Using Crowdsourcing in the Design of Context-Sensitive Help for Web Applications

Parmit K. Chilana, Amy J. Ko, and Jacob O. Wobbrock
The Information School | DUB Group
University of Washington
Seattle, WA 98195 USA
{pchilana, ajko, wobbrock}@uw.edu

ABSTRACT
Users are increasingly searching for information online to learn how to use applications and recover from unexpected application behaviors. Even though forums, wikis, and help documents may contain rich information about resolving a large number of software issues, these resources are usually scattered and disconnected from the context in which the user seeks help. In our research, we are exploring the design of a crowdsourced help tool that allows users to share knowledge of support issues within the context of the application.

Author Keywords
Context-sensitive help, embedded help, crowdsourcing

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Human Factors.

INTRODUCTION
Despite the availability of application-specific help on the web, there are two inherent problems for users searching for help. First, users struggle in finding help online because search engines require users to be precise in articulating the issue that they are facing. This is problematic given that users usually leave the context of what they are doing to go seek help and have to recall rather than recognize the subject of their issue when describing the problem. Secondly, the retrieval of relevant solutions depends on how well search queries provided by users match descriptions of technical writers or other users who may have reported similar problems in forums.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

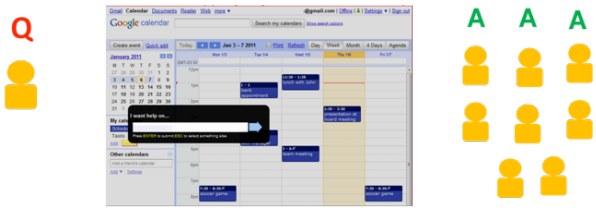

Figure 1: Users will be able to point to keywords and objects on the screen and explore relevant questions and solutions contributed by other users.

OUR APPROACH
In our research, we are inventing techniques that allow users to find help by pointing to visual referents, such as keywords and objects on the screen that are relevant to the subject of an application issue. Whereas users have infinite ways of describing the subject of their issue in natural language queries, our approach restricts the large space of queries to a small subset of queries that can be expressed using visual referents on the screen that are common among all users. Furthermore, the solutions that users see are crowdsourced from communities of users.

CURRENT PROGRESS
We have carried out preliminary studies to understand visual referent selection for help scenarios in real-world applications. The findings showed that when users seek help using visual referents, there are 3 possible outcomes: (1) users select a single referent, converging on keywords in labels, headings, titles, names of interactive elements (i.e., buttons), and other types of text within the interface that may be relevant to a given problem; (2) users select different referents within the same page view; or (3) users point to different manifestations of the same referent. We are currently expanding on these findings by exploring the structural and semantic relationships between visual referents in real-world applications and selections made by a large group of users.

We are developing prototypes that will be used to create a corpus of visual referents for real application scenarios. Once we have this corpus of data, we will devise metrics for determining similar attributes between referents and invent algorithms for matching similar referents. The next

focus will be on devising front-end techniques for browsing existing help attached to visual referents, and creating interfaces for asking new questions and authoring solutions to existing questions.

DEVELOPING A REPRESENTATIVE CORPUS

One challenge in creating a representative corpus is the number of selections that will be required to adequately mimic a community of online help requests. For example, in our preliminary investigations of the Facebook support forums, we have found several hundred goals and, for each, dozens of user interface contexts in which each goal is pursued. We hope to gather at least 50 selections for each goal context/pair, meaning that we will need on the order of hundreds of thousands of help selections. Obviously, soliciting such selections in a conventional face-to-face lab setting would be infeasible. Therefore, we are deploying these tasks using Amazon's Mechanical Turk (mTurk) platform. This service has been successfully used in several recent research experiments in HCI, but to our knowledge, we are the first to gather user interface selections using the mTurk platform.

Apart from technical challenges in creating customized tasks and deploying them through mTurk, the use of mTurk also poses some validity challenges. First and foremost, because users are not monitored by experimenters, there is little guarantee that the selections they provide to scenarios will come from a reasonable understanding of the scenario; participants could game the system, selecting arbitrary elements in order to receive payment. Therefore, we are incorporating two multiple-choice comprehension questions for each task, each with five options, requiring participants to successfully indicate their understanding of the scenario to receive payment (and limiting the chance of random responses to 1 in 25). Our pilot tests have shown that this pre-task comprehension test reinforces task-related understanding and helps minimize potential noise in the selection data. We are also collecting demographic information about Turkers, such as their native language, application expertise, task-specific expertise, and overall technical expertise as they could potentially impact our overall data. By assessing correlations between these factors and selections made by users, we will be able to better determine whether we need to develop other strategies for mitigating any bias in the mTurk data.

Once the corpus is created using selections from mTurk, we will identify a set of similarity principles and use them to devise a retrieval algorithm that maps new user help selections to existing selections. To evaluate the efficiency of our retrieval algorithm and whether or not users are able to find a solution to their problem by making on-screen selections, we will again use the mTurk platform and repeat the study using the same scenarios. However, for evaluation purposes, the focus will be on retrieving solutions and we will test on a different set of Turkers than those who participated in the corpus creation.

CONTRIBUTIONS

Our current research represents a new way of doing context-sensitive, smart help for web applications using crowdsourced solutions. This could have direct impact on improving the value of web applications by (1) decreasing user frustration by providing just-in time, contextual help in the language of users, (2) allowing new users to participate, customize, and contribute new feature ideas more efficiently, and (3) helping web developers make more informed design changes by aggregating large-scale data about users' help-seeking activities. The overall benefit will be an enhanced user experience for millions of users.

CROWDSOURCING AND THE FUTURE OF HCI

We believe that crowdsourcing will become an essential component of any research in HCI that tries to make inferences about human-behavior at a large scale or tries to introduce new tools or techniques for web-based interactions. We may begin to see parallel benefits in HCI practice as well—in the same way as usability testing has become a mainstay in software design, crowdsourcing-based evaluations may change the face of how software design decisions are made in the near future.

However, even though crowdsourcing platforms such as mTurk have opened up new avenues for research and practice, there is increased need for a set of best practices for using these platforms. Currently, individual researchers are not only engineering customized tasks, but also working towards mitigating potential threats to validity and insuring high quality of data collected through crowdsourcing platforms. While it is important to continue to share these experiences through research publications and workshops, perhaps in the future we may benefit from having a more defined core of generic principles that can be applied towards any research involving crowds.

AUTHOR BIO

Parmit Chilana is a PhD student at the Information School at the University of Washington. Her research interests span different aspects of usability, software support, and software development. In recent projects, she has looked at how users express unwanted software behaviors and how support professionals and developers diagnose and respond to user-reported software issues in open source and commercial contexts. She is currently exploring the design of contextual-help tools for web applications which leverage crowdsourced solutions from users. Parmit has also collaborated on various projects in the biomedical and health informatics field related to improving design of software used for scientific research. Parmit received her MS from the University of Illinois at Urbana-Champaign and BSc from Simon Fraser University, Canada. She was a recipient Facebook's inaugural PhD fellowship in 2010 and Canada's SSHRC doctoral fellowship in 2009.