
Heuristic Evaluation Techniques for Collaborative Software

Abigail Kirigin

The MITRE Corporation
202 Burlington Road
M/S M325
Bedford, MA 01730 USA
abigail@mitre.org

Gary Klein

The MITRE Corporation
7525 Colshire Drive
M/S H505
McLean, VA 22102 USA
gklein@mitre.org

Len Adelman

George Mason University
Department of Systems
Engineering & Operations Research
4400 University Dr.
Fairfax, Virginia 22030-4444 USA
ladelman@gmu.edu

Abstract

Heuristic Evaluations are most often conducted to evaluate the usability of a software system's interface. However when evaluating collaborative software, it is critical to understand not only how well the interface design meets these general standards, but also how well it is designed to meet the collaboration needs of the users. In order to address this issue, the traditional heuristic evaluation process was modified to assess the tools' usability in supporting collaborative behaviors defined in a previously developed Collaboration Evaluation Framework (Klein and Adelman, 2005).

In this report we describe the key findings from our heuristic evaluation of the collaborative usability of Groove v3.0, InfoWorkSpace v2.5, and Lotus Sametime. For instance, failure to preserve common ground had a major impact on the effective use of all three tools. Generally, the tools need to find the middle ground of attracting the users' attention without distracting them from their tasks. Transmission flexibility was the greatest success of all three tools.

Keywords

Computer supported collaborative work, usability methods, heuristic evaluation

ACM Classification Keywords

H.5.3: Evaluation/methodology, computer-supported cooperative work

Introduction

Heuristic Evaluations are most often conducted to evaluate the usability of a system's interface. Heuristics such as Nielsen and Molich's (1990) support evaluation of systems meant to support a single user's task, such as document or photo editing. However, collaborative software is designed to support multiple users jointly working on a task. Collaborative software supports the collaboration itself, not each user's individual tasks. Therefore, when evaluating collaborative systems, it is critical to understand not only how well the interface design meets general heuristic standards, but also how well it is designed to meet the collaboration needs of the users. Rather than evaluate the usability of tools for each individual's task process, we were interested in evaluating the usability of the tools to facilitate jointly doing the task process. In order to address this issue, the traditional heuristic evaluation process was modified to assess the tools' usability in supporting the collaborative behaviors that facilitate mutual participation in the task process. These behaviors have been defined in a previously developed Collaboration Evaluation Framework (Klein and Adelman, 2005), which is described briefly in the next section.

After introducing the reader to the Collaboration Evaluation Framework, we describe the methodology and key findings of a heuristic evaluation of the collaborative usability we conducted on Groove v3.0, InfoWorkSpace (IWS) v2.5, and Lotus Sametime. Groove, IWS and Sametime are three collaborative

software tools meant to support computer-based group work, through features such as file sharing, text chatting, and shared whiteboards.

The Collaboration Evaluation Framework

The Collaboration Evaluation Framework (CEF) was developed to assess the impact of collaboration per se on the performance of a task. The CEF shows how the characteristics of a joint task process are related to the collaborative behaviors and task transmissions of the participants. This allows one to evaluate a technology's impact on the process, behaviors and transmissions.

Based on Thompson's (1967) seminal work the CEF describes the types of task environments in which collaboration occurs, and the dimensions of coordination, task processes, and interdependence found in joint task processes. Applying ideas from Clark (1996) the CEF further decomposes the classes of coordination into collaborative behaviors. Combined with traditional usability analyses, the CEF provides a way to evaluate the tools against critical elements of collaboration, and understand where the tool needs to improve in order to more successfully support the joint task. For the purposes of the current evaluation, the usability analysis focused only on the tools' support for the collaborative behaviors that are needed for coordination.

Thompson (1967) defines three types of coordination: standardized, planned, and mutual adjustment.

Under standardization, there are established rules or routines for how people should coordinate their activity. As with traffic rules, standardization improves performance per unit cost, by reducing coordination

costs in both financial and cognitive terms because rules remove many uncertainties about how people should coordinate their behaviors. Standardization functions best in stable task environments.

In some task environments, team members must plan their coordination processes based on the task at hand. They will establish task-dependent schedules, work assignments and milestones.

When the task environment doesn't lend itself to standardization or even planning, team members have to co-ordinate through continuous mutual adjustment to each others' activities. This requires constant communication to make sure that coordination requirements (and expectations) are clear and that activities are performed with minimal confusion and maximum benefit. As a result, mutual adjustment is the most costly form of coordination. This can happen, for example, when the task environment is very dynamic and unpredictable.

Clark (1996) describes the behaviors that people engage in to carry out joint actions, like a conversation. Extrapolating from Clark at least eight collaborative behaviors can be identified:

- Connection – locating with whom to collaborate and how to contact them
- Transmission – sending a message
- Notification – alerting the intended party of an incoming transmission
- Identification – designating the sender, receiver, and subject of a transmission

- Common Ground Preservation – establishing and maintaining a shared context and meanings in transmissions
- Confirmation – notifying the sender of a transmission that it has been received
- Synchronization – orchestrating actions to facilitate joint action
- Election – group process of selecting among alternatives

Each type of coordination requires a different subset of collaborative behaviors.

Often, team members use mutual adjustment to plan their coordination processes based on the task at hand. They will then establish planned task-dependent schedules, work assignments and milestones, some of which may be standardized. Therefore, the usability analysis assessed the tools functions with regard to all eight behaviors, and then assessed the functions' suitability to support each form of coordination.

<u>Collaborative Behaviors</u>	<u>M</u>	<u>P</u>	<u>S</u>
Connection	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Notification	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Identification	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Common Ground Preservation	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Transmission	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Confirmation	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Synchronization	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Election	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 1. Different coordination tasks [(M)utual Adjustment,

(Planned, Standardized] require different collaborative behaviors.

If the team coordinated through mutual adjustment, they would need to identify with whom to connect, notify them when information has been sent, transmit and identify the nature of the information, confirm that the information has been received and any subsequent synchronization of when to respond to it.

In contrast, if they can standardize on a procedure for updating a shared file synchronized postings, then additional human actions for connection, notification, confirmation, and synchronization can be virtually eliminated. So, technology (in this case a shared database) actually can facilitate moving from the more numerous (and therefore more costly) mutual adjustment behaviors to less expensive (and faster) standardization.

Methodology

The process for conducting a heuristic evaluation for collaborative usability was nearly the same as for conducting a traditional evaluation, in that the usability analysts evaluated the software against a set of heuristics, recording gaps, suggesting improvements, and noting exceptionally well-designed features. However, to exercise the collaborative functionality of the tools, the usability analysts used them both in a standalone mode, as one would for a traditional evaluation, and also to collaborate with either other usability analysts or other users familiar with the tools. They asked themselves questions relating to the collaborative usability of the tools, drawing from the principles in the Collaboration Evaluation Framework. For each feature of the tool, the usability analysts

asked, "To what extent does the feature facilitate each of the eight collaboration behaviors?" For each collaboration behavior, they would first describe the feature's capability to support it, if one existed. Then they would record the usability findings, both positive and negative, which affect the users' ability to accomplish that collaboration behavior. Dissecting the feature in this way enabled them to conclude which coordination type(s) — Mutual Adjustment, Planned, Standardized —the feature effectively supported and why.

The following is an excerpt of the analysis conducted on Groove's Workspace Chat feature:

Common Ground Preservation: The chat workspace window begins as a piece of the workspace window. When it is there, the workspace window provides it with a direct link to the "common ground". However, the chat workspace window can be separated from the workspace window, and once that happens the link between the chat and the 'common ground' is broken.

Usability Implications: Gap - Allowing the workspace chat to open in its own window potentially breaks the 'mental link' between that chat space and the workspace. Users may not realize that what they type there shows up to all members in the workspace. Users lose the context of the chat when it can be separated from the workspace.

Suggested modifications: The benefit to allowing the workspace chat to open in its own window is that it allows for more notification space. So there is a tradeoff that needs to be mitigated here. Here are two options: ideally 1) do not allow the workspace chat to be separated into its own window but instead allow the chat window more space

to grow in the chat window, but an alternative is 2) if chat popup windows must be allowed, mark them more clearly as tied to the workspace chat.

Coordination suitability: Because the Workspace Chat feature supports many of the collaborative behaviors, it is suited for use during Mutual Adjustment Coordination. The feature's faults in common ground preservation and notification are ameliorated due to the ongoing discussion nature of mutual adjustment. Ongoing discussions will prevent people from losing the context of the window(s) they are in, which would occur more easily if they were to leave and come back and have to remember where they were and what they were doing. As long as members are active in the chat, they will see messages when they arrive, before they get moved off of the screen.

Key Findings

Overall, Groove, IWS and Sametime have a similar core feature set, which provides mutual adjustment through chat, and facilities for planned and standardized transmission through bulletin boards and file cabinets. They also provide shared drawing spaces and screen sharing capabilities. However, they could better support the coordination tasks by improving the design and usability of features that support the collaboration behaviors.

Connection: All three tools could have benefited from more flexibility in connection. Each tool requires that participants first register with the software by establishing an account. In all three, users must have the software up and running on their machine in order to be invited to join a workspace (Groove) / room (IWS) / meeting (Sametime). All three tools would benefit from tying their connection features into a

ubiquitous tool such as email. This way, when one person wishes to have another person join them in the tool, they could simply send an invitation sent to the recipient's email box, which would allow that person to quickly sign in and connect to the sender. Even though the tools may support this type of feature, they are not automatically configured to support it. To the extent that the tools already support this feature, it should be automatically configured by the software upon installation.

Notification: Generally, the tools need to find the middle ground of attracting the users' attention, without distracting them from their tasks. Under some circumstances, Groove's alerting system may overload the user with updates, and force them to break the context of their work to attend to the alert, while IWS and Sametime do little to no alerting as to the presence of unread information.

Identification: Sametime and IWS could be made more effective through additional information identification. Neither tool put date nor time stamps on chat messages sent, leaving participants in the dark as to freshness of information and to the collaboration environment in general. Neither Groove, IWS, nor Sametime labeled source or date in drawing and other co-editing tools.

Common Ground Preservation: Failure to preserve common ground had a major impact on the effective use of all three tools. Groove and IWS allowed elements of the workspace/room to be separated into their own windows, which may destroy the link between elements of those spaces. These tools can leave users at a loss as to who has access to which windows and

where things belong. Sametime's meeting room did not allow these separations; however, Sametime's entire tool suite had two entirely separate applications, one for mutual adjustment and one for planned and standardized communication. These applications had no interoperation link. In order to enable successful collaboration there would need to be a common ground between the three types of coordination across the three applications.

Transmission: Transmission flexibility was the greatest success of all three tools. All three allowed a range of transmission types including text, files, audio, video, line art, and pictures. Providing multiple options for transmission type increases the richness of collaboration possibilities.

Confirmation: Increased automatic confirmation would improve all three tools. Users are not able to confirm automatically what other people are looking at or have seen. None of the tools provided support for viewing a history of operations in the tool.

Synchronization: None of the three tools provided a structured environment in which to synchronize their actions. For all three, chat would probably be the best feature to facilitate activity coordination. Although more effective than email, discussion posts or other features meant to facilitate planned/standardized coordination, chat is still less effective than face-to-face and voice methods, as it lacks the other cues commonly found in these mediums, such as gestures and intonation.

Conclusion

When evaluating collaborative software, it is critical to understand not only how well the interface design meets general heuristic standards, but also how well it is designed to meet the collaboration needs of the users. By modifying a traditional heuristic evaluation with the collaborative behaviors and coordination tasks from the Collaboration Evaluation Framework, one can successfully assess the usability of collaborative software.

The heuristic evaluation of collaborative usability that was conducted on Groove, IWS and Sametime has enhanced our understanding of the factors that contributed to differences in performance between the teams using each of these tools, and will help designers of these tools produce products that more completely and therefore more successfully support collaboration.

References

- [1] Clark, H.H. 1996. Using Language. Cambridge University Press.
- [2] Klein, G.L. & Adelman, L. 2005. A Collaboration Evaluation Framework. 2005 International Conference On Intelligence Analysis, McLean, VA.
- [3] Thompson, J.D. 1967. Organizations in action. NY: McGraw-Hill.
- [4] Nielsen, J. and R. Molich. 1990. Heuristic Evaluation of User Interfaces. New York, NY: ACM Press.

Acknowledgements

The work upon which this paper is based was supported by funding from the intelligence community. We wish to thank the funding organization and other contractor organizations that supported that work.