Group Task Analysis for Groupware Usability Evaluations

David Pinelle and Carl Gutwin
Department of Computer Science, University of Saskatchewan
57 Campus Drive, Saskatoon, SK, Canada S7N 5A9
[david.pinelle, carl.gutwin]@usask.ca

Abstract

Techniques for inspecting the usability of groupware applications have recently been proposed. These techniques focus on the mechanics of collaboration rather than the work context in which a system is used, and offer time and cost savings by not requiring actual users or fully-functional prototypes. Although these techniques are valuable, adding information about task and work context could improve the quality of inspection results. We introduce a method for analysing group tasks that can be used to add context to discount groupware evaluation techniques. Our method allows for the specification of collaborative scenarios and tasks by considering the mechanics of collaboration, levels of coupling during task performance, and variability in task execution. We describe how this type of task analysis could be used in a new inspection technique based on cognitive walkthrough.

Keywords: groupware evaluation, task analysis, mechanics of collaboration, groupware walkthrough

1. Introduction

Groupware is traditionally considered to be difficult to evaluate, especially when compared to single-user software. Group dynamics, social culture, and organizational structure are highly variable among user groups [4, 9]. These contextual factors should be addressed early in groupware development; however, it is difficult to evaluate an application's success in the absence of real-world users and work contexts [5]. Ethnographic evaluation techniques are commonly utilized to evaluate groupware, but these techniques often do not fit easily into early stages of groupware development because of their costs and requirement for fully-functional prototypes.

There has been recent interest in developing formative usability inspection methods that look at groupware applications from a different perspective – that of the basic mechanics of collaboration [1,6]. These techniques are inexpensive since they do not require real users or a functional prototype. The techniques do not attempt to address the situated cultural or organizational aspects of group work within particular workplaces. Instead, they address the low-level characteristics of collaboration that must be addressed in order to create a usable groupware application. Gutwin and Greenberg [6] present these characteristics as a set of mechanics that include explicit communication, implicit communication, coordination, planning, monitoring, assistance, and protection.

In a recent study [14] the mechanics of collaboration were used to carry out a usability inspection of a groupware application. One of the findings was that although a purely mechanical view can catch many usability problems, it would be valuable to have discount techniques that maintain a sense of the actual tasks that will be carried out in the system. For example, evaluators were not aware that group members tended to carry out their work asynchronously, and this knowledge would have changed the set of assumptions they used during the evaluation [14].

To address the need for context in discount groupware evaluations, we have developed a task analysis model for group work that adds valuable task information to inspection techniques. The task model provides a framework for analyzing real-world contextual information in preparation for carrying out an inspection or walkthrough. Unlike traditional task analysis methodologies, the task model adds explicit consideration for collaborative processes such as the mechanics of collaboration, levels of coupling during task performance, and variability in task execution.

In this paper, we first outline the mechanics of collaboration, and then present the task model and discuss how it can be used to specify and analyze group work. We close by discussing how the group task descriptions obtained through the analysis could be used in task-based evaluation methodologies such as cognitive walkthrough.

2. The Mechanics of Collaboration

Recent groupware inspection methodologies are based on the claim that some groupware usability problems are not strongly tied to social or organizational issues, but rather are caused by insufficient support for the basic activities of collaboration. These *mechanics of collaboration* represent the small-scale actions and interactions that group members must carry out in order to get a shared task done [6]. These actions are part of the teamwork (the work of working together) rather than part of the taskwork (the work that carries out the task). There are seven major activities that are covered by the mechanics of collaboration.

Explicit communication. Group members must be able to provide each other with information. Verbal, written, and gestural communication are cornerstones of collaboration.

Implicit communication. People also pick up information that is implicitly produced by others going about their activities—information from artifacts being manipulated, or information from others' movements and actions.

Coordination of action. People organize their actions in a shared workspace so that they do not conflict with others. Shared resources and tools require that turns be taken, and some tasks require that actions happen in particular orders.

Planning. Some types of planning activities are carried out in a shared workspace, such as dividing up the task, reserving areas of the workspace for future use, or plotting courses of action by simulating them in the workspace.

Monitoring. People generally need to keep track of others in the workspace: who is there, where they are working, and what they are doing. In addition, situations such as expert-novice collaboration require more explicit monitoring.

Assistance. Group members provide help to one another when it is needed. Assistance may be opportunistic and informal, where the situation makes it easy for one person to help another, or it may be explicitly requested.

Protection. One danger in group work is that others may alter others' work inappropriately. People therefore keep an eye on their artifacts and take action to protect their work.

In the task analysis method described in the next section, the mechanics of collaboration will be used as the lowest level of task description – that is, the mechanics are the vehicle through which the group work is carried out.

3. Group task analysis

The addition of one or more people to a task leads to a variety of differences in task execution and performance when compared with single-user tasks. In particular, we use the following as main assumptions underlying our analysis method:

- Task descriptions should include consideration for a variety of coupling styles – both loosely-coupled and tightly-coupled interaction should be considered valid ways of carrying out particular tasks;
- Task descriptions should allow alternate courses of action – group task execution is more variable than with individual tasks, and descriptions that are strictly linear can cause important alternate paths to be overlooked;
- Task descriptions should be rigorously analyzed so that they include the mechanics of collaboration that are essential for carrying out the task (e.g. communication with gesturing, or monitoring another member's actions).
- It is not practical to identify a correct sequence of actions to complete tasks in a groupware system due to the high degree of variability in group work (e.g. due to coupling, interdependence of actions and goals).

We begin our discussion of the task analysis model by outlining the difficulties inherent in specifying group tasks. Then we present the task model and discuss each of its component parts, and finally discuss how the task model can be used to analyze contextual information from real world work settings.

3.1. Difficulties specifying group tasks

It is difficult to capture the complexities of collaboration in task descriptions. Groupware must support both taskwork and teamwork, and standard task analysis methodologies (e.g. see [2]) are oriented toward capturing taskwork. It is not clear how teamwork should be captured and represented in task descriptions.

Many common group tasks such as idea generation, plan generation, and decision making [7] are highly interactive and do not have a linear sequence of steps; therefore, they are difficult to specify exactly. This complexity brings into question the assumption in which users have initial goals upon entering the system and that a linear sequence of "correct actions" can be identified to reach those goals. In a group, a user might start without clear goals and decide their own actions by reacting to the actions of other users [15]. Users' goals and users' actions are interdependent, and each user often uses the actions of others and shared plans as a context against which they decide their next action.

The degree of variability in group task execution is increased by differences in levels of coupling while carrying out group tasks. Coupling refers to the level of interaction among group members while carrying out a task, and can range from loosely-coupled interactions, where little group interaction takes place, to tightly-coupled interactions, where users interact frequently. Users tend to switch between these modes regularly while carrying out group tasks. However, the mode users decide to use at any given time is unpredictable and linear task descriptions may overlook one of these modes.

3.2. Task model

In order to address the difficulties inherent in specifying group tasks, we developed a task model to facilitate the analysis of real-world group work scenarios. The task model does not use sequential ordering as its major means of organization, but instead organizes tasks into a hierarchical structure. This hierarchical structure accommodates the uncertainty of group task execution. A reasonable range of methods for carrying out a task can be represented in the analysis results, and then the possibilities can be examined in an evaluation. All steps in the analysis results, therefore, are not necessarily needed to accomplish a task. Likewise, tasks (and their component parts) may be repeated, carried out iteratively, or in some cases executed in alternate sequences.

The major components of the task model are scenarios, tasks, individual and collaborative subtasks, and actions. These are shown below in their hierarchical order, and are described in turn below.

Scenario: High-level description of activities related to achieving a specific outcome. Scenarios contain the following information: high level activity description, user specification, group goal, set of circumstances

- **Tasks:** Basic components of scenario, usually explicitly stated in scenario activity description. Describes *what* occurs in a scenario event, but not *how* it occurs
  - **Individual subtasks:** The loosely-coupled, individual work in a task
  - **Collaborative subtasks:** The tightly-coupled work in a task. Carried out by a mechanic of collaboration
    - **Actions:** Common ways to perform a collaborative subtask

Scenarios. Task scenarios are commonly used as an evaluation tool in user testing and in discount usability engineering [8,13]. Scenarios are a descriptive formalization of the work that users will likely perform using the software. A scenario typically contains multiple tasks and provides contextual information about the users and the circumstances under which the tasks are commonly carried out [13]. For our purposes here, a scenario consists of the following elements: a high-level activity description, a user specification (user description and the knowledge users are likely to have), an outcome (the intended group goal in this case), and a set of circumstances under which the scenario is carried out [8].

Tasks. Tasks are the building blocks of scenarios, and they are often explicitly stated in the scenario activity description. Tasks are high-level statements that describe *what* occurs in a scenario, but not *how* an event occurs.

Subtasks. Each task can be analyzed at a finer granularity and can be divided into subtasks that specify *how* a task is carried out. In group work, tasks can often be accomplished with different levels of coupling among group members. For this reason, tasks are analyzed in terms of *individual subtasks* (loosely-coupled) and *collaborative subtasks* (tightly coupled). Some tasks are realistically accomplished with only a single level of coupling (e.g. decision-making tasks require tightly coupled interactions). However, in other tasks, both collaborative and individual subtasks can be included as viable alternate methods for accomplishing the task.

The mechanics of collaboration provide a useful framework for specifying collaborative subtasks. Many of the teamwork aspects of real-world work are easily overlooked when specifying tasks and translating them into groupware applications. For example, gestural communication may be an important part of completing a task, but it is more easily overlooked than is verbal communication. The mechanics of collaboration provide a thorough approach for capturing these aspects in collaborative subtask specifications.

Actions. The mechanics of collaboration do not provide a complete picture of task execution. Therefore, we include an additional specification in the task model. Each collaborative subtask can be carried out through a set of possible actions. For example, if a collaborative subtask is to identify an object (an explicit communication mechanic), common actions for accomplishing this may be verbal communication or pointing. Either of these actions is sufficient for accomplishing the identification subtask; therefore, actions are usually presented as a list of reasonable alternatives.

3.3. Using the task model

In this section we describe how the task model can be used to analyze real-world contextual information. We discuss how the scenario should be specified and how the task analysis should be conducted.

Scenarios should specify real-world work activities that are carried out by the target users. A scenario should specify an activity that the target users commonly perform in the real world—an activity that they are also likely to want to carry out using the groupware system. Scenario activity descriptions can be specified by observing the target users while they perform work activities. The activity description should be recorded as a high-level description of the activities required to achieve a group goal.

In addition to the activity description, the scenario should contain: a description of the desired outcome (the group goal), a description of the group of likely users, and a description of the circumstances under which the scenario is commonly performed. These components provide important contextual information that is needed when the walkthrough is conducted. It is important to capture information about the types of users that are likely to perform the specified scenario, including their expertise and knowledge. This information can be useful to evaluators when they attempt to ascertain how users will approach the system. Likewise, common circumstances surrounding the scenario provide important insight to the evaluators. For example, information about the physical location of group members during the scenario (i.e. widely distributed or co-located) or about timing of interactions (i.e. synchronous or asynchronous collaboration) could prove valuable to the evaluators when assessing the groupware's usability in supporting the scenario. Finally, goal descriptions provide insight into the motivation of group members and also provide an important metric for evaluating the system.

Once the scenario has been recorded, the activity description must then be analyzed to extract tasks, subtasks, and actions. The approach here is not to determine the absolute sequence of tasks that will occur in the scenario, but instead to identify activities that should be supported in order to cover a reasonable range of possibilities.

The tasks should be extracted from the scenario. Individual tasks are typically still high-level descriptions of activities, and are often explicitly stated within a scenario. Tasks state *what* is being done, not *how* it is being done. These high-level task descriptions should then be broken into subtasks. Each task can potentially have several subtasks. Subtasks describe *how* the task will be accomplished. In order to accurately record a subtask, consideration must be given to likely levels of coupling in each subtask—that is, will a subtask be carried out with loosely coupled interactions (individual subtask) or with tightly coupled interactions (collaborative subtask). If both types of coupling are common, then they should both be recorded so they can be assessed during the walkthrough. While subtasks may be sequential at times, they also may be listed as alternate methods for accomplishing the task.

Collaborative subtasks require special consideration during the analysis process. Each collaborative subtask should correspond to one of the mechanics of collaboration, and this should be recorded in the subtask. However, the mechanics do not specify exactly how they should be carried out. For each collaborative subtask, then, it is necessary to specify the action that will be carried out to accomplish the subtask. These actions should be listed as a set of likely alternatives that can then be explored during the walkthrough.

3.4. Example task description

As an example of the method, we present one part of a task description derived from a two-person car shopping scenario. The group goal is to identify a car for possible purchase from a set of candidates, and the likely users are couples who will share use of the car, and who likely each have a preconceived notion of the features they are looking for in a car. The following task description is based on the circumstances of two users co-located in an automobile showroom, who may explore the showroom individually or collaboratively. The example shows the first two tasks in the scenario (explore available cars, and identify a car for closer inspection)

Task I. Explore the available cars
- Subtask A. INDIVIDUAL: Individual exploration of showroom
- Subtask B. COLLABORATIVE: Monitor other person [*Monitor]*
  - Action 1. Observe location, orientation, actions, and gaze direction

Task II. Identify a car for closer group inspection
- Subtask A. COLLABORATIVE: Explicitly indicate interest [*Explicit communication]*
  - Action 1. Pointing with finger or gaze
  - Action 2. Verbal indication with referring gestures
  - Action 3. Verbal indication based on car features
  - Action 4. Combined position and verbal indication
- Subtask B. COLLABORATIVE: Implicitly indicate interest [*Implicit communication]*
  - Action 1. Actions indicate interest
- Subtask C. COLLABORATIVE: Monitor other person [*Monitor]*
  - Action 1. Observe location, orientation, actions, and gaze direction

Task III. …

Figure 1. Partial task description for a car purchase scenario. Mechanics are shown in [ ].

4. Using task descriptions in evaluation

Group task descriptions can be used to add task context to a variety of groupware inspection techniques. However, perhaps the most appropriate methodology to take advantage of the task information is cognitive walkthrough. We have begun the development of a methodology called *groupware walkthrough*, an adaptation of cognitive walkthrough for the purpose of evaluating groupware applications [11].

Cognitive walkthrough is a commonly utilized usability inspection technique for single user software development [12]. It was developed to allow designers to evaluate software in the early stages of design while taking into consideration common user tasks and goals. The groupware walkthrough methodology is a substantive modification of cognitive walkthrough. Changes were made to accommodate multiple user descriptions, uncertainty in group task performance, and group work metrics.

The task descriptions from the task analysis model form the basis for the walkthough. To carry out a groupware walkthrough, evaluators step through the task analysis results and explore support for each task, subtask, and action within the prototype or application. A strict sequential execution of these steps is not necessary, but breadth of feature testing is important for covering a wide range of potential collaborative situations. In instances where two levels of coupling are supported in subtasks, these should both be explored as viable alternatives for carrying out a task. Likewise, evaluators should give consideration to alternate actions and how support (or lack thereof) will affect the usability of the system.

5. Conclusions

In this paper, we introduced a method for analysing and specifying group tasks. This work was motivated by the need to incorporate contextual information into formative groupware evaluations. The task model takes into consideration the mechanics of collaboration, levels of coupling, and variability in collaborative task performance.

The task model presents a hierarchical structure for analyzing real-world task scenarios. When these scenarios are analyzed, information is partitioned into tasks, individual subtasks, collaborative subtasks, and actions. We discussed how the model can be used to extract contextual information from real-world work settings. The descriptions from this task-analysis method can be used to contextualize a variety of inspection techniques including groupware walkthrough, a methodology based on cognitive walkthrough. Our future work in this area involves fleshing out the groupware walkthrough technique and testing both the analysis and evaluation methods in real groupware design situations.

Acknowledgements. We gratefully acknowledge the support of the National Institute of Standards and Technology (NIST) and the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] Baker, K., Greenberg, S. and Gutwin, C. Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration. To appear, *Proc. 8th IFIP EHCI*.

[2] Dix, A., Finlay, J., Abowd, G., Russell, B. *Human-Computer Interaction*. Prentice Hall Europe, 1998, 408-412.

[3] Greenberg, S., Fitzpatrick, G., Gutwin, C. and Kaplan, S. Adapting the Locales Framework for Heuristic Evaluation of Groupware. Proceedings of OZCHI'99 Australian Conference on Computer Human Interaction.

[4] Grudin, J. Groupware and Social Dynamics: Eight Challenges for Developers. Commun. ACM 37, 1, 92-105.

[5] 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.

[6] Gutwin, C. and Greenberg, S. The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces. *Proceedings of 9th IEEE WETICE Workshop*, 98-103.

[7] McGrath, J. E., 1984, *Groups, interaction, and performance*. Englewood Cliffs, N. J.: Prentice Hall.

[8] Nielsen, J., 1993, *Usability Engineering*, Boston, MA, Academic Press.

[9] Orlikowski, W.J. Learning From Notes: Organizational Issues in Groupware Implementation. *CSCW '92*, 362-369.

[10] Pinelle, D. and Gutwin, C. A Review of Groupware Evaluations. *Proceedings of 9th IEEE WETICE*, 86-91.

[11] Pinelle, D. & Gutwin, C. Groupware Walkthrough: Adding Context to Groupware Usability Evaluation. In preparation.

[12] 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, 741-773, 1992.

[13] Rubin, J., 1994, *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests*. New York: John Wiley & Sons, Inc.

[14] Steves, M., Morse, E., Gutwin, C. and Greenberg, S. A Comparison of Observation and Inspection Methods for Evaluating Groupware. In submission, *GROUP 2001*.

[15] Suchman, A.L. *Plans and Situated Action: The Problem of Human-Machine Communication.* Cambridge University Press, 1987.