Evaluating Groupware Prototypes with Discount Methods

Kristin Dew
University of Washington
Dept. of Human Centered Design & Engineering
Box 354809
kndew@uw.edu

Anne M. Turner
University of Washington
Dept. of Health Services
Box 354809
amturner@uw.edu

Loma Desai
University of Washington
Information School
Box 354809
lomad@uw.edu

Nathalie Martin
University of Washington
Dept. of Biomedical & Health Informatics
Box 354809
nmartin@uw.edu

Katrin Kirchhoff
University of Washington
Dept. of Electrical Engineering
Box 352500
kk2@uw.edu

CSCW '15 Companion, Mar 14-18 2015, Vancouver, BC, Canada
ACM 978-1-4503-2946-0/15/03.
http://dx.doi.org/10.1145/2685553.2699002

Abstract
Evaluating a prototype is necessary to user-centered software design, but evaluating groupware systems prior to full deployment can be challenging and costly. Existing groupware evaluation methods focus on individual users, expert inspections, or require the system to be rolled out. We describe a method – based on Gutwin and Greenberg's mechanics of collaboration (MoC) framework – for evaluating prototype groupware systems that have not yet been deployed with the minimum number of users needed to be truly functional. We believe this is a valuable method for evaluating early prototype groupware.

Author Keywords
Usability testing; groupware; scenarios; evaluation; user-centered design; mechanics of collaboration.

ACM Classification Keywords
H.5.2 User Interfaces: evaluation/methodology, user-centered design. H.5.3 Groupware and Organization Interfaces: computer-supported cooperative work, evaluation/methodology.

Introduction
Evaluating groupware, especially in early iterations, can be difficult and expensive. Usability testing with groupware that requires a certain minimum number of

users to be collaborative compounds these issues. We have been developing a groupware prototype machine translation (MT) and post-editing tool called the Public Health Automatic System for Translation (PHAST). Here we report on the methods used to help simulate a functioning system and gather user feedback.

Related Work

Evaluating systems in the field with target users is an important part of the user-centered design cycle, but this can be challenging for groupware [1]. While a few high-level frameworks have been proposed for groupware usability evaluations [2-4], there remains a shortage of field methods, particularly when the system cannot yet be deployed [1, 5]. Inspection and other discount methods are inexpensive and easy to use, as they do not require a full deployment and a lab with specialized equipment, but inspection may miss important information from users and the field [5, 6]. Scenarios are among the most useful methods for evaluating groupware systems because they can elicit feedback on contextual factors that may not otherwise be visible to the evaluator [7-9]. This is particularly important when the system has not yet been deployed. Gutwin and Greenberg's mechanics of collaboration (MoC) provide a way to map scenarios to a larger framework for evaluating a system's support for collaboration [2]. Our team tested if scenarios and MoC could be used to evaluate prototype groupware.

System Overview

We designed PHAST, a collaborative web-based translation tool, to facilitate the integration of MT into the typical translation workflow of public health employees (phastsystem.org) [10, 11]. In previous studies we found that few public health departments have sufficient bilingual staff with all the language expertise necessary to meet their translation needs, so PHAST is designed to draw from language expertise across the state [12]. Public health employees can upload documents for translation in a language their own department lacks, then PHAST notifies users from other health departments who have expertise in the target language that a new document has been added and is ready for post-editing. The latter health department's staff with that language skill can make corrections to the MT output generated by PHAST. In this way, health materials are translated and corrected collaboratively, then shared amongst participating health departments.

Methods

In order to thoroughly test the collaborative aspects of PHAST, we would have to deploy it to several health departments. In the absence of validated discount usability methods to evaluate such a prototype, we situated the tasks in collaborative scenarios and the MoC to gather feedback from target users on whether or not PHAST's current iteration adequately supports collaborative work for document translation.

We conducted usability testing with a total of 10 target users from two public health departments in western Washington State – 6 from one and 4 from the other, the most each department could volunteer. An introductory meeting was held to explain the study and show PHAST's main functions, but it was not detailed enough to be leading. Onsite testing took place using the live PHAST website on a laptop PC. We video recorded to capture the screen and audio. Each session had a moderator and note-taker; the moderator asked each participant to think aloud while completing tasks and asked probe questions when appropriate.

We used scenarios based on earlier user interviews to elicit feedback on PHAST's collaborative aspects by situating 10 tasks within simulated cooperation with another public health department, mimicking what would likely occur if the system were deployed (e.g., "Let's say a Department of Health employee has uploaded a document about Chickenpox in Spanish for translation and post-editing. You want to help by post-editing the translation. How would you do this?"). We collected data on effectiveness, efficiency, and satisfaction [2] with task success/failure; number and severity (on a scale of 1-5) of errors; as expected/not as expected statements; and perceived satisfaction (on a scale of 1-5, with 1 representing completing the task with no problems and stating it was as expected or better, and 5 meaning failure with complaints).

We used the MoC framework to evaluate how well PHAST supports collaboration. Using Atlas.ti, three team members blindly coded the videos with a codebook that included codes for each task, for each of the aforementioned indicators, and for each of the MoC mechanics (explicit communication, consequential communication, coordination of action, planning, monitoring, assistance, and protection). We merged our blindly coded Atlas.ti files and resolved redundant coding. We used Atlas.ti's code co-occurrence tool, plotting each of the MoC codes against the effectiveness (success/failure), efficiency (error count and severity), and satisfaction (as expected/not as expected and satisfaction 1-5) codes. The tool displays the degree to which two codes overlap, so we used this qualitatively as a way of finding the degree to which PHAST supports collaboration. We clustered the results into: needs some improvement, mixed, adequate, moderately high, and high. For example, the consequential communication code strongly overlapped with codes for high satisfaction, so PHAST's support for that mechanic is high. The table below summarizes the findings and visualizes the MoC evaluation model [2].

Mechanics of Collaboration	Effectiveness	Efficiency	Satisfac-tion
Explicit communication	Adequate	Adequate	Adequate
Consequential communication	Mixed	Mixed	High
Coordination of action	Needs some improvement	Needs some improvement	Moderately high
Monitoring	Needs some improvement	Needs some improvement	Moderately high
Assistance	Adequate	Adequate	Mixed
Protection	Needs some improvement	Mixed	Moderately high

Table 1: Mechanics of collaboration results summary [2]

Discussion

By creating scenarios to elicit feedback about the collaborative aspects of the system and mapping the MoC to functional tasks in PHAST, we were able to gather feedback about PHAST's groupware features without a deployment. Most of the MoC were coded in multiple tasks and system features, and one (planning) was too high-level to use, so the code co-occurrence tool was helpful for parsing the data. We found that the MoC findings extended and reframed the individual findings in a way that clearly highlighted the system's support for collaboration and where changes to PHAST would be most valuable.

Conclusion

We have reported on our experience evaluating an undeployed groupware prototype machine translation (MT) and post-editing system called PHAST. We produced high-level, qualitative results, demonstrating that combining scenarios with the MoC as a groupware

evaluation framework provided useful formative data for improving support for collaboration within PHAST prior to deployment. We suggest this as a method for evaluators of prototype groupware systems.

Acknowledgements

This study was funded by National Library of Medicine (NLM) grant #1R01LM010811-01. Its content is the sole responsibility of the authors and does not necessarily represent the view of the NLM. We would like to thank Adrian Laurenzi for his work in developing PHAST and the health department staff who have volunteered their time.

References

[1] Neale, D. C., Carroll, J. M., & Rosson, M. B. Evaluating computer-supported cooperative work: models and frameworks. 2004 ACM conference on Computer supported cooperative work, ACM (November 2004), 112-121.

[2] Gutwin, C., & Greenberg, S. (2000). The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. In Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE 2000. IEEE 9th International Workshops, IEEE (2000), 98-103.

[3] Damianos, L., Hirschman, L., Kozierok, R., Kurtz, J., Greenberg, A., Walls, K., ... & Scholtz, J. Evaluation for collaborative systems. ACM Computing Surveys (1999), 31(2es), 15.

[4] Mendes de Araujo, R., Santoro, F. M., & Borges, M. R. A conceptual framework for designing and conducting groupware evaluations (2004). International journal of computer applications in technology, 19(3), 139-150.

[5] Steves, M. P., Morse, E., Gutwin, C., & Greenberg, S. A comparison of usage evaluation and inspection methods for assessing groupware usability. 2001 International ACM SIGGROUP Conference on Supporting Group Work, ACM (September 2001), 125-134.

[6] Nielsen, J. (1994). Guerrilla HCI: Using discount usability engineering to penetrate the intimidation barrier. Cost-justifying usability, 245-272.

[7] Haynes, S. R., Purao, S., & Skattebo, A. L. Situating evaluation in scenarios of use. 2004 ACM conference on Computer supported cooperative work, ACM (November 2004), 92-101.

[8] Turner, A. M., Reeder, B., & Wallace, J. C. A resource management tool for public health continuity of operations during disasters. Disaster medicine and public health preparedness (2013), 7(02), 146-152.

[9] Rosson, M. B., & Carroll, J. M. (2009). Scenario based design. Human-computer interaction. Boca Raton, FL, 145-162.

[10] Laurenzi, A., Brownstein, M., Turner, A. M., Kientz, J. A., & Kirchhoff, K. A web-based collaborative translation management system for public health workers. CHI'13 Extended Abstracts on Human Factors in Computing Systems, ACM (April 2012), 511-516.

[11] Turner, A. M., Brownstein, M. K., Cole, K., Karasz, H., & Kirchhoff, K. Modeling workflow to design machine translation applications for public health practice. Journal of Biomedical Informatics (2014).

[12] Turner, A. M., Mandel, H., & Capurro, D. Local Health Department Translation Processes: Potential of Machine Translation Technologies to Help Meet Needs. AMIA Annual Symposium Proceedings, AMIA (2013), 1378.