Adapting the Locales Framework for Heuristic Evaluation of Groupware

Saul Greenberg
Geraldine Fitzpatrick
Carl Gutwin
Simon Kaplan

Cite as:

Greenberg, S., Fitzpatrick, G., Gutwin, C. and Kaplan, S. (1999). **Adapting the Locales Framework for Heuristic Evaluation of Groupware.** Report 99-643-06, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4. www.cpsc.ucalgary.ca/grouplab/papers/

ADAPTING THE LOCALES FRAMEWORK FOR HEURISTIC EVALUATION OF GROUPWARE

Saul Greenberg[1], Geraldine Fitzpatrick[2], Carl Gutwin[3] and Simon Kaplan[1]

[1]Department of Computer Science, University of Calgary, Calgary, Alberta, T2N 1N4, Canada
email: saul@cpsc.ucalgary.ca
http://www.cpsc.ucalgary.ca/~saul

[2]Department of Computer Science and Electrical Engineering / CRC for Distributed Systems Technology, University of Queensland, St. Lucia, QLD 4072, Australia
email: [ger, s.kaplan]@dstc.edu.au
http: www.dstc.edu.au

[3]Department of Computer Science, University of Saskatchewan, Saskatoon, Canada
email: gutwin@cs.usask.ca
http: www.cs.usask.ca/faculty/gutwin

ABSTRACT

Heuristic evaluation is a rapid, cheap and effective way for identifying usability problems in single user systems. However, current heuristics do not provide guidance for discovering problems specific groupware usability. In this paper, we take the Locales Framework and restate it as heuristics appropriate for evaluating groupware. These are: 1) Provide locales; 2) Provide awareness within locales; 3) Allow individual views; 4) Allow people to manage and stay aware of their evolving interactions; and 5) Provide a way to organize and relate locales to one another. To see if these new heuristics are useful in practice, we used them to inspect the interface of Teamwave Workplace, a commercial groupware product. We were successful at identifying the strengths of Teamwave as well as both major and minor interface problems.

KEY WORDS:

Groupware evaluation, heuristic evaluation, inspection methods, locales framework.

1 INTRODUCTION

HCI researchers and practitioners now have a good repertoire of evaluation methodologies that they can use to determine the usability of conventional single user systems. These methodologies range from: those generating quantitative analyses *vs.* qualitative subjective descriptions; those that are precise but expensive to perform *vs.* those that are approximate but cheap to do; and those that require user involvement *vs.* inspection techniques that can be done without user participation. For example, controlled experimentation typically produce objective quantitative results, are precise but expensive, and require observations of users. In contrast, inspection techniques such as heuristic evaluation and task-centered walkthroughs typically produce qualitative and subjective descriptions of usability problems, are inexact but very cheap to do, and require no user involvement. The point is that evaluators can choose an appropriate methodology to fit the needs and nuances of their situation.

In contrast, there is no equivalent set of evaluation methodologies readily available to groupware researchers and practitioners. While some methods have been adapted from other disciplines, these are typically too expensive to use for everyday evaluations, are difficult to learn, or are not appropriate for the questions being asked. As one example, evaluators using ethnographic methods are usually experienced ethnographers, and the studies are almost always expensive in terms of time, logistics and analyses. Similarly, controlled experimentation applied to groupware evaluation are difficult to design and analyze due the variability inherent in group composition and interaction, the lack of proven metrics that are capable of measuring things of interest in groupware, and the resilience of groups in adapting their behaviors to different controlled situations.

The consequence is that groupware is difficult to evaluate. As a result, the CSCW community now includes as part of its research agenda the development of evaluation methodologies for groupware. For example, some researchers are developing a 'discount' version of ethnography called *design ethnography*, where observations and analyses are specific to design implications rather than to a general understanding of a culture. Other researchers are trying to identify metrics specific to groupware.

In this vein, our goal is to see if heuristic evaluation, a cheap but effective evaluation methodology, can be adapted to groupware evaluation. We recognized that existing heuristics, while excellent for evaluating single user systems, do not provide the evaluator with any guidance for discovering problems specific to groupware. Thus our first step in pursuing this goal was to create a new set of heuristics. Fortunately, we felt that an existing theory of group behavior called *the Locales Framework* was a promising place to begin.

In this paper, we present our adaptation of the Locales Framework as heuristics for evaluating groupware. In Section 2, we give a brief summary of both heuristic evaluation and the Locales Framework. In Section 3, we list the groupware heuristics adapted from the Locales Framework. In Section 4, we summarize our experiences using these heuristics to inspect the interface of Teamwave Workplace, a commercial groupware product. We close by outlining the next steps in our research agenda.

2 BACKGROUND

In this section, we briefly review the basic premises of heuristic evaluation and the Locales Framework. Further details are available through the references: both are well documented and well-known to the research community.

2.1 Heuristic Evaluation

Heuristic evaluation (HE) a widely-accepted discount evaluation method for diagnosing potential usability problems in user interfaces (Nielsen 1993, 1994a,b). HE defines a particular interface inspection process (Nielsen and Mack 1994) where several evaluators examine an interface and judge its compliance with recognized usability principles called 'heuristics' (Nielsen, 1994a, p. 26). Non-compliant aspects of the interface are captured as interface bug reports, where evaluators describe the problem, its severity, and perhaps even suggestions of how to fix it.

HE popular is popular with both researchers and industry for a variety of reasons. First, it is low cost: it can usually be done in a few hours, and it does not require any end-user involvement. Second, it is easy to learn: the heuristics are well documented and worked examples are available (e.g., Nielsen 1994a; 1995). Third, it is becoming part of the standard HCI curriculum (e.g., Greenberg 1996) and thus known to many HCI practitioners. Fourth, it can be used fairly effectively by trained non-usability experts (Nielsen 1994a). Finally, it has a good success rate, where the pooled results of 3-5 evaluators typically identify about 75-80% of all usability problems (Nielsen 1995).

What makes HE particularly powerful is that considerable thought has gone into the selection of heuristics. Starting with a large body of literature of interface guidelines, Nielsen (1994b) examined how well particular guidelines explained usability problems held in a database. The result was a set of about ten recommended heuristics, where the set collectively describes the majority of problems commonly found in today's graphical user interfaces. The catch is that these heuristics are very general, such as "Prevent errors" (Nielsen 1994a). This level of generality means that heuristics are motherhood statements that serve only to guide the inspection rather than prescribe it: an evaluator must exercise his or her judgement and interpretation skills to uncover particular problems.

2.2 The Locales Framework

The Locales Framework was developed as a principled approach to help people understand the nature of social activity and work, and how a locale (or place) can support these activities (Fitzpatrick 1998; Fitzpatrick, Mansfield, and Kaplan 1996). In this framework, a social world is a group of people with some common purpose, a site for collaboration, and some means to communicate. A locale is the actual site in which a group collaborates, the actual means by which people communicate, and the actual means by which the work is achieved. For example, a team with the goal of designing a software system would form the social world. If the team met in a meeting room, the room and all its artifacts—its visual and auditory communication, its whiteboard, its table, and the paper within it—would form a physical locale. If instead the team met through a groupware system in conjunction with a telephone, then that would form a virtual locale. While a virtual locale may not be actually realized in physical space, it still provides the site and means for the social world to collaborate.

More formally, the locales framework comprises five aspects, as summarized here but detailed in Fitzpatrick, Mansfield, and Kaplan (1996) and in full in Chapter 8 of Fitzpatrick (1998).

- *Locale foundations* define a collection of people, resources and artifacts (tools, objects, information) in relation to the central purpose of the social world.
- *Mutuality* considers those interactions within locales that maintain a sense of shared place.
- *Individual view over multiple locales* acknowledges that individuals can be participating in many locales.
- *Interaction trajectories* concern how courses of action evolve over time.
- *Civic structures* concern how interactions fit within a broader communal level.

3 DERIVING GROUPWARE HEURISTICS FROM THE LOCALES FRAMEWORK

Nielsen's analysis that resulted in his recommended heuristics were derived from how well they explained usability problems found in single user environments, but not in groupware. To understand why this makes existing heuristics insufficient for groupware evaluation, we first describe the differences between single user usability and groupware usability. We then present new heuristics based on the Locales Framework.

S*ingle-user usability* is the degree to which a system is effective, efficient, and pleasant to use, given a certain set of users and tasks (e.g. Shackel 1990; Neilsen 1992). Usability thus emphasizes 'task work': how a person does the domain tasks and activity that produces things like drawings, documents, or models. Clearly, a groupware system must also allow task work to proceed effectively, efficiently, and pleasantly. However, groupware must go beyond task work and support teamwork—the work of working together—in order to be truly usable. Teamwork involves several activities. For example, group members must communicate, organize joint action, provide assistance, coordinate activity, divide labor, and monitor each other's work. Our conception of *groupware usability* focuses on teamwork, and we define it as the degree to which a groupware system supports the activities of collaboration (Gutwin and Greenberg 1998). Although teamwork also involves social and affective activities, we limit our definition to those that accomplish the mechanics of collaboration: communication, coordination, planning, monitoring, and assistance.

While existing usability heuristics could identify problems in task work, they say nothing about teamwork. Yet developing new heuristics for group work is not straight forward. Nielson had a good corpus of interface development guidelines to work with, but there is no equivalent corpus specific to groupware and teamwork. As an alternative, we decided to use the Locales Framework as the basis for new heuristics. It seemed amenable to this because (and similar to a corpus of guidelines) it is a general rather than specialized framework for understanding fundamental aspects of teamwork, and because it describes a small set of different but inter-dependent perspectives of the characteristics of teamwork, where each could be reconsidered as a heuristic. It is also one of the few theories or frameworks specifically created with groupware in mind. It has also been validated as both a way to understand existing work practices, and to motivate the design of new systems. Thus it is reasonable to expect that it can also be used as a standard against which to evaluate groupware, particularly those that claim to support long-term team activity within some kind of 'space' or 'place'.

The Locales Framework was not developed as a usability evaluation method. Rather, it was developed to understand the nature of social practices in the workaday world, and to help inform the design of groupware. Still, converting the framework into heuristics is straightforward. Using the text and descriptions found in Chapters 1 and 8 in Fitzpatrick (1998), we recast them as statements that asks whether a groupware's interface allow certain social phenomenon to occur. As with Neilson's heuristics, these statements are brief reminders to help guide evaluators. Detailed explanation describing the nuances behind each statement are found in Fitzpatrick (1998).

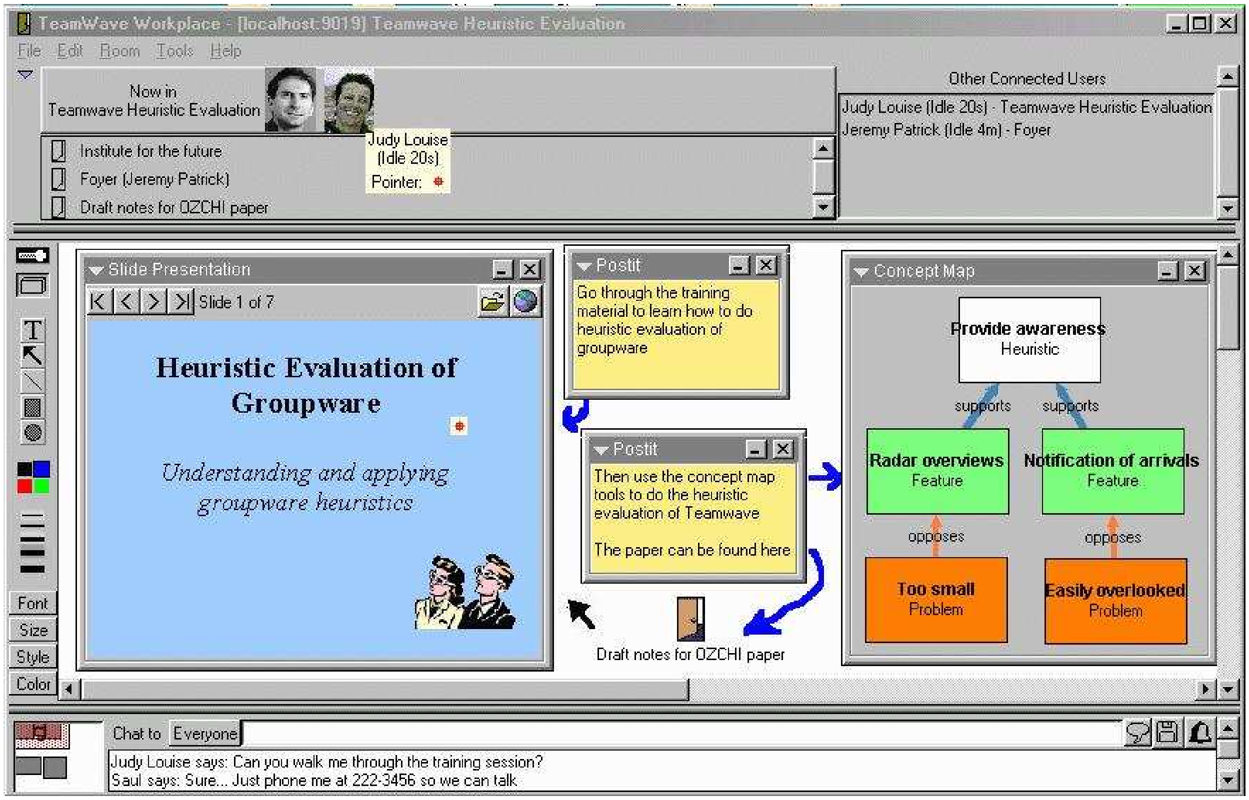

Figure 1. A Snapshot of a Teamwave Workplace Room

Provide centers (locales) that collect people, artifacts and resources in relation to the central purpose of the social world. A locale provides the site, means and resources for a group to pursue team and task work. Locales should be dynamic so they can evolve along with the people, the artifacts, and the purposes that defines them (Harrison and Dourish 1996).

Provide awareness (mutuality) within locales that helps people maintain a sense of shared place and that keeps them informed about shared activity. Mutuality includes one person's awareness of others, the artifacts comprising the locale, where things are located, and how things are changing (Gutwin and Greenberg 1999).

Allow individual views so one can view a locale or to aggregate multiple locales as it relates to one's responsibilities, activities, and interests. A particular person should be able to view locales from his or her particular perspective and in a way that reflects their degree of focus and participation.

Allow people to manage and stay aware of their evolving interactions over time. This includes a group's control over past, present and future aspects of routine and non-routine work; how people coordinate and negotiate plans and activities over time; how people leverage past experiences; how breakdowns are noticed and repaired; and how processes are supported.

Provide a way to organize and relate locales to one another (civic structures*).* Locales are rarely independent of one another: people need a way to structure the locales in a meaningful way, to find their way between locales, to create new locales, and to remove old ones.

The Locales Framework would not claim that all groupware applications should support all heuristics. Rather, the heuristics point to aspects that might be useful in a total collaborative work context, depending on the specific shared task needs. Still, evaluation of groupware independent of known tasks and total use context can suggest areas of system strengths, and also areas of weakness that might need to be compensated for in other ways.

4 CASE STUDY: EVALUATING TEAMWAVE WORKPLACE

We performed a heuristic evaluation on a groupware system to give us a sense of whether the new heuristics are useful in practice We chose as a case study Teamwave Workplace (http://www.teamwave.com), a state-of-the-art commercial Internet groupware product based on a room metaphor. There were several reasons for our choice.

1. TeamWave is designed on a spatial metaphor, while the Locales Framework promotes a place metaphor. Because there are strong similarities (as well as strong differences!) between these metaphors, we felt that the heuristics would be particularly appropriate for evaluating Teamwave.
2. TeamWave is a groupware environment designed to support teamwork; users then craft this environment to fit specific task work needs. This emphasis of teamwork over task work matches the emphasis of the heuristics.
3. Some of the authors were involved in the development of TeamRooms, a research precursor of TeamWave (Roseman and Greenberg 1996). Others had development of the Locales Framework. Although independently produced, we were collectively familiar with the framework and the design rationale behind the software.
4. Two of the authors had previously debated against each other as members of a conference panel: one argued for groupware design based on the Locales Framework, while the other argued for design based around a room metaphor as provided in Teamwave. We wanted to see if the heuristics would help us overcome our biases and come to a common understanding of the strengths and problems found in TeamWave's room metaphor.
5. We believe Teamwave one of the best commercial collaborative environments, and thus appropriate to analyze.

4.1 A brief description of Teamwave Workplace

Teamwave Workplace is a groupware environment designed around a spatial metaphor. Its users can create any number of 'rooms', each being a bounded space. When two or more people enter a room, they are immediately connected together: all people within the room then see the same thing. Within a room, people can write and draw on the back wall, and can add one or more groupware applications into a room. For example, Figure 1 illustrates a room crafted for a team whose purpose is to learn groupware evaluation and to perform a heuristic evaluation. The 'Slide Presentation' application contains a training presentation; several 'Postit' applications along with arrows drawn on the back wall give instructions; a 'Concept Map' tool records the group's evaluation, and a 'Doorway' leads to a related room that contains a draft of this paper. Because rooms are large, people can only see part of its contents: a radar overview (bottom left) provides awareness of the space by showing a miniature of the entire room. People within a room can textually chat with each other (bottom pane). They can also see: who else is in this room (top right pane); a list of other rooms and their occupants (lower top pane). People are also notified of who else is in the community and their comings and goings (top right). Finally, rooms are persistant: when people leave and re-enter a room at a later time, its contents are remain unchanged (unless someone had changed things within a room in the interim). There are many other features of Teamwave: full descriptions as well as trial versions of the software can be obtained from www.teamwave.com.

4.2 Problems identified

Through the heuristic evaluation, we found that Teamwave workplace supports many, but not all of the properties described in the Locales framework. Its major strengths and problems are summarized below.

Provide centers (locales). One of the strengths of Teamwave is that it provides a locale to the group through its familiar room metaphor. As a locale, the room is the site, and the applications added to a room are the means for doing the work. Most real time collaborative resources are almost free within a room: all people have to do is enter a common room. Because new applications can be added and removed at will, it seems reasonable that the room as local can evolve along with the people and purposes that define it (as illustrated in Figure 1). On the down side, Teamwave's only resource for direct communication by people within a room is a chat facility. Because audio and video communication must be done through other means, the room as locale lacks an essential resource to support rich communications. To be fair to Teamwave, we should add that this was a conscious decision on their part: the developers were constrained because they wanted Teamwave to be able to work over modest speed communication links (e.g., modems); they wanted to avoid the many platform dependant issues with today's audio and video standards. They also felt that existing communication channels, such as a telephone, would be an adequate replacement for the chat facility when required.

Provide awareness within locales. Teamwave provides several mechanisms for awareness. People looking at the same portion of a room see all actions as they occur in real time, including each other's telepointer movements. The radar overview provides awareness of the other parts of the room,. As well, people are kept informed via notifications in the top panes of other people's coming and goings. In spite of this, there are problems. The radar overview is too small to give a good sense of another's activity in a different corner of the room. The text-based notifications of people's arrival and departures are easily missed, especially if a user decides to save screen space by resize this information pane to a single line. Finally, telepointer tracking (which provides awareness of where others are and what they are about to do) is difficult because they are too small and their movements somewhat jerky.

Allow individual views. Teamwave allows people within a room to look at different parts of it, and lets an administrator assign different levels of access permissions to particular people within a room. Aside from this, different people cannot have differing views of a room. Thus the Teamwave approach is only reasonable if people consistently share similar responsibilities, activities and interests. Teamwave also provides no good way to aggregate multiple locales. Although users can copy an application to another room, the hard walls of a room essentially bound and separate locales into discrete entities.

People should be able to manage and stay aware of their evolving interactions. Because rooms and their contents persist, each room is a dynamic entity that evolves over time. Thus being aware of the current room state is easy. However, a large and perhaps serious problem is that people have almost no awareness of what others are doing when they do not share the same room: there is no indication of what those people are working on, whether they are active, and so on. In terms of understanding past actions, Teamwave provides a version control system that allows people to list and visit a set of previous 'snapshots' taken of a room and its contents. However, the interface to the version control system is awkward, making it unwieldy to stay aware of prior changes e.g., what changes happened since a person was last in the room, who else had entered the room previously, who had performed particular changes, etc. Finally, Teamwave provides little in the way of process management; processes must be defined and supplied (if at all) by social convention or via individual applications.

Provide a way to organize and relate locales to one another. Except for the linking of rooms by doorways, there is no way to organize and relate rooms together. At the top level, all rooms are presented as a single, flat list. Nor is there a way to provide a different structure for different people.

4.3 Discussion

After applying these heuristics to Teamwave Workplace, we felt that we had identified several design problems. In particular, the 'hard walls' inherent in its room metaphor limited awareness across rooms. It also prohibited a person's ability to aggregate multiple locales to fit a particular perspective. The fixed nature of space within a room also disallowed individual perspectives of a room. As well, the heuristics helped us realize that Teamwave did not have any good ability to structure rooms together. Rooms also tended to present themselves as very much in the here and now; it was awkward to see how things changed over time, and no process could be established to create a structure for how people would manage themselves and their artifacts as the room evolved. Finally, the heuristics helped identify smaller usability bugs, such as the fact that the radar overview is too small to provide good awareness, and that notifications of people's arrival and departures may be overlooked.

Of course, the severity of these problems have not been discussed; if these aspects prove important for the task at hand e.g., rich communication channels, individual views, etc., then Teamwave should be modified, or complementary tools and facilities added, or social workarounds developed.

Finally, we still believe Teamwave to be one of the richest collaborative environments today. While the heuristics identified problems, they also suggest that Teamwave supports the basic idea of locale; the system just lacks a few of the subtle aspects that accompany the idea of locale.

5 SUMMARY AND FUTURE WORK

Our initial development and application of groupware heuristics are promising, as indicated by our case study. Although familiar with Teamwave, the heuristic evaluation indicated potential problems that we would have missed (although their severity are likely task and situation dependant), and hinted at what else might be needed to support rich collaboration.

Of course, much is left to do. In keeping with Nielson's development of the heuristic evaluation method, we (as a community) need to create databases of groupware usability problems and use this database to see how well these particular heuristics, as well as others, serve to identify major problems. We must also examine how well people can use these heuristics in practice; while conventional heuristics are easy to learn and apply, the subtle nuances of teamwork implied by these groupware heuristics may require a greater level of evaluator training and experience. Finally, we have to understand whether these heuristics are applicable to task-based groupware, which comprise the majority of existing groupware systems today (e.g., shared whiteboards, shared editors, brainstorming tools, etc.). Most of these applications would fail miserably when judged by these heuristics. For now, it is likely that the Locales-based heuristics are best used for evaluating design subtleties in general groupware environments (such as Teamwave Workplace) rather than specific groupware applications.

6 ACKNOWLEDGEMENTS

This research was sponsored by the National Institute of Standards and Technology, by Microsoft Research, by the National Science and Engineering Research Council of Canada, and by the Cooperative Research Centre Program through the Department of Industry, Science & Resources of the Commonwealth of Australia. This work has also been supported in part by the United States Defence Advanced Research Projects Agency under grants F30602-96-2-0264 and F30603-94-C-0161 (both administered by the US Air Force through Rome Laboratories). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, expressed or implied, of the Defence Advanced Projects Research Agency or the U.S. Government. Finally, we are indebted to Teamwave Inc. for supplying the software.

7 REFERENCES

1. Fitzpatrick, G. (1998) *The Locales Framework: Understanding and Designing for Cooperative Work.* PhD Thesis, Department of Computer Science and Electrical Engineering, The University of Queensland.
2. Greenberg, S. (1996). Teaching Human Computer Interaction to Programmers. *ACM Interactions*, **3:4**, 62-76, July-August, ACM Press.
3. Gutwin, C. and Greenberg, S. (1999**)** A Framework of Awareness for Small Groups in Shared-Workspace Groupware. TR-99-1, Dept Comp. Science, U. Saskatchewan, Canada. www.cpsc.ucalgary.ca/grouplab/papers.
4. Gutwin, C. and Greenberg, S. (1998). The Effects of Workspace Awareness Support on the Usability of Real-Time Distributed Groupware. RR-98-632-23, Dept Computer Science, U. Calgary, Canada. www.cpsc.ucalgary.ca/grouplab/papers
5. Harrison, S. and Dourish, P. (1996) Re-place-ing space: The roles of place and space in collaborative systems. *Proc Conference on Computer Supported Cooperative Work*, 67-76, ACM Press.
6. Nielsen, J. (1993). *Usability Engineering.* Academic Press.
7. Nielsen, J. (1994a). *Heuristic Evaluation.* In Nielsen and Mack (1994)
8. Nielsen, J. (1994b). Enhancing the explanatory power of usability heuristics. *Proc Conference on Human Factors in Computing Systems*, 152-158, ACM Press.
9. Nielsen, J. and Mack, R.L. (1994). *Usability Inspection Methods.* Wiley.
10. Roseman, M. and Greenberg, S. (1996). TeamRooms: Network Places for Collaboration. *Proc Conference on Computer Supported Cooperative Work.* ACM Press.
11. Shackel, B. (1990). Human Factors and Usability. In *Human-Computer Interaction: Selected Readings*. J Preece and L Keller, eds. Prentice Hall, 1990.