

Groupware Walkthrough: Adding Context to Groupware Usability Evaluation

David Pinelle and Carl Gutwin

Department of Computer Science, University of Saskatchewan
57 Campus Drive, Saskatoon, SK, S7N 5A9, Canada
david.pinelle@usask.ca, carl.gutwin@usask.ca

ABSTRACT

Discount usability evaluation methods have recently been introduced as a way to assess groupware systems. However, one criticism of these techniques is that they do not make use of information about users and their work contexts. To address this problem, we developed groupware walkthrough, a new usability inspection technique for groupware. The technique is a substantive modification of cognitive walkthrough to include consideration for the complexities of teamwork. The two components of groupware walkthrough are a task model for identifying and analysing real-world collaborative tasks, and a walkthrough process for assessing a system's support for those tasks. Groupware walkthrough is a low-cost technique that can identify collaboration-specific usability problems and can find problems that would not be revealed through other inspection methods.

Keywords

Cognitive walkthrough, usability inspection, mechanics of collaboration, groupware walkthrough

INTRODUCTION

As improvements are made in distributed computing technologies, groupware systems are becoming common. In spite of this progress, many groupware applications still have serious usability problems. It has recently been suggested that discount inspection techniques can prove useful in reducing the number of usability problems in groupware [8,2]. These methods are a major departure from traditional field-based approaches of evaluating multi-user systems [7,11]. They can be used early in the design cycle, and are inexpensive since they do not require real users or a functioning prototype. Since these approaches are not carried out in the workplace, they do not presume to address the complex cultural and organizational aspects of group work, leaving assessment of these factors for other techniques. Instead, they address support for the low-level activities of collaboration that are common to any interacting group.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2002, April 20-25, 2002, Minneapolis, Minnesota, USA.
Copyright 2002 ACM 1-58113-453-3/02/0004...\$5.00.

However, one of the problems with existing inspection methods is that they do not consider contextual information about users and tasks. A recent study showed that although inspections can catch many usability problems, a sense of the actual work context would improve results [15]. One example from this study is that evaluators were not aware that certain tasks were carried out asynchronously, and this knowledge would have changed their interpretation of the problems found.

To address the need for context in groupware usability evaluations, we developed *groupware walkthrough*, a methodology based on cognitive walkthrough [13]. Cognitive walkthrough is a successful inspection technique for single-user software that incorporates user descriptions and tasks. However, it does not account for multiple concurrent users or for the dynamic nature of group work. We made substantial modifications to cognitive walkthrough in order to include consideration for the teamwork aspects of collaboration.

Our methodology has two components: a group task model that captures the variability and multiple courses of action in group work, and a walkthrough process that guides evaluators as they step through tasks and evaluate the groupware interface. In this paper, we present the details of groupware walkthrough and how it can be used to carry out a usability inspection. We describe the results of a case study to show how the technique can be applied to a real-world work situation, how it can be used early in application development, and how it is useful at revealing usability problems that would not be found using other discount evaluation methods.

COGNITIVE WALKTHROUGH

Cognitive walkthrough [13] is a popular inspection method for single user software. It allows designers to evaluate software in the early stages of development while still taking the use context into consideration. The technique does not require a working prototype and can be performed without users, providing benefits in terms of time and expense. Cognitive walkthrough involves three types of contextual information: a description of the users and the knowledge they possess, descriptions of the tasks the users will perform with the system, and a list of the correct actions a user must perform to accomplish the tasks with a particular prototype. During a walkthrough, evaluators step through the task sequences and consider

whether users will select the correct actions when performing each task with the prototype.

The success of cognitive walkthrough for single-user software raises the question of whether the technique can be used to evaluate groupware as well. Ereback and Höök [5] investigated this question, and found that although the technique holds promise for CSCW, there were several problems with the method. In particular, the cognitive walkthrough process became more difficult when applied to groups because of the complexities of specifying group tasks and managing multiple user and task descriptions concurrently.

A second set of problems arise from cognitive walkthrough's insistence on a single correct sequence of actions for completing a task. The complexity of group tasks and the interdependence among group members' actions means that the way in which a group task is completed, the order of the task actions, and even the degree of collaboration will be highly variable.

Building on Ereback and Höök's initial work, we have modified cognitive walkthrough to address these problems and make it more suitable for evaluating groupware. To help manage the complexities of multiple concurrent group tasks and user descriptions, we designed a method for analyzing and specifying collaborative tasks [12]. To deal with task variability and alternate courses of action, tasks are modeled as a set of likely alternate paths for achieving an intended outcome. Each possible alternate path is explored during the walkthrough, and the support for each path in the interface is evaluated, but the ability of the user to follow a predetermined sequence of "correct actions" in the interface is not enforced. Instead, the technique focuses on the users' experiences with the interface while carrying out tasks, and the interface's support for helping the group fulfill the intended outcome.

GROUP TASK ANALYSIS

The first component of groupware walkthrough is the task model. We first provide a discussion of the level of analysis that is used in the model, and then set out the model's structure and the steps involved in task analysis.

Level of analysis

Collaboration involves two distinct kinds of work: *taskwork* (the actions that must occur to complete the task) and *teamwork* (the actions that group members must carry out in order to complete a task as a group). Traditional task analysis approaches focus on taskwork (e.g. [1,4]); however, to address the complexities of group work, teamwork must be analyzed as well.

Teamwork can be analyzed at either a high or low level of abstraction. The high level, including social and organizational aspects, is considered in van der Veer et al's Group Task Analysis (GTA) framework [16,17]. This

framework and its associated methodologies focus on task specification based on ethnographic observation. GTA is an effective way to gather information about the work context, but we felt that its level of analysis was too high to be used effectively in a walkthrough situation. To capture the level of detail needed for a walkthrough, we consider teamwork at a lower level, that of the mechanics of collaboration. However, since high-level analysis usually precedes low-level analysis, these two analysis techniques can be used in conjunction with each other.

The *mechanics of collaboration* are basic activities of shared work—the small-scale actions and interactions that group members must carry out in order to get a task done in a collaborative fashion. They are the things that will be common to a shared task even across a variety of social and organizational factors. The mechanics have been introduced elsewhere [8], but are summarized below in Figure 1. The mechanics are a useful level of analysis for a walkthrough because they provide a fine-grained view of teamwork, allowing collaboration to be broken down into specific actions that evaluators can assess one at a time.

<p><i>Explicit communication</i>: intentional provision of information, either through speech, text, or gesture.</p> <p><i>Monitoring</i>: gathering information given off by others through consequential communication or feedthrough.</p> <p><i>Coordination</i>: synchronizing actions and managing access to shared resources.</p> <p><i>Planning</i>: division of labour, reserving areas of the workspace for future use, or plotting courses of action.</p> <p><i>Assistance</i>: provision of help to one another, either upon request or opportunistically.</p> <p><i>Protection</i>: actions taken to prevent change to or deletion of a person's existing artifacts and work.</p>

Figure 1. The mechanics of collaboration ([8]).

Task model

The task model provides a basic framework for analyzing contextual information to support usability walkthroughs [12]. Unlike single-user task analysis methodologies, the task model adds consideration for the mechanics of collaboration, variability in levels of collaboration, and variability in task execution. The model is based on hierarchical decomposition of real-world group work scenarios. When tasks are analyzed, they are arranged in sequential order where appropriate, but alternate or common paths through a task are considered as well. The major components in the model are shown in Figure 2.

Scenarios. Task scenarios are commonly used as a tool in user testing and in discount usability engineering [14,10]. Scenarios are descriptive formalizations of the work users carry out in the real world, typically containing multiple tasks and providing contextual information about the users

and the circumstances under which the tasks are commonly carried out [14]. For our work, scenarios consist of a general activity description, descriptions of users and summaries of the knowledge they are likely to have, an intended outcome, and a set of circumstances in which the scenario is carried out [10].

Tasks. Tasks are the building blocks of scenarios, and may be explicitly stated in the scenario activity description. Tasks are phrased as statements that describe what occurs in a scenario, but not how an event occurs.

Subtasks. Each task can be divided into subtasks that specify how the task is carried out. In group work, tasks can be carried out by different combinations of teamwork and taskwork. For this reason, subtasks are categorized as either individual (i.e. taskwork) or collaborative (i.e. teamwork). Collaborative subtasks are specified using the mechanics of collaboration (Figure 1).

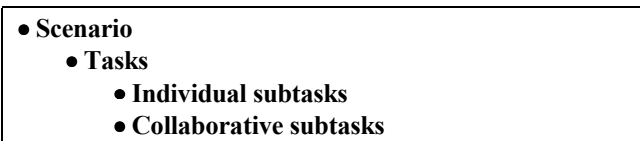


Figure 2. Hierarchical task model

Task analysis process

Group tasks are analyzed by first collecting observational data about the work, then identifying episodes of collaborative interaction within the overall work flow. Scenarios and tasks can then be specified, and collaborative subtasks can be analyzed in terms of the mechanics of collaboration.

Collecting and interpreting observational data

In order to accurately specify scenarios and analyse group tasks, data from the real-world work situation must be collected. Much has been written about how such observations can be conducted (e.g. [3,6,9]); here we focus on how these data are used to generate a group task analysis in preparation for groupware walkthrough.

The first step in making sense of a group’s work is to model the individual scenarios that occur in the work of each group member. The sequences of scenarios for each person can then be combined to determine where interaction and collaboration occur within the workflow (e.g. Figure 3). The scenarios that include collaboration represent the points where each group member’s model converges with that of at least one other group member. These scenarios are the ones that can be investigated in a groupware walkthrough. Scenarios that represent individual work can be evaluated using cognitive walkthrough or another single-user evaluation technique.

Preparing the scenarios

Once collaborative scenarios have been identified from the group scenario diagram, the scenarios can be specified in

greater detail. The scenario specification provides the evaluators with knowledge they can use to help them understand the users and the real world circumstances that surround their tasks, and so more detail in these descriptions is usually warranted. Figure 4 shows an example scenario specification (substantially abbreviated here due to space limitations).

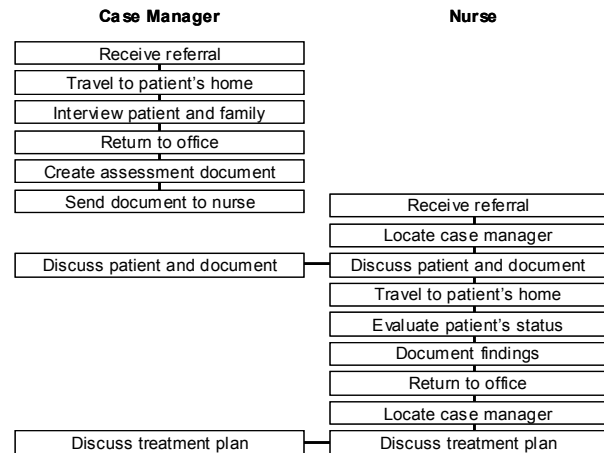


Figure 3. Home care workflows for a nurse and a case manager. Horizontal lines indicate collaboration.

Scenario: Discuss patient and document

Activity description. A nurse receives a care plan and an assessment document from a referring case manager. The nurse reviews the assessment and wants to gather further information before visiting the patient, so she attempts to locate the case manager in the office. When the nurse finds the case manager, she approaches him and asks him for further information about the patient. The nurse places the assessment document on the desk and points to areas of the assessment where she has questions. The case manager and nurse discuss the patient’s situation.

User specification. Case managers are responsible for coordinating community based patient care. They evaluate patients, create care plans, and then refer patients to other community-based services. Case managers are fairly experienced with computer systems.

User specification. Nursing is responsible for providing a wide range of nursing services in the community. They evaluate patients to identify specific care needs and then work with patients and other home care disciplines to meet those needs. Nurses do not currently use computers in their job, and many report being slow typists.

Intended outcome. Exchange information concerning patient. Nurse, in particular, wants further information about the patient before visiting them for the first time.

Circumstances. Nurses and case managers have office space in same building and usually try to meet face to face during the day. However, these meetings are difficult to arrange, and collaboration by other means (such as voice mail) is not always timely or private.

Figure 4. Example home care scenario specification.

Specifying tasks and subtasks

Once the scenario has been recorded, tasks should be extracted from the scenario activity description (an activity description will usually contain several tasks) and from field-based observation data. The approach here is not to determine the absolute sequence of tasks that occur in the scenario, but to identify activities that should be supported in order to cover a reasonable range of possible alternatives.

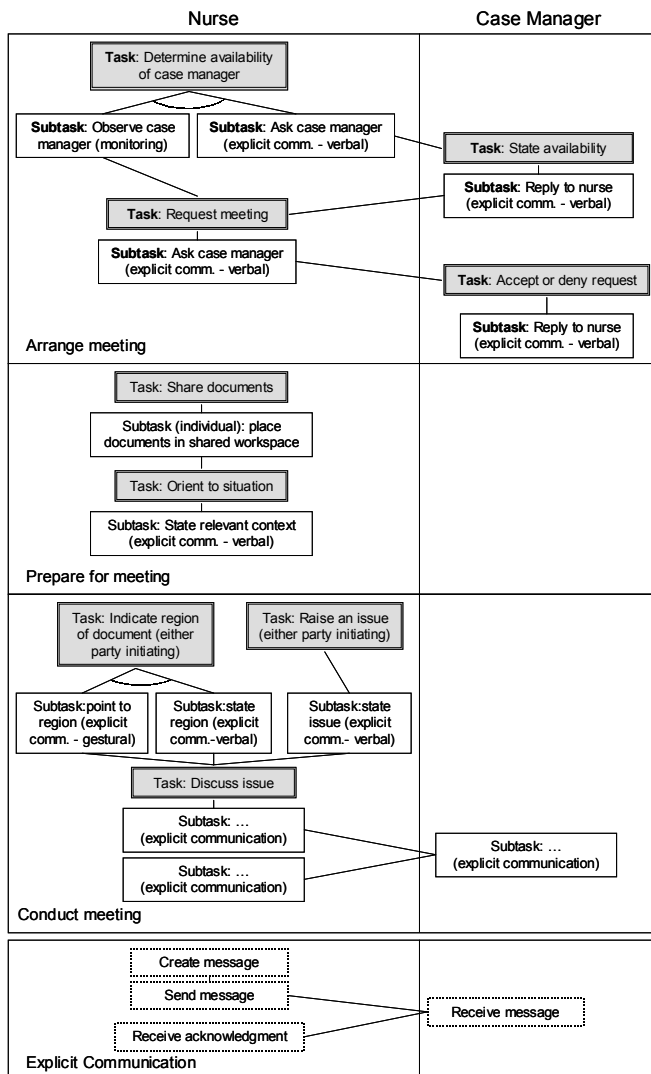


Figure 5. Task analysis diagram for one scenario, and task “subroutine” detailing explicit communication.

We find it useful to represent tasks in a diagram format such as that shown in Figure 5. Diagrams can display relationships between tasks and subtasks, task flow through a scenario, and alternate courses of action (here using AND/OR graph notation).

Subtasks are determined for each task by considering how the task was carried out in the work setting; that is, which

mechanics of collaboration were used. In addition, certain of the mechanics themselves can be broken down into task “subroutines” that have a set of actions that must be carried out. In the example shown in Figure 5, explicit communication can be represented as a sequence of actions in which messages are created, sent, received, and acknowledged (although over some channels certain of these steps will be implicit).

GROUPWARE WALKTHROUGH METHODOLOGY

Once scenario specifications and task analyses have been compiled, a groupware walkthrough for a particular prototype can be carried out. Figure 6 shows the groupware walkthrough process. Evaluators step through the tasks and determine how well the interface supports group members in working toward and achieving the intended outcome. The technique can be applied to any groupware design, ranging from low fidelity prototypes to functioning applications. However, the technique is intended to be formative, where results are used as redesign information in an iterative design cycle.

For each collaborative scenario:

- Review the scenario to become familiar with users, intended outcome, and surrounding circumstances
- For each task in the scenario:
 - Attempt to carry out each alternate subtask.
 - Record how each subtask was carried out.
 - Record problems, but then assume they are fixed and continue.
- After each task, ask the following questions [8]:
 - Can the task be performed *effectively*—does the interface supply the means to do it (and correctly)?
 - Can it be performed *efficiently*—would the group make the effort required to perform the task?
 - Can it be performed with *satisfaction*—would the group be motivated to do this task, and would they be happy with the outcome?

After all tasks, determine whether the interface allows the group to achieve their overall intended outcome.

Figure 6. Steps in a groupware walkthrough

There are several specific issues in the walkthrough process that can be considered in more detail.

Task exploration path. Alternate paths of task execution are modeled in the task analysis, and each possible path should be explored in the groupware walkthrough. We have found that proceeding in a breadth-first fashion through the task analysis results is useful in helping to understand the relationships among tasks and subtasks.

Recording results. Throughout the walkthrough, the evaluators record information that they derive from their experiences with the interface. The information takes three forms: lists of problems they encounter while attempting to

carry out tasks, answers to questions that are used as qualitative metrics, and a general summary of the overall success of the scenario. Questions about effectiveness, efficiency, and satisfaction are intended as more general assessments of tasks [8] and an opportunity for evaluators to state things that do not appear in specific problem reports. The overall summary records how well the interface supports the group in achieving the intended outcome associated with the scenario.

Multiple evaluators. In groupware walkthrough, multiple evaluators help to manage the complexity of concurrent tasks and multiple user descriptions. When multiple evaluators are available, the walkthrough can be carried out in two ways. First, the evaluators can jointly step through the tasks for all users and can discuss the features of the system. Alternatively, each evaluator can take on the role of one of the users and can carry out their tasks as specified in the task analysis.

Managing walkthrough results. Once a groupware walkthrough has been carried out, the evaluators must deal with the pages of notes they generated. The most effective way that we have found to deal with these results is to have a structured meeting in which the design is reviewed, and each evaluator presents their findings as they relate to specific aspects of the design. This is similar to synthesis in heuristic evaluation [10].

CASE STUDY

We used groupware walkthrough to evaluate an early prototype groupware system for home care teams in a local health district. In the following sections we introduce the domain, review our task analysis, and detail the results of the walkthrough.

The work domain: home care teams

In home care, mobile workers provide patients with healthcare services in their home. A patient can receive services from multiple disciplines, including nurses, case managers, and therapists. Workers spend most of their time out of the office, and so they often do not see each other face to face. However, since multiple workers can work with the same patient, their actions are interdependent. In order for home care workers to move toward desirable outcomes, they must work together.

Since home care workers are mobile, operate out of different locations, and have highly variable schedules, it is difficult for them to collaborate. They need to exchange information with others in a timely fashion, but it is difficult for them to meet face to face during the day. Communication methods such as telephones and pagers are not always practical since privacy can be an issue when in a client’s home, and because conference calls cannot currently be set up in the field.

We are developing a groupware application to allow home care teams to communicate, share documents, and hold case conferences. During our early design work, we came up with several low-fidelity mockups, and the interface tested here is one of these.

Task analysis in the home care domain

We followed the analysis process described above to find out about the work domain and specify tasks for a walkthrough. We interviewed members of each home care discipline, and then spent time with each discipline in the field, observing them as they worked.

Once observational data were collected, we modeled the work scenarios for each home care clinician. For this case study, we focused on two roles, a nurse and a case manager. Figure 3 shows example workflows for these two people. Three scenarios were found to have substantial collaborative components. An example scenario description is given in Figure 4. After the scenario specifications were completed, we carried out a task analysis for each scenario. Figure 5 presents the task analysis of the “Discuss patient and document” scenario.

Groupware prototype

Figure 7 shows a prototype interface designed to allow team members to communicate and to use shared documents (e.g. evaluations, progress notes, or discharge summaries) to support discussion about patients.

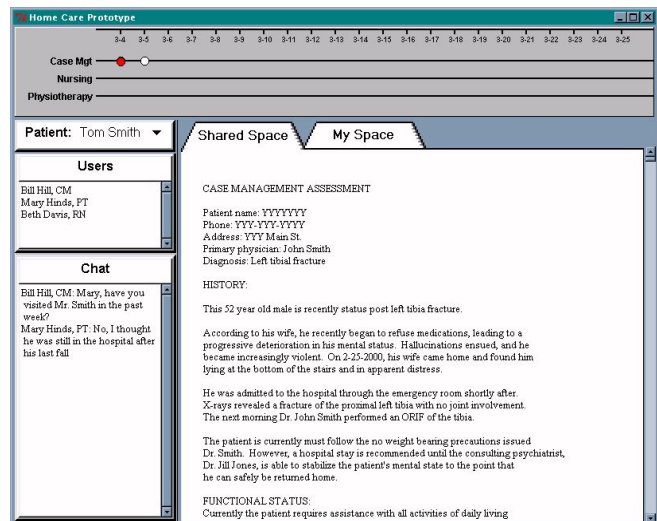


Figure 7. Groupware prototype.

In the interface, documents are visualized on a timeline that indicates the date they were generated and the discipline (e.g. manager, nursing, physical therapy) that created them. Users can view documents in a shared workspace that is visible by the whole group, or in a private workspace (“My workspace” in Figure 7). To select a document, the user clicks on its icon in the timeline at the top of the screen.

Communication is carried out through a text chat tool. A window above the chat area shows a list of users who are currently logged on to the system, and telepointers are available in the shared document space to facilitate awareness of activity and gestural communication.

Groupware walkthrough results

We carried out a groupware walkthrough of the prototype described above. Here we report on some of our results from assessing the tasks shown in Figure 5. In our first example we give a more detailed description of the walkthrough experience, and in the succeeding examples we focus only on the usability problems found.

Example task 1: determine availability of case manager

This task is carried out by the nurse, and can be done in two ways: either by explicitly asking the manager, or by monitoring the case manager to gather evidence about their availability. We considered the latter subtask first. We found that certain kinds of availability information are present in the interface: for example, it is possible to see whether the manager is logged in, whether they are currently engaged in a discussion with someone else using the chat tool, and whether they are currently using their pointer in the shared space. However, if the case manager is logged in but not active in shared work areas, there is not enough information to determine whether or not they are available. After encountering this situation, we noted the problem and moved on to the next subtask.

The second subtask involves asking the case manager about their availability, and is the first explicit communication task that we tested. Since the basics of communication will be the same for most communicative acts, we used this first instance as a general test of this mechanic of collaboration. We examined whether we could create messages, have them arrive at the receiver, and get feedback that the messages had been received.

In attempting to carry out the task, we immediately noticed that there was no entry box in the chat tool – an oversight when the low-fidelity prototype was designed. We noted the problem, and then assumed it was corrected so that we could continue with the task. We then observed that there were two potential problems in communicating with the case manager: first, that it could be difficult for the manager to notice that a relevant message had arrived, and therefore, it could be difficult for the nurse to understand that the message had been received; and second, that there was no way for the nurse to speak only to the case manager, since the chat tool is public to the entire group.

After the task was completed, we considered the three summary questions about effectiveness, efficiency, and satisfaction. We noted that there were problems with effectiveness in determining availability with either of the possible subtasks. When considering efficiency and

satisfaction, we also recorded the possibility that users would have difficulty communicating by typing, since some of the workers are not regular computer users.

Example task 2: share documents

In this task the nurse puts the assessment document into the shared workspace for discussion. While working through this task we encountered a major usability issue: that there is only one shared space, and if it is already in use by other members of the group, then the nurse and manager have no space for their discussion. Even if the space is available, there may be privacy concerns that would lead the nurse and manager to want a separate work area. Furthermore, it is impossible to tell who has moved documents from the timeline to the shared space. A second problem we found is that it is difficult to determine whether the other person is looking at the shared workspace or the private space.

Example task 3: indicate area of document for discussion

In this task, either of the meeting participants can indicate a part of the document for discussion. The task can be carried out through either explicit verbal communication, or explicit gestural communication. For gesturing in the shared space, the system provides telepointers that follow each person's mouse cursor. However, in carrying out this task it became clear that seeing another person's gestures would be impossible if you were scrolled to a different section of the document. When we worked through the other alternate subtask (indicating by verbal communication), it seemed that specifying parts of the document verbally could be difficult and inefficient since certain documents do not contain good landmarks (such as page numbers or regular headings) for creating descriptions. Therefore, regardless of how we carry out the task, we run into either problems with effectiveness or with efficiency.

Summary

The walkthrough revealed a variety of usability problems. Some of these were simple oversights, such as the lack of an entry box for the chat tool, but several others were more subtle. The main issues raised were:

- explicit communication through the text chat tool could be problematic because the receiver of a message might not see the message;
- communication through text chat could also be frustrating because of the variable typing skills present in the user group;
- determining a person's availability through monitoring can be difficult because the system does not provide enough information;
- a single shared workspace and a single text chat area do not allow multiple meetings of subgroups within the home care team;
- identifying items in shared documents through

pointing can fail when the other person is scrolled to another part of the document.

Working through these problems in the walkthrough led to a number of redesign ideas, and we were quickly able to come up with design revisions to address many of the difficulties. For example, we are now considering:

- the capability to send a chat message to a specific set of people rather than the entire group;
- additional notification when a chat message contains a person's name or is explicitly sent to them (highlighting of the message and a preview pane to show the message);
- the capability to create new temporary shared workspaces that appear as additional tabs;
- multi-user scrollbars as an indication of where others are located in a document that is in the shared space;
- idle time indicators for the login list to increase the amount of information for determining availability.

DISCUSSION

In the next sections, we look at the type of results that arose from our case study, consider what our methodology revealed compared to other discount techniques, and discuss where groupware walkthrough fits in the groupware evaluator's toolbox.

Looking at what we found

The groupware walkthrough revealed simple things that might be obvious to a trained designer's eye, and also things that we had not thought of previously. It is useful to note, however, that since the technique is lightweight, it is not necessary for designers to exhaustively revise and polish their designs before evaluating them; instead, groupware walkthrough can be used as much as a design tool as an evaluation tool. The prototype considered in our case study was a very early mockup, and the technique worked well even with this rough design; by having real task information we were forced to confront some of the difficulties that the interface would have. Even though we assessed the prototype at a very early stage, some serious issues were raised – particularly the problems with single and multiple shared workspaces – that will result in fairly substantial redesign work.

In addition, the information gathered in the walkthrough was obtained at relatively low cost: approximately two days of field work spent with the nurse and case manager, one day spent writing up and checking the task analysis, and one-half day spent carrying out the walkthrough itself. For the next prototype that we evaluate, the walkthrough can be carried out in only another half a day. This means that we can now test many iterations of a design, or test many different design directions, fairly cheaply.

How walkthrough works

The value of the walkthrough methodology (both cognitive

and groupware) is that it provides an external record of the details of the users and the task, freeing the designer from having to remember everything about the work situation. The difficulties inherent in managing this complexity are one of the reasons why designer introspection is such a poor technique for evaluating interfaces. The descriptions and processes captured by the walkthrough methodology allow evaluators to address an entire task without having to remember all of the details. In group tasks, this complexity is considerably increased, making it all the more difficult for the designer to keep the relevant constraints and interactions in mind, and so the value of an external reference and structure should be likewise greater.

The complexity inherent in group tasks raises the issue of how groupware walkthrough will scale to larger groups. Although it is clear that task analysis will require more work as group size increases, the method does scale, at least to groups of three and four. One of the major differences between groups of two and groups of three or more is that some of the mechanics of collaboration become more complex. For example, monitoring and communication with larger groups requires that people keep track of how embodiments map to people, something that is easy when there is only one other person in the collaboration. With these additional complications in mind, however, we have found that larger tasks do not become exponentially more complex than smaller ones, but rather include more duplication of activities and more possible alternates.

Adding context to discount methods

The purpose of developing groupware walkthrough was to add real-world context to discount usability evaluation in order to improve on the results of an assessment. In our experience, groupware walkthrough does reveal problems and issues that might not be identified by other discount methods.

The additional contextual information helps evaluators find and interpret both small and large problems. For example, the potential frustrations of typing might not have been uncovered if an inspector was unfamiliar with the users. Similarly, the need for multiple shared workspaces became apparent when we considered the requirements of a specific task involving a nurse and a case manager, but would not have been obvious to an evaluator who was only considering conferences for the entire team. Third, we identified the need for more extensive availability information only after working through the details of the "determine availability" task, a problem that might easily have been overlooked in heuristic evaluation [2,8].

CONCLUSION

In this paper we have described groupware walkthrough, a contextualized usability inspection method that can be a

valuable addition to the evaluator's toolbox. The technique can be carried out at reasonably low cost like other discount usability techniques, but it also adds contextual information about the work domain, enabling evaluators to test the interface in relation to real tasks and real users. Along with other low cost methods such as groupware heuristic evaluation, it makes possible the iterative development and testing of multi-user systems. Groupware walkthrough can be used early in the development process, even on rough low-fidelity prototypes, and allows developers to use formative evaluation as a regular part of the design cycle.

Groupware walkthrough is not intended to replace other field methods: since it uses a fairly low level of analysis (the mechanics of collaboration), it cannot capture many of the social and organizational aspects of groupware use in a real work setting. However, groupware walkthrough can ensure that when a development team does carry out a field evaluation of the system, most of the mechanical usability problems will have been discovered and addressed, allowing the situated evaluation to focus on higher-level issues.

REFERENCES

- Annett, J. and Duncan, K.D. Task analysis and training design. *Occupational Psychology*, 12, (1967), 211-221.
- Baker, K., Greenberg, S. and Gutwin, C. Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration, *Proc. 8th IFIP Working Conference on Engineering for HCI (EHCI 2001)*.
- Beyer, H., Holtzblatt, K. *Contextual Design: Defining Customer-Centered Systems*. Academic Press, San Diego CA, 1998.
- Diaper, D. *Task Analysis for Human-Computer Interaction*, Ellis Horwood, Chichester, 1989.
- Ereback A.-L. and, Höök K. Using Cognitive Walkthrough for Evaluating a CSCW Application. *Proc. CHI '94*, ACM Press, 91-92.
- Garfinkel, H. *Studies in Ethnomethodology*. Polity Press, Cambridge, 1967.
- Grudin, J., 1990, Groupware and Cooperative Work: Problems and Prospects. In *The Art of Human Computer Interface Design*, B. Laurel, Ed. Addison-Wesley, 171-185.
- Gutwin, C. and Greenberg, S. The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. *Proc. 9th IEEE WETICE 2000*, IEEE Press, 98-103.
- Hughes, J., King, V., Rodden, T., Andersen, H. Moving out from the control room: ethnography in system design. *Proc CSCW'94*, ACM Press, 429-439.
- Nielsen, J. *Usability Engineering*. Academic Press, Boston MA, 1993.
- Orlikowski, W.J. Learning From Notes: Organizational Issues in Groupware Implementation. *Proc. CSCW '92*, ACM Press, 362-369.
- Pinelle, D. and Gutwin, C. Group Task Analysis for Groupware Usability Evaluations. *Proceedings of the 10th IEEE WETICE 2001*, IEEE Press, 102-107.
- Polson, P., Lewis, C., Rieman, J., and Wharton, C. Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces. *International Journal of Man-Machine Studies*, 36, (1992), 741-73.
- Rubin, J. *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. John Wiley & Sons, New York, 1994.
- Steves, M., Morse, E., Gutwin, C. and Greenberg, S. A Comparison of Observation and Inspection Methods for Evaluating Groupware. *Proc. GROUP 2001*, ACM Press, 125-134.
- van der Veer, G.C., Lenting, B.F., and Bergevoet, B.A.J. GTA: Groupware Task Analysis – Modeling Complexity. *Acta Psychologica*, 91, (1996), 297-322.
- van der Veer, G.C. and van Welie, M. Task Based Groupware Design: Putting Theory Into Practice, *Proc. DIS2000*, ACM Press, 326-337.