Methodologies and Mechanism Design in Group Awareness Support for Internet-Based Real-Time Distributed Collaboration

Minh Hong Tran, Gitesh K. Raikundalia, and Yun Yang

Centre for Internet Computing and E-Commerce (CICEC)
Swinburne University of Technology
PO Box 218, Hawthorn, Australia, 3122
{mtran, graikundalia, yyang}@it.swin.edu.au

Abstract. The first purpose of this paper is to provide an overview of the most commonly-used awareness mechanisms, namely, What You See Is What I See, telepointers, multi-user scrollbars, radar views and distortion-oriented views. These mechanisms were derived from researchers' intuition, without prior experimental investigation of what awareness information end-users really need. This research utilised a completely user-centered approach to determine relevant awareness mechanisms. The novelty of this approach is in the use of usability experiments to identify awareness mechanisms. In addition to the illustration of several innovative mechanisms, this research has also successfully differentiated the importance of different awareness information in maintaining group awareness. The significance of different awareness information has been thoroughly compared. These results help designers to know which information must be provided to all team members.

1 Introduction

Real-time, distributed, collaborative writing systems facilitate the task of joint authorship in a distributed environment. Many systems have been produced such as GROVE [6], REDUCE [23], SASSE [1] and ShrEdit [5], but only a small number amongst them are used in the real world. The reason for this lack of usage is that existing systems have not yet matched the diversity and richness of group interaction. Knowing the activities of other collaborators is a basic requirement for group interaction. In face-to-face interaction, people find it naturally easy to maintain a sense of awareness about who is in a workspace, what others' responsibilities are, what others are doing and so on. However, when collaborators are geographically distributed, supporting spontaneous interaction is much more difficult due to various reasons such as limited capabilities of input and output devices, restricted views or weak communication [13].

To support synchronous distributed collaborative writing effectively, systems must provide *group awareness* (GA). GA is defined as "*an understanding of the activities of others, which provides a context for your own activity*" [5]. GA provides information

X. Zhou, Y. Zhang, and M.E. Orlowska (Eds.): APWeb 2003, LNCS 2642, pp. 357–369, 2003.

such as users' identities, their past actions, current activities and future intentions. GA plays an essential and integral role in collaboration by simplifying communication, supporting coordination [6], managing coupling, assisting "anticipation" [13] and supporting "convention" [10]. Various awareness mechanisms have been produced to support GA. Examples of commonly used mechanisms include telepointers [9], radar views [14], multi-user scrollbars [1, 14], and distortion-oriented views [7, 8, 22]. The systems can also incorporate audio and video facilities for supporting communication [2, 4]. Although these mechanisms have enriched GA to some extent, many aspects of GA have not yet been supported. Some awareness mechanisms are implemented in an ad-hoc manner and some may even generate contradictory results. For instance, multi-user scrollbars were reported ineffective in a collaborative page layout task [14]. However, multi-user scrollbars were found useful in collaborative editing as reported in [1].

The remainder of this paper is organised as follows. Section 2 reviews five awareness mechanisms: *What You See Is What I See*, *telepointers*, *multi-users scrollbars*, *radar views* and *distortion-oriented views*. Section 3 introduces the authors' user-centered methodology and the experiment used to identify awareness mechanisms. Section 4 describes several novel mechanisms followed by further discussions in Section 5. The paper is concluded in Section 6 with directions for the authors' future research and development.

2 Overview of Existing Awareness Mechanisms

What You See Is What I See (WYSIWIS)

Conventionally, maintaining GA in face-to-face interaction relies heavily on the short distance between people, i.e., physical proximity [12]. To support GA, early groupware systems provide close view proximity by enforcing strict-WYSIWIS views, where all users see the same set of cursors and the same view of a document. Strict-WYSIWIS applications allow only one user to use mouse or keyboard at a time [17, 18]. Strict-WYSIWIS is relatively easy to implement from a designer's point of view and it helps to maintain GA in the sense that users can identify easily what others are doing as they see the same view.

However, enforcing all users to see the same view makes strict-WYSIWIS an inflexible style of collaboration in which the users are unexpectedly forced to work as a tightly-coupled unit [17]. This design ensures that users can stay aware of others' activities, but it is often too restrictive for many kinds of collaboration where user regularly move back and forth between individual and shared work [16]. Another drawback of strict-WYSIWIS is disruption since all users' movements and actions are entirely visible to all of the users [17].

Due to the inflexibility of strict-WYSIWIS, recent synchronous collaborative editors have relaxed the strict application of WYSIWIS to accommodate natural interactions. As a result, a looser variation of WYSIWIS has been introduced, namely *relaxed-WYSIWIS* view sharing. The idea of relaxed-WYSIWIS view sharing systems is that users are allowed to move and change their viewports independently. However, when users view different parts of the document, they are effectively blinded to the actions happening outside their viewports. To resolve this problem, many awareness mechanisms have been developed specifically for relaxed-WYSIWIS view sharing such as telepointers, multi-user scrollbars, radar views and distortion-oriented views as discussed next.

Telepointers

Telepointers allow a user to see both his/her own mouse cursor and others' mouse cursors. To distinguish telepointers of different users, each telepointer can be assigned a different colour or a different shape and/or even have the user's name and image attached to it [9, 15].

The major advantage of telepointers is providing awareness about users' presence and their activities, foci of attention and degree of interest. By watching telepointers move, a local user knows that the remote user is currently working in the workspace. In addition, telepointers allow users to know that an activity is occurring, and often the kind of actions as well, if names of users' actions are attached to the telepointers.

Although telepointers are able to provide the mouse positions of other users, telepointers can only be used when all participants see the same information on the screen. Telepointers fail to convey awareness information when they are removed from the local user's view. For example, if two users are working on two completely different areas of a document, telepointers become inaccessible. Another problem in implementing telepointers is disruption. Since telepointers show all of the other users' movements, telepointers can easily distract a local user. Another limitation of the telepointer is that it fails to provide users with information about the location at which a remote user is working. In the case of collaborative drawing, telepointers are able to show working locations of other users, because the position of a mouse cursor is a position at which a user is drawing. However, in the case of document editing, the position of a mouse cursor is not a position at which a user is typing. Therefore, the position of a telepointer does not indicate exactly the location at which a remote user is working in a document. In addition to the issues mentioned above, there are other unsolved issues in implementing telepointers such as:

- *Cursor size*: What is the most appropriate cursor size when there are two, four or more users in the workspace? Should cursor size be unchanged or reduced proportionally as group size grows?
- *Cursor lifetime,* especially for inactive cursors: Whilst displaying cursors at all times is useful to indicate who is present and what are they doing, presence of an inactive cursor is intrusive to larger groups [15].
- *Cursor position*: Since users are allowed to scroll and adjust screen size independently, whether cursors should be positioned relatively to the underlying objects in the workspace or absolutely with regard to the window [9].

Multi-user Scrollbars

Multi-user scrollbars show both a local user's position and remote users' positions in a document. The viewport of each user is represented in multi-user scrollbars as a coloured bar locating at the right-hand side of the window [1, 14]. There are two variations of multi-user scrollbars. The major difference between these two variations is that in version 1, each remote scrollbar is located in a different vertical region; however, in version 2, all remote scrollbars are located in the same vertical region. These two variations of multi-user scrollbars offer the same advantages. First, they provide information about other users' locations in the document such as whether users are near the top, the middle or the end of the document. When users have completely different views, telepointers are of no value since they are not visible, but multi-users scrollbars are still capable of conveying information about other users' whereabouts in the workspace. Second, multi-user scrollbars deliver information about relative positions amongst users. Another benefit of the multi-user scrollbar is that this scrollbar makes navigating to another user's location simple by moving the local scrollbar to the same level as the remote user's scrollbar.

The first disadvantage of multi-user scrollbars is that it is difficult to determine the exact position in the document where other users are located, what they can see and what they are doing. This is because all the scrollbar shows is the relative position of the user in the document. The second disadvantage of multi-user scrollbars occurs when a large number of users are working in the workspace. In version 1, the display of a large number of remote users' scrollbars causes space constraints – it forces the area of the document portion viewed to be reduced. In the case of version 2, when views of more than two users intersect, it is difficult to know exactly the location of remote users because many remote scrollbars overlap one another. The third disadvantage is disruption in using multi-user scrollbars. When remote users perform a substantial amount of vertical movement in the document, the remote scrollbars move extensively. Since the scrollbars are placed next to the view of the document, constant movement of remote users' scrollbars become very distracting to a local user.

Radar Views

Radar views render the entire shared workspace within a miniature overview, on which each user's location of activity is superimposed. Radar views present the locations of other users as rectangles. To indicate which rectangle belongs to whom, each rectangle can have a different text label and/or a picture of the user [14]. Telepointers can be added in miniature form to show others' mouse positions and movements. A major advantage of radar views is to provide a high-level view of the shared workspace. A radar view can be seen as an integration that combines advantages offered by telepointers and multi-user scrollbars within a single mechanism. In collaborative writing, radar views provide information about the general structure of the document, locations in the document at which other users are working and their activities upon the document when users work in the same view or different views.

However, radar views have four major problems need to be overcome. First, the major problem with a miniaturisation technique is that it has limited scalability. The low-resolution representation of radar views conceals details of users' actions; a radar view of an extremely large data space contains too little detail to be useful. Second, radar views create a "physical and contextual virtual gap" [7] between local details and global contexts. Third, if users perform a substantial amount of movement, the rectangles in the miniature view change constantly, causing a user's screen to appear very busy. Finally, a problem similar to that of telepointers – radar views fail to distinguish between viewing areas and working areas. Users' viewports drawn in a radar view could be either their viewing or working regions. As a result, if a solely radar view is implemented in an editor, information about other users' working locations provided by radar views could be either insufficient or incorrect.

Distortion-oriented Views

To overcome radar view's limitations, especially to bridge the gap between local details and the global structure of a document, a distortion-oriented display is used. Distortion-oriented views include two categories: *magnifying lenses* (irrelevant to GA) and *fisheye views*. Fisheye views present a single view which displays both local detail and global context on a continuous "surface" [7].

Fisheye views offer two central advantages, which are not provided by any mechanisms described above. First, a fisheye representation bridges the gap between local detail and the global context by providing a seamless and smooth transition between these two views. Second, the usage of multiple focal points – assigning one focal point to each user – allows fisheye views to reveal the location of other users and the details of their actions performed upon the workspace [8]. Extensive technical detail of how fisheye views function is described clearly in [8, 11, 22].

Although fisheye views can provide improved GA of where other users are and what they are doing in a shared workspace, some awareness elements are still poorly supported and some technical issues need to be solved. First, remote users' focal points can be out of a local user's view, when a shared document is too large to fit in the local user's view. The local user apparently loses track of the remote users' whereabouts and what they are doing. Second, since magnified regions representing other users' working areas are adjusted by a local user, these areas do not exactly match with actual viewport size. Third, when more than two enlarged areas overlap they hide one another. This is problematic, as part of document appears to be lost and a local user could be misled when assuming that a hidden remote user has left the workspace.

3 Integrated Approach for Determining Awareness Mechanisms

GA research has exploited a solely "bottom-up" approach in finding awareness mechanisms, i.e., testing mechanisms based on researcher's intuition without prior experimental study of what awareness information end-users really need. Although the solely "bottom-up" approach has exposed some mechanisms, many other possible mechanisms are easily overlooked or require excessive experimentation to be found. To design usable mechanisms, a designer must be directed by the philosophy of user-centered design. This research adopts a completely user-centered approach – a "top-down" approach merged with a "bottom-up" approach [19]. However, this paper only focuses on the work of the "top-down" approach in identifying awareness mechanisms.

- *Top-down approach*

This research started with a "top-down" approach before the "bottom-up" approach. The top-down approach determines user needs and identifies relevant awareness mechanisms by conducting a laboratory-based usability experiment – a top-down experiment. The top-down experiment was conducted with REDUCE – Real-time Distributed Unconstrained Cooperative Editor [23]. REDUCE was selected as the editor for experiments because it has been recommended by the prestigious ACM SIGGROUP (Special Interest Group on Supporting Group Work) as a demonstration editor for trial worldwide. At present, REDUCE provides almost no GA support; hence, conducting the experiment with REDUCE determines valuable awareness information and highly applicable mechanisms. This section describes briefly the setting of the top-down experiment. The full details of the setting were described exhaustively in [19].

- *Bottom-up approach*

The functionality awareness mechanisms derived from the "top-down" approach must be evaluated. After the mechanisms are implemented in REDUCE, the "bottom-up" approach will then determine further awareness support by conducting another series of usability experiments.

Top-Down Experiment

The usability experiment involved *ten pairs* of subjects working on three writing tasks, including *creative writing* (CW) (e.g., writing short essays), *technical document preparation* (TD) (e.g., writing research papers) and *brainstorming* (BS) (e.g., generating ideas). Four pairs worked on CW, three pairs worked on TD and three pairs worked on BS.

Subjects included lecturers and Ph.D. students in Information Technology. They were selected as experimental subjects because their experience in cooperative work and their well-established computer knowledge would give valuable feedback. Identifying relevant awareness mechanisms is the major objective of this research, so choosing technologically-trained experimental subjects is key to the experiment. Each pair participated in a two-and-a-half hour session including:

- *Training* (30 minutes): Each subject was fully trained in using REDUCE.
- *Experiment* (1 hour): Subjects worked on two tasks – one task with verbal communication for thirty minutes and on another task without verbal communication for thirty minutes. Conducting the experiments with and without support of verbal communication allowed identification of problems users had in collaborating and the workarounds users resorted to when silence occurred.
- *Questionnaire and interview* (1 hour): A questionnaire includes twenty five-point scale questions and seventeen open-ended questions. Subjects completed the questionnaire within an interview to discuss awareness information and awareness mechanisms.

4 Awareness Mechanisms from the Top-Down Experiment

Modification Director Mechanism

Assisting users in monitoring who edits their work and which parts of their work are being modified is the inspiration for the emergence of *Modification Director* (MD). The MD maintains awareness by notifying users instantly when their work is modified by other users. Whenever a user's work is modified, a corresponding coloured icon flashes on the local user's screen. The user can easily view the modified area simply by clicking on the flashing icon [21]. MD notifies users immediately whenever their work is modified by others and allows users to find out quickly which part of their work is altered and who makes the modification. Although a pop-up window is used in MD, the window is controlled by users; the window only pops up when a user clicks a flashing icon. Hence, disruption is significantly minimised, because a user is aware of the presence of the pop-up window.

A design issue needs to be addressed when implementing MD; if user A modifies more than two different sections of user B's work at the same time, which section should be shown in the pop-up window? One solution to resolve this problem is to organise the pop-up window in page format, which allows users to go backward and forward amongst the modified areas.

Toggle Multi-user Scrollbar Mechanism

The *Toggle Multi-user Scrollbar* (TMS) is an advanced variation of traditional multi-user scrollbars introduced in [1, 14]. TMS provides all functionality available in traditional multi-user scrollbar plus the ability to hide and show remote scrollbars [20]. Additionally, TMS provides the ability to hide remote user scrollbars. TMS also displays a remote user's name next to their scrollbar, which prevents a user from memorising and mapping many pairs of users and corresponding scrollbars.

One problem in designing TMS occurs when viewports of more than two users intersect, which causes remote scrollbars to overlap. A solution to overcome this problem is to make remote scrollbars transparent; transparency allows all remote scrollbars to be visible when overlapping one another. Another problem with TMS is when the remote scrollbar is set to visible; the screen could be very busy due to a substantial amount of vertical movement. One resolution to this problem is to allow a local user to choose which particular remote scrollbar is visible.

Split Window View Mechanism

In collaborative writing, authors' working areas can be different from their viewing areas [3]. Existing mechanisms such as telepointers, multi-user scrollbars and radar views show users' current viewing areas, but not their working areas. The *Split Window View* (SWV) mechanism enables a user to view both other users' viewing *and* working sections. In SWV, remote insertion cursors are also added to the display. Remote insertion cursors assist a local user in tracking down easily and precisely other users' working areas [21]. These cursors have different functionality to telepointers, thus do not suffer from the disadvantages of telepointers. SWV allows a user to observe more than one users' work at the same time and a user is able to request explicitly whether or not the user wants to retain other users' views. Thus the presentation of awareness information is controlled by users.

Although SWV allows a user to view other collaborators' work simultaneously, this mechanism raises two problems in display and design issues: *space constraints* and *display fidelity*. Since the visible size of a screen is limited, a problem occurs when having multiple views in a single window screen at the same time. The more users that are viewed, the smaller the size of the main editor. This limits the local user's main view of the workspace. Additionally, when both viewing and working areas of a particular user are displayed, SWV might need to implement a low-fidelity presentation of the document in order to fit the entire view in a window of half normal size. Depending on the fidelity, the contents in miniature views can be difficult to read and to understand. Besides that, with low-fidelity views users might have difficulty seeing the remote insertion cursors. If the fidelity is not too low, horizontal scrollbars can be added to each view.

Dynamic Task List Mechanism

The *Dynamic Task List* (DTL) is a task-based awareness mechanism that presents an active and frequently-updated list of all collaborators' tasks [20]. Users' names, their corresponding text colours and their tasks are shown in the list. The display of this list is immediately updated whenever there are changes in collaborators' tasks such as when a new task is assigned, a task is modified or a task is removed from the list. The DTL provides an active awareness information presentation: users can select a particular task to view the corresponding section of the document.

DTL provides high-level awareness information about members' responsibilities, i.e., the tasks for which each member is responsible. DTL also presents a relative comparison and correlation of workloads of all collaborators. For example, when more than one collaborator is responsible for one common task, DTL provides all collaborators with sufficient information about other collaborators with whom they need to cooperate closely. None of the mechanisms covered in Section 2 is capable of supporting this type of awareness information.

Although DTL conveys information about what comments others make on any collaborator's work, the nature of information delivery (a pop-up window) could be intrusive. Especially with those who do not want to know other people's comments while concentrating seriously on their own work. Additionally, though DTL is viewed as possibly a useful and valuable addition to a real-time editor, collaborators need to provide extra effort, apart from writing goals, to build and to maintain the task list. Besides that, the task list might not be useful for a small document, as it is neither convenient nor effective to form the list of tasks for a small-sized document.

5 Further Discussions on Awareness

The preceding sections have reviewed existing awareness mechanisms derived from a solely bottom-up approach and described innovative mechanisms determined by the top-down approach used by the authors. Further issues to be addressed are covered here.

- The top-down approach reveals that many important aspects of group awareness have been mostly ignored by previous research.

Using the top-down approach is better than using a solely bottom-up approach in the sense that the top-down approach ensures that identified mechanisms are definitely based on real data, not imagination, about these users, and what users need. Most previous research has mainly focused on inventing mechanisms to provide information about members' location in the workspace, i.e., information about where in a document other users are currently viewing. However, the results of the top-down approach shows that other important information needs to be provided to end-users such as information about other users' responsibilities, their current working areas and their

Awareness elements	**Mean***	**Std Dev**
• Being able to *comment* on what other users have done	4.53	0.51
• Knowing what *actions* other users are *currently* taking	4.50	0.61
• In the case of nonverbal communication, *having a communication tool* that supports communication between users	4.50	0.61
• Knowing parts of a document on which other users are currently *working*	4.50	0.61
• Knowing tasks for which other users are *responsible*	4.35	0.75
• Knowing if other users know what you have been doing	4.25	0.72
• Knowing *who* is in the workspace	4.15	0.81
• Knowing if other users are *satisfied* with what you have done	4.10	0.64
• Knowing parts of a document at which other users are currently *looking*	3.95	0.83
• Knowing to what extent a portion of a document has been *completed*	3.85	0.88
• Having *voice* communication	3.80	1.11
• Knowing what actions other users are going to take in the *future*	3.75	1.07
• Seeing the position of other users' *cursors*	3.70	0.80
• Knowing to what extent you have completed your work compared to the extent others have completed their work	3.50	0.76
• Being able to view the list of *past* actions carried out by a specific user	3.40	1.14
• Knowing how much time has *elapsed* since other users have used REDUCE	3.40	1.23
• Having *video* communication	3.25	0.97
• Knowing *how long* other users have been in the workspace	2.40	1.19
• Knowing where other users are *geographically located*	1.68	0.95

*Rated on a 5-point scale ranging from 1-"not important" to 5-"very important" current actions and so on. As shown in the table, "Knowing if other users are satisfied with what you have done" is even more important than knowing other users' viewports, however none of current mechanisms support this type of group awareness. Awareness mechanisms discovered by the top-down approach have demonstrated that designers must consider such information as fundamental requirements when developing awareness mechanisms. For instance, the emergence of Dynamic Task List indicates that supporting group-structural awareness, which has been largely ignored by previous research, is as important as supporting workspace awareness.

- Comparison of different awareness elements

Each question in the five-point scale questionnaire represents one *awareness element*. Thus the importance of an awareness element was determined by the value of the *mean* of a corresponding question. The higher a mean is, the more important that awareness element is. The table shows a sorted list of the means and associated standard deviations of different awareness elements. Overall, standard deviations are small, meaning there is little spread in the scores of each question. The four most important awareness elements rated by the experimental subjects were "Being able to comment on what other users have done" (4.53), "Knowing what actions other users are currently taking" (4.50), "In the case of nonverbal communication, having a communication tool that supports communication between users" (4.50), and "Knowing parts of a document on which other users are currently working" (4.50).

As shown in the table, "Knowing if other users are satisfied with what you have done" was considered as important information by the subjects. It is more or less as important as "Knowing who is in the workspace". However, none of previous research has developed mechanisms to support such information. Interestingly enough, the relative importance of knowing other users' "*past*", "*current*" and "*future*" actions is also revealed: knowing other users' current actions is the most important, whereas knowing other users' past actions is the least important.

"Knowing where other users are geographically located" was considered the least important by the experimental subjects: the majority of the subjects believed that such information was not all at important. In the interviews, the subjects explained that it was much more important for them to know who other authors are rather than where they are physically located.

- Other suggestions on supporting group awareness

In addition to the discovery of the mechanisms addressed in the previous section, the interview was also used to identify the difficulties, which subjects had to face when writing collaboratively. Several group awareness issues, which are not supported by those mechanisms, have been identified by the interview.

Issue 1: Subjects suggested that a system should allow a user to tell others which part of document that user wants other users to look at.

Issue 2: Several subjects suggested that a system should provide an effective chatting tool, which allows document objects to be embedded into a message before being transferred to the other end. A chat tool should also support private and public chat options.

Issue 3: Almost all subjects suggested that an editor should use *colour* to indicate the status of document objects.

Issue 4: Nearly half of the subjects wanted the system to provide information about "How long other users remain in the shared workspace".

6 Conclusion and Future Work

This paper has thoroughly reviewed the principal awareness mechanisms of What You See Is What I See (WYSIWIS), telepointers, multi-user scrollbars, radar views and distortion-oriented views. *WYSIWIS* is useful in tightly-coupled interaction, but too inflexible to accommodate natural collaboration when users often shift between individual and group work. *Telepointers* provide information about users' presence, but telepointers are inaccessible when users have wholly different viewports. Telepointers could be very intrusive to users in many cases. *Multi-user scrollbars* present relative locations of users, but fail to provide exact users' locations and activities in the workspace. *Radar views* provide a high-level view of the entire workspace, but create a virtual gap between a local view and the global structure of a document. Radar views scale poorly for a large workspace as data presented in radar views becomes too small to be useful. *Distortion-oriented views* display both local details and the global view on one window. However, distortion-oriented views are problematic when remote users' focal points are outside a local user's view or when more than two focal sections overlap.

This research adopts a completely user-centered approach in designing awareness mechanisms; that is, an integrated approach merging a "top-down" approach with a "bottom-up" approach. The "top-down" approach has discovered four innovative mechanisms of *Toggle Multi-user Scrollbar* (TMS), *Split Window View* (SWV), *Modification Director* (MD), and *Dynamic Task List* (DTL). In brief, *TMS* tackles the issue of distraction residing in traditional multi-user scrollbars by giving users the ability to control the visibility of remote scrollbars. *SWV* allows users to gauge easily other users' viewing areas and working areas in the shared document. *MD* notifies a user instantly whenever their work is modified by other users. *DTL* provides high-level information about users' responsibilities, a correlation of workloads of all users. This research provides the successful *differentiation* of the importance of various awareness elements. The results of comparing different awareness elements help designers to have better understanding of which information should be provided to support group awareness. No analysis of the importance of awareness elements has been done by other researchers.

In the future, a "bottom-up" approach will evaluate the effectiveness of these awareness mechanisms by conducting a further set of usability experiments.

Acknowledgement. We sincerely thank Professor Penelope Sanderson for her initial support and valuable insight of this research. We gratefully acknowledge SCHIL for useability experiments support, and Professor Chengzheng Sun for REDUCE support.

References

1. Baecker, R. M., Nastos, D., Posner, I. R., Mawby, K. L.: The User-centred Iterative Design of Collaborative Writing Software, Proc of CHI'93, Amsterdam (1993) 399–405.
2. Beaudouin-Lafon, M., Karsenty, A.: Transparency and Awareness in a Real-Time Groupware System, Proc of UIST'92, CA (1992) 171–180.
3. Benford, S., Bowers, J., Greenhalgh, C., Snowdon, D.: User Embodiment in Collaborative Virtual Environments, Proc of CHI'95, USA, (1995) 242–249.
4. Bly, S. A., Harrisor, S. R., Irwin, S.: Media Spaces: Bringing People Together in a Video, Audio, and Computing Environment, Communications of ACM, vol. 36 (1993) 28–47.
5. Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces, Proc of CSCW'92, Toronto, Canada (1992) 107–114.
6. Ellis, C. A., Gibbs, S. J., Rein, G.: Groupware: Some Issues and Experiences, *Communications of the ACM,* vol. 34 (1991) 39–58.
7. Greenberg, S., Gutwin, C., Cockburn, A.: Awareness through Fisheye Views in relaxed-WYSIWIS Groupware, Proc of Graphics Interface 1996, Toronto, Canada (1996) 28–38.
8. Greenberg, S., Gutwin, C., Cockburn, A.: Using Distortion-oriented Displays to Support Workspace Awareness, Proc of the HCI'96, Imperial College, London (1996) 229–314.
9. Greenberg, S., Gutwin, C., Roseman, M.: Semantic Telepointers for Groupware, Proc of OzCHI'96, Australia (1996) 54–61.
10. Grudin, J.: Groupware and Social Dynamics: Eight Challenges for Developers, Communications of the ACM, vol. 37 (1994) 92–105.
11. Gutwin, C.: Improving Focus Targeting in Interactive Fisheye Views, Proc of CHI'02, Minneapolis, Minnesota, USA (2002) 267–274.
12. Gutwin, C., Greenberg, S.: The Effects of Workspace Awareness Support on the Usability of Real-time Distributed Groupware, ACM Transactions on Computer-Human Interaction, vol. 6 (1999) 243–281.
13. Gutwin, C., Greenberg, S.: A Descriptive Framework of Workspace Awareness for Real-Time Groupware, Computer Supported Cooperative Work, The Journal of Collaborative Computing, vol. 11 (2002).
14. Gutwin, C., Roseman, M., Greenberg, S.: A Usability Study of Awareness Widgets in a Shared Workspace Groupware System, Proc of CSCW'96, Massachusetts (1996) 258–267.
15. Hayne, S., Pendergast, M., Greenberg, S.: Implementing Gesturing with Cursors in Group Support Systems, Journal of Management Information Systems, vol. 10 (1994) 42–61.
16. Salvador, T., Scholtz, J., Larson, J.: The Denver Model for Groupware Design, ACM SIGCHI Bulletin, vol. 28 (1996) 52–58.
17. Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., Tatar, D.: WYSIWIS Revised: Early Experiences with Multiuser Interfaces, ACM Transactions on Office Information Systems, vol. 5 (1987) 147–167.
18. Suthers, D. D.: Architectures for Computer Supported Collaborative Learning, IEEE International Conference on Advanced Learning Technologies (ICALT'01), Madison, Wisconsin (2001) 25–28.

19. Tran, M. H.: A Group Awareness Framework for Real-time Cooperative Editing based on Experimental Study using REDUCE, Research Report of Master of Information Technology, School of Information Technology, Swinburne University of Technology, Australia (2002).
20. Tran, M. H., Raikundalia, G. K., Yang, Y.: Usability Experiments for Determining Document-Related Awareness in Real-time Cooperative Editing, Proc of the Sixth Australasian Document Computing Symposium ADCS'01, Coffs Harbour (2001) 95–98.
21. Tran, M. H., Raikundalia, G. K., Yang, Y.: Split Window View and Modification Director: Innovative Awareness Mechanisms in Real-time Collaborative Writing, Proc of the Conference on Human Factors HF'2002 (CD ISBN: 0-85590-789-5), Melbourne, Australia (2002).
22. Weir, P., Cockburn, A.: Distortion-oriented Workspace Awareness in DOME, British Computer Society Conference on Human-Computer Interaction, Sheffield Hallam University, Sheffield (1998) 239–252.
23. Yang, Y., Sun, C., Zhang, Y., Jia, X.: Real-time Cooperative Editing on the Internet, IEEE Internet Computing, vol. 4 (2000) 18–25.