is a professor with Graduate School of Information Science, Nagoya University, Japan. His research interests include knowledge/data engineering, computer-supported collaborative learning, parallel and distributed process interaction, document understanding, and drawing interpretation. He is a member of IPSJ, IEICE, JSSST, JSAI, JSiSE, ACM, AAAI, AACE, and IEEE Computer Society.