Contemporary Engineering Sciences, Vol. 8, 2015, no. 7, 303 - 314
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2015.4323

Handwriting Multi-Tablet Application Supporting Ad Hoc Collaborative Work

Taisuke Inoue

Department of Information Science
Shonan Institute of Technology, Fujisawa, Japan

Makoto Oya

Department of Information Science
Shonan Institute of Technology, Fujisawa, Japan

Shigeyoshi Watanabe

Department of Information Science
Shonan Institute of Technology, Fujisawa, Japan

Abstract

The near field wireless communication functionality known as Bluetooth is loaded as standard on tablets like iPad. Using this near field wireless communication functionality enables two tablets to connect together directly, thus supporting communication between people who just happen to meet up unplanned in that location. This study concerns the development of a collaborative handwriting application that uses a collaborative work support format utilizing near field wireless communication on tablets, known as ad hoc collaborative work support that enables the joint creation of a handwritten object.

Keywords: multi-tablet application; ad hoc collaborative work

1. Introduction

With the development in mobile devices in recent years, high-functionality portable terminals, such as tablets, have become commonplace. Using this near field wireless communication functionality, tablets can communicate directly and be used with each other, thus supporting informal, unplanned communication [1] [2] between people who just happened to meet in that location. In this study, we used near field wireless communication on tablets. The collaborative work support format is referred to as ad hoc collaborative work support.

Table 1 is a table comparing the traditional CSCW (Computer Supported Cooperative Work) and ad hoc collaborative work support. From Table 1, we can see that the main features of previous collaborative work support collaborative work [3] [4] [5] [6] are that it is conducted in a planned manner, and as the participants exist in separate locations, communication occurs via the network in an indirect way.

The main features of the ad hoc collaborative work support method, on the other hand, are that as participants gather in a certain place and start collaborative work, the collaborative work occurs in an incidental way. Further, it can be said that conversation between the participants occurs in a direct manner.

Table 1. comparing the previsious CSCW and ad hoc Collaborative work

Compare item		CSCW	Collaboration work for adhoc
Form of Collaboration work	Plan	Intentionally	Unforeseen
	Communicate for a participant	Indirect	Direct
	Transmit for work subject	Transmission delay (messaging)	Transmission realtime (instantly read or write)
Management for a participant and place		System management	Participant Management for myself
Composition system	Participants	PC or Network terminal	Tablet
	Management server	Available	None
	Communication	Network (Internet)	Near field communication (Bluetooth)
System architecture		Centralized control system	Autonomous control system

From these characteristics, we can say that as there are differences between the previous CSCW and ad hoc collaborative work support, the previous format cannot be adapted. With this study, we develop a collaborative handwriting application where a handwritten object can be jointly created by the users gathering in that location.

2. Scenarios and Functionality

While performing this development, we created scenarios based on user actions, in order to clarify how users use tablets and perform collaborative work. There were 5 scenarios as follows.

- Participants bringing tablets voluntarily gather together and perform collaborative work
- The participants engage in face-to-face direct communication

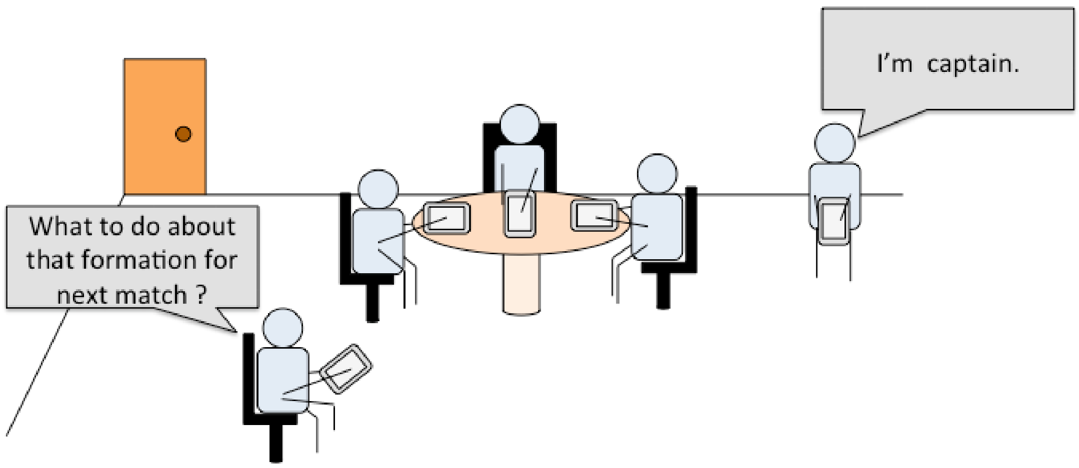

Figure 1. Participation voluntarily gather, engage direct communication

- There is no management server and tablets communicate autonomously with Peer to Peer communication
- The lines drawn on the tablets are reflected immediately on the other terminals

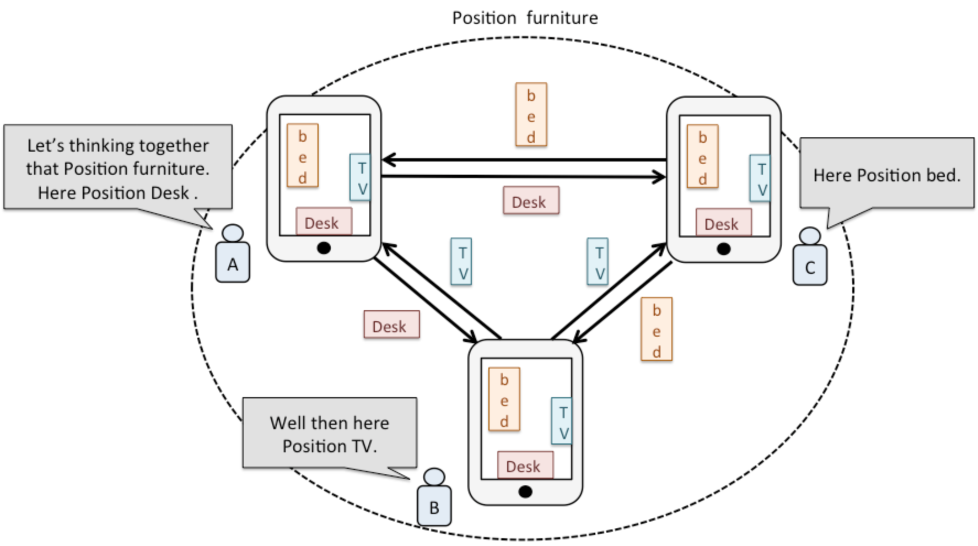

Figure 2. The lines drawn on the tablets are reflected immediately on the other terminals

- A place and participants in that place are managed by the participants themselves

Table 2. List the Function

Function name	explanation
Stroke Object Function	A Stroke Object management.
Stroke Object Creation	Stroke Object Creation, a finger trace on screen.
Stroke Object Drawing	Stroke Object Drawing for screen.
Stroke Object Transmission	Transmission of a Stroke Object for other tablets.
Stroke Object Editing	Delete, Editing(rotation, enlargement etc..) Function.
Place Management	Participation , leave, make automatically , and delete automatically , for place.
Conservation and Restoration Function	Conservation , and Restoration for a joint object.

Table 2 shows lists the functions learned to be necessary as the results of analysing the 5 scenarios. The 5 functions among these that are necessary to perform collaborative work using handwriting are the stroke object function, stroke object creation, stroke object drawing and stroke object transmission. In this paper, we demonstrate the method of realizing these 5 functions.

3. Implementation

a. Implementation of Realizing the Stroke Object Function

A *Stroke object* refers to a single stroke line drawn by a user touching the screen until they release his/her finger. A Stroke object consists of a single color and a thickness. The stroke object function is a function for saving the stroke object created by the user within the memory of the tablet.

Figure 3 is a diagram showing the stroke object function. The stroke object function involves 3 functions. With the memory initialization function, when initialized in tablet memory, the line color and line thickness are saved to memory. The coordinate addition function is a function for adding coordinates that continue the stroke object recorded within memory. With the coordinate access function, in addition to the function for referencing the stroke object coordinates recorded in memory, through the provisions of three functions, it can save the stroke function to memory within the tablet or read from the memory.

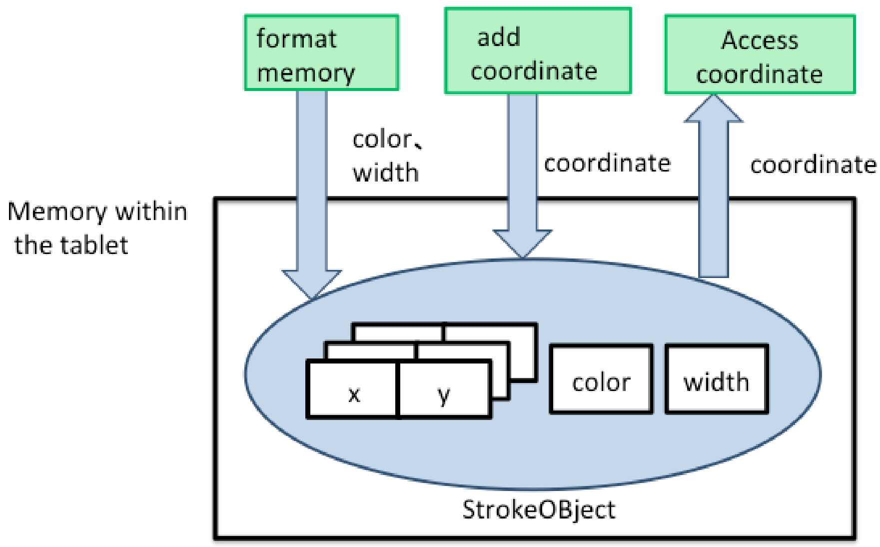

Figure 3. Functions for a stroke object

Multiple stroke objects are recorded in the memory within the tablet. When referencing and adding the coordinates for multiple existing stroke objects, it is necessary to select one stroke object from among the multiple existing stroke objects. For this reason, it is necessary to add an ID with a unique value for each stroke object, known as a *stroke ID*. By specifying the stroke ID, it is possible to add coordinates or reference coordinates for the selected stroke object.

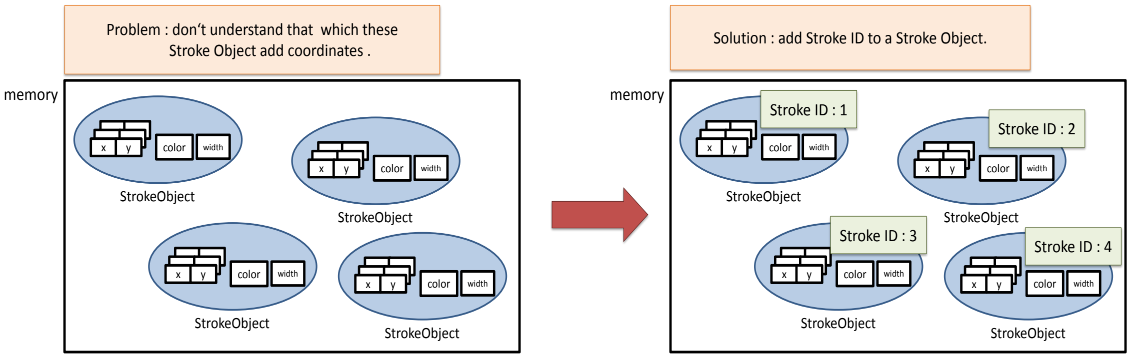

Figure 4. Stroke ID

b. Implementation of the Stroke Object Creation

Stroke Object Creation is a function that adds the coordinates for the stroke object and creating the stroke object in memory, when the user performs an operation with their finger using the stroke object function. The user performs touch operations when they touch the screen with their finger, drag operations when they move their touching finger and release operations when they release their touching finger. When these operations take place, an interrupt is generated within the tablet and the locational coordinates of the finger on the screen can be captured. When this interrupt process occurs, certain functions are used as the stroke object function. In the case of a touch operation, memory initialization is used. Further, when drag or release are performed, the coordinate addition function is used. This enables the stroke object to be created within memory.

c. Implementation of the Stroke Object Drawing

When making successive drawings on the screen, this can be realized using the drawing process.

When making successive drawings, there is the problem of how to draw smooth lines (figure 5). In this study, in order to resolve this problem, the drawing of the line from the tip coordinates to the previous coordinates is stored, and by drawing from the tip to the previous coordinates and the coordinates previous to that, it is possible to draw a smooth line.

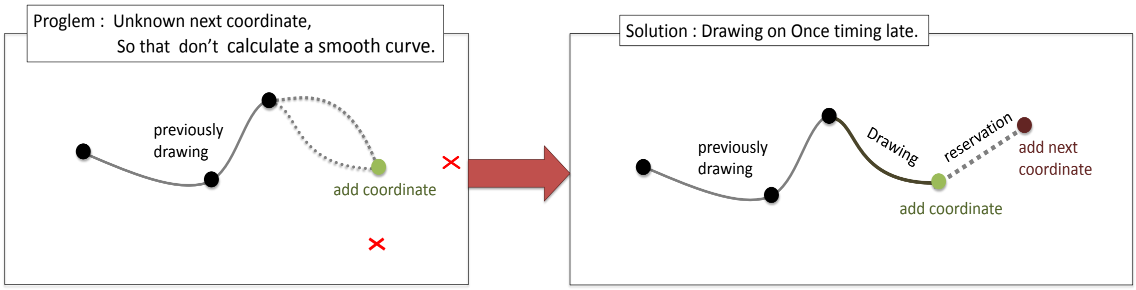

Figure 5. Drawing a smooth curve

Further, calculation of the curved line from the coordinates added for the stroke object uses a B-Spline curve show bellow. The B-Spline curve has the characteristic that part of the whole curve can be obtained and is suited for creating successive drawings in this way. In this study, we use 3D B-Spline curves, with 4 point coordinates to calculate the lines.

$$\mathrm{P(t)} = \sum_{j=-1}^{2} N(t-j) \quad (-1 \leq \mathrm{t} \leq 0)$$

$$\mathrm{N(a)} = \begin{cases} \dfrac{(3|a|^3 - 6|a|^2 + 4)}{6} & (-1 < a < 1) \\ -\dfrac{(|a| - 2)^3}{6} & (-2 < a < 2) \\ 0 & (a \leq -2, a \leq 2) \end{cases}$$

Figure 6 shows the flow for drawing the stroke object. As shown in the figure, when an operation is performed by a user, an interrupt ios is generated and by adding coordinates, the coordinates are added into memory. Next, the successive drawing is executed. Successive drawing takes place through repetition of this flow.

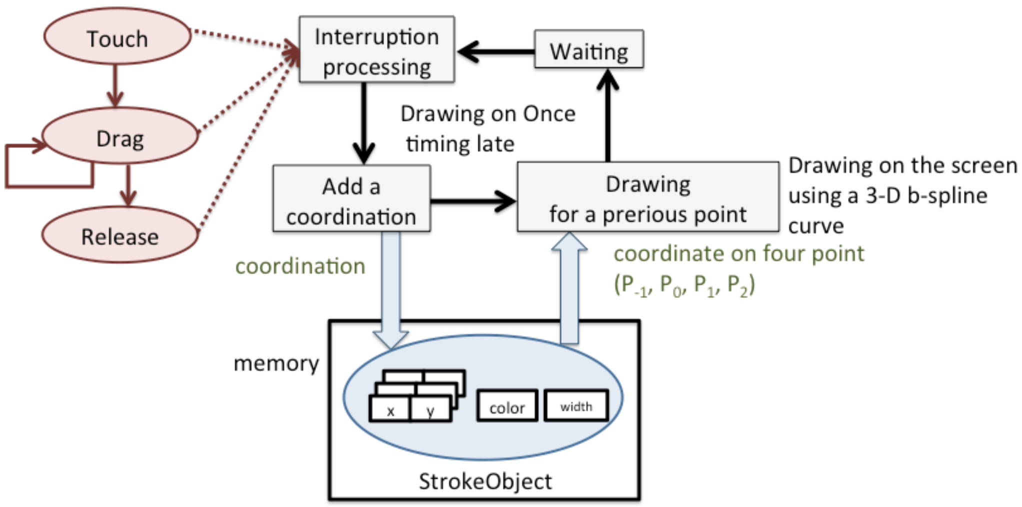

Figure 6. Method of Realizing the Stroke Object Drawing

d. **Stroke Object Transmission**

Figure 7 is a diagram showing the stroke object transmission function. The stroke object is transmitted when an interruption is generated. With this interruption, in addition to the coordinates being added and successive drawing executed, the coordinates are transmitted to the other tablet with which communication is taking place. On the receiving terminal side, the coordinate addition function is used and the received coordinates are added to the stroke object, and drawn on the screen. By processing the stroke object on each terminal in this way, it is possible to draw simultaneously to the screen.

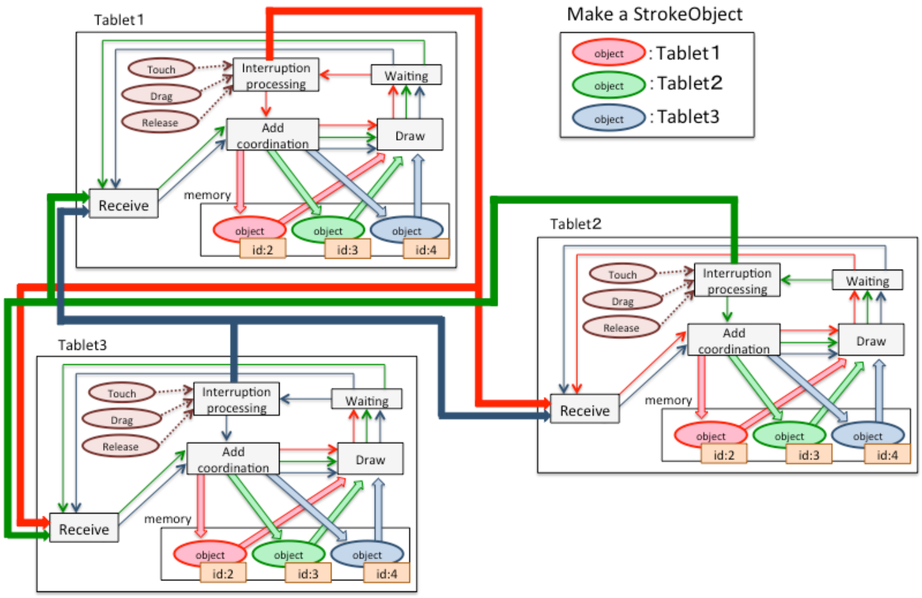

Figure 7. Transmission of Stroke Objects

However, there causes one problem here. This is the problem that when the *stroke ID* attached to the stroke object is different on each terminal, the coordinates may be added to a different stroke object. In order to resolve this problem, in this study the *stroke ID* is set to the same value. We have implemented the *stroke ID* determination function.

e. the Stroke ID Determination Function

Figure 8 shows a diagram of the *stroke ID* determination function. The *stroke ID* determination function is implemented using an algorithm applying a 2-phase commit. In the figure, A is trying to create a stroke object. A first performs ① and ②. Next, B and C, based on the content of ② sent by A, perform ③, and in ④, returns the result to A. Next, where A is sent an OK from all members, it performs ⑥, and if even one NO is returned, ⑥' is performed to determine the *stroke ID*.

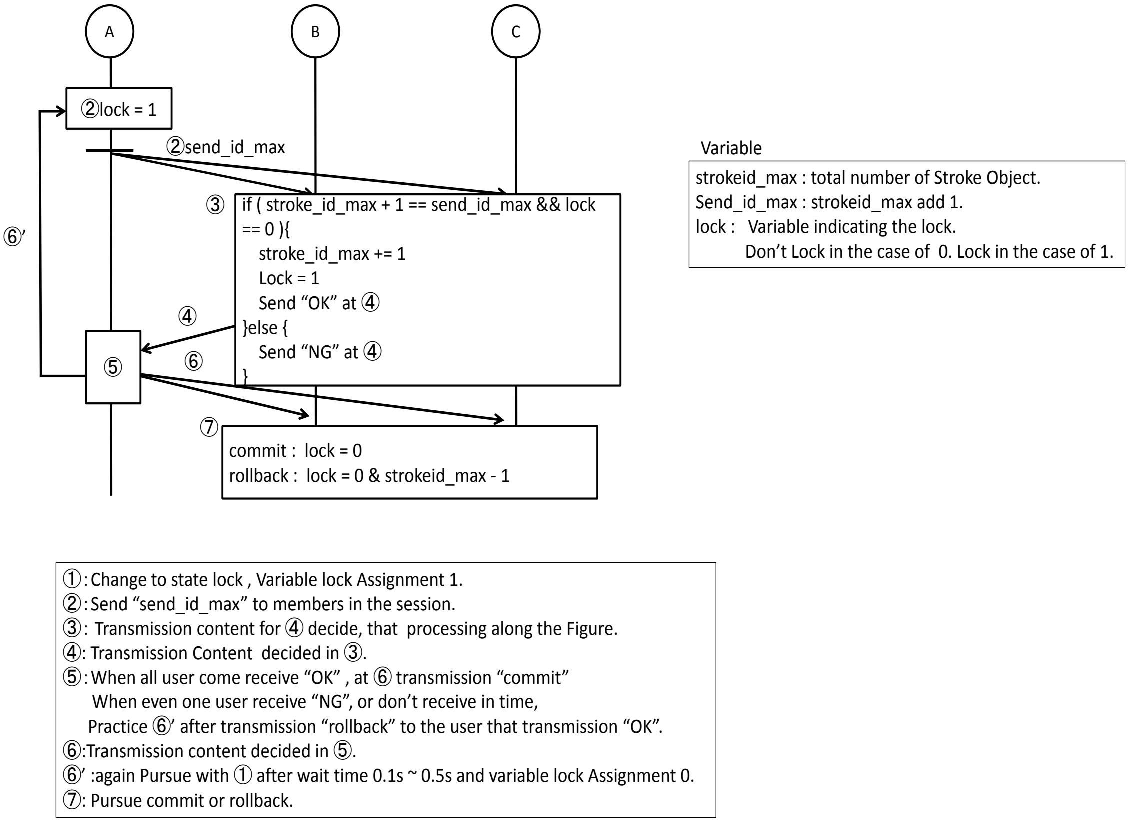

Figure 8. Algorithm for the Stroke ID Determination Function

4. Software Development

In order to verify whether the method of 5. is correct, we developed an application based on this method of realization. The application development was aimed at the iPad 2, Objective-C was used as the development language, and the functions when using for OpenGL for drawing to the screen is implemented using the GKSession class within iOS. The stroke object function is realized by developing a standard handwriting class that becomes the standard handwriting class on the tablet.

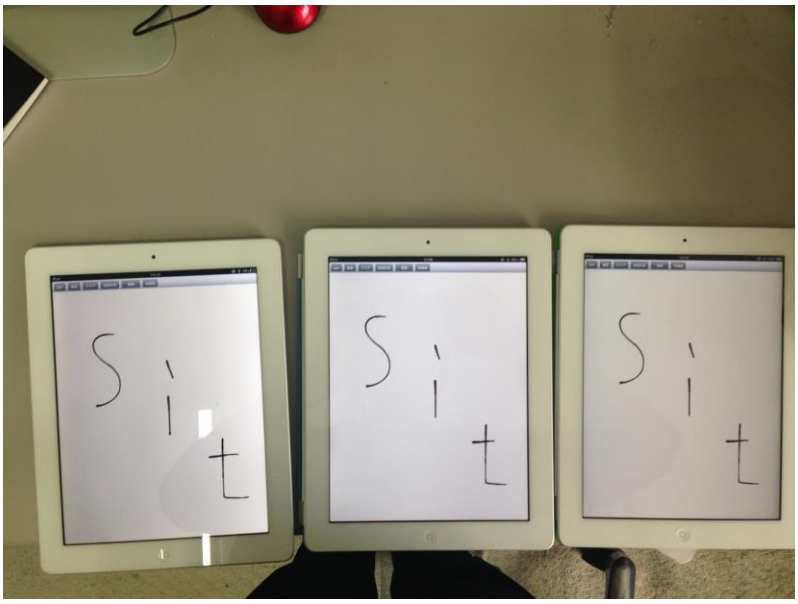

Figure 9. Development result for iPad

5. Summary

In this study, we propose a collaborative handwriting application as an example of ad hoc collaborative work support, and by extracting functions using these scenarios, we have developed an application based on the implementation format, and verified that the implementation format is appropriate. This kind of application is, as far as we know, new and not currently seen elsewhere.

References

[1] R. E .Kraut, R. S. Fish, R. W.Root, and B. L.C.halfonte: Informal communication in organizations: Form, functions, and technology, In S. Oskamp & S. Spacapan (Eds.), Peoples's Reactions to Technology, pp.145-199, Sage Publications London, 1990.

[2] S. Whittaker, J.Swanson, J.Kucan, and C.sidner: TeleNotes: Managing Lightweight Interactions in the Desktop, ACM Trans,CHI, vol.4, no.2 oo.137-168, 1997. http://dx.doi.org/10.1145/254945.254958

[3] M.Roseman, and S.Greenberg: TeamRooms: Network Places for Collaboration, CSCW"96 Processing of the 1996 ACM conference on Computer supported cooperative work, p325-333, 1996. http://dx.doi.org/10.1145/240080.240319

[4] S.A. Bly, S.R. Harrison and S. Irwin: Media Spaces: bringing people together in a video, audio, and computing environment, Communication of the ACM, vol.36, No.1, pp.28-47, January 1993.
http://dx.doi.org/10.1145/151233.151235

[5] M. Stefik, G. Foster, D. G. Bobrow, K. Kahn, S. Lanning, and L. Suchman: Beyond the chalkboard: computer support for collaboration and problem solving in meetings, Communication of the ACM, vol.30, no.1, pp.32-47, 1987.
http://dx.doi.org/10.1145/7885.7887

[6] Hiroshi Ishii: CSCW and Groupware, ohmsha, ISBN4-274-60787-X, 1995(in Japanese).

[7] Hiroki Asai, Hyato Yamane: Extraction of Emphazied Words from on-Line handwritten NoteBooks, DBSJ Journal, vol.10, No.1, pp.67-72, 2011(in Japanese).

[8] Taisuke Inoue, Makoto Oya: A System supporting ad-hoc collaborative works using multi-tablet, the 76th National Convention of IPSJ, 3ZB-2, 2014(in Japanese).

Received: March 15, 2014; Published: March 12, 2015