Experiences in Conducting an Online Field Study of an Open-source, Extensible Software Platform

Leah Findlater[1], Jen Hawkins[2], Joanna McGrenere[1], David Modjeska[2]

[1]Department of Computer Science
University of British Columbia
Vancouver, Canada
{lkf, joanna}@cs.ubc.ca

[2]IBM Toronto Software Lab
8200 Warden Ave.
Markham, Canada
{jlhawkin, modjeska}@ca.ibm.com

Abstract. We identify several challenges in conducting a large-scale, online field study with 90 users of an open-source, integrated development environment. By reflecting on the benefits and challenges of specific methodological decisions and contextual constraints of this study, we hope to provide insight for other researchers designing similar types of studies, as well as to encourage discussion about how best to balance these trade-offs. In particular, we highlight the online nature of this study, and the impact of working with an open-source, extensible software application.

1 Introduction

Feature-rich software applications often provide more features than are used by an individual user [3,4]. Layered interfaces tailored to users' needs have been proposed to address this issue: commands are grouped into logical layers so the user can work in a simple layer with a core set of commands before enabling, as needed, more complex layers [5]. To understand the feasibility of creating a layered model for a feature-rich application, we conducted a field study to collect command usage of Eclipse Web Tools Platform (WTP), an open-source, integrated development environment (www.eclipse.org/webtools). Our goal was to cluster the usage data to understand how easily commands could be grouped into layers and to extract design lessons from those layers. Cluster analysis requires a large amount of data, so we focused on collecting usage log data rather than on more intensive qualitative techniques. We reflect on the main challenges in conducting this study, particularly its online nature and the experience of working with open-source, extensible software. Our aim is to provide insight for researchers designing similar studies, and to highlight issues for further discussion within the community.

2 Methodology and Data

Over an eight-month period we collected usage logs from 90 users; these logs ranged in length from minimal (less than a day) for 8 users to more than five months for several users. Combined, the logs contain over 1.6 million command events, including menu selections, toolbar selections, and keyboard shortcuts. Recruitment of users was done by advertising on the Eclipse WTP website, newsgroups, and user groups, and by announcements at Eclipse-focused conferences.

We modified and extended the Mylyn Monitor (www.eclipse.org/mylyn), which logs events, including menu, toolbar and keyboard shortcut commands. These events are stripped of identifying information and logged to a local file, which the logger prompts the user to upload to an HTTP server once a week. Participants also completed a brief background questionnaire when they first installed the logger.

Our goal was to distill each user's data to a list of commands paired with usage frequencies, and then to cluster the commands by user. Unfortunately, we encountered two unanticipated challenges: (1) we had expected to recruit several hundred users through our online system, but only recruited 90; (2) as discussed in the next section, the extensible, open-source nature of the software application introduced ambiguities into the data, and resulted in a much larger command set than expected.

3 Challenges Encountered

3.1 Open-source, extensible project resulted in highly variable data

Many open-source projects are also extensible, which proved a major challenge in collecting quality usage data from Eclipse WTP. Users can install any number of third-party plug-ins on top of the base platform, and even create their own plug-ins. Analyzing which additional plug-ins users had installed can provide a rich basis for understanding customization patterns, but the lack of conformance to coding standards in some of these plug-ins greatly hindered our ability to collect good quality log data. Third-party plug-in developers are encouraged, but not forced, to follow standard function naming conventions, and the result is that many plug-ins do not conform. Even the base platform does not always follow standard naming conventions, which led to many ambiguities in the data.

On the positive side, open-source software also provides benefits. For example, participants wanted the logging module to be open-source; this anticipated transparency of researchers' goals can increase trust and participation. A sense of community, responsibility, and shared altruistic goals within a project's participants may also encourage higher levels of participation in usability studies. This community sense might be easier to realize if the researcher is also a member of that community, which was not the case for our study, a fact that might have impacted recruitment efforts focused on building word-of-mouth interest (e.g., conferences and newsgroups). Instead, it was more effective to consistently advertise on the high-traffic Eclipse WTP homepage.

Proposed solution:

- **Enforce consistent naming conventions.** From a software engineering standpoint, our difficulties highlight the broad need to enforce consistent naming conventions in development projects, particularly those with an extensible application. This finding supports conclusions of Murphy, Kersten, and Findlater, who conducted a similar field study with Eclipse [1].
- **Develop an experimental version of the application.** To address the issue of ambiguity for the purposes of a single study, one possibility is to create an experimental version of the application, where these problems are fixed in the source code. Beyond the effort required to do this, however, the drawbacks are that participants would have to install an entirely new version of the application,

rather than only a small plug-in, and that future product patches might not be compatible with the experimental version.

3.2 Online, large-scale participation precluded qualitative data collection

In a field study, the balance between quantitative data (of which a large amount is often needed) and qualitative data (which is usually more labour-intensive to collect) depends on the goals of the particular study [2]. Since our analysis required a large amount of quantitative data, we chose to conduct the study fully online, allowing us to take advantage of online recruitment and an automatic uploading mechanism to involve a large number of participants. As a result, it would have been infeasible to use observational or interview methods to collect additional qualitative data from each participant because of the number of participants and their geographic distribution (across North America and Europe). We also designed the study to minimize the impact of the logging tool on the participant's work, and to require minimal effort on his or her behalf. As such, we did not ask participants to provide regular self-reported data, such as task boundaries and descriptions, in addition to the logged usage. Consequently, we collected very little qualitative data. Since our quantitative data was much noisier than expected, richer qualitative data would have been invaluable in providing another level of abstraction at which to interpret usage patterns.

Proposed solutions:

- **Collect qualitative data from a representative subset of users.** While it might not always be feasible to collect observational or interview data from every user, it might be useful to do so from a representative subset of users. In our study, geographical constraints would have presented a challenge in collecting data from representative users, so this would be a more practical approach for a study where users are more easily accessible.
- **Collect lightweight qualitative data from all users.** In retrospect, it would have been useful to ask users to delineate major tasks and to provide short descriptions of their work, even if it meant interrupting them occasionally. This could have been done through automatic or user-invokable pop-ups to collect diary-style entries, user comments, and breaks between tasks. The task information would have been useful for interpreting lower-level patterns of command usage, as well as for filtering out some of the ambiguity introduced by third-party plug-ins.

3.3 Pilot testing is difficult with exploratory usage analysis

As with many data mining techniques, our planned analysis was exploratory (i.e., we did not know whether command usage would fall into distinct clusters or not). It was therefore difficult to use a pilot period to determine whether the data we were collecting was of high quality. Preliminary analysis of the first 22 users' data showed evidence of use of approximately 800 unique commands, and the distribution of the most frequently-used commands was as expected. However, 22 users were too few to support clustering analysis. Moreover, we did not have a mechanism to predict whether the data would ultimately be appropriate for such treatment (i.e., after we had recruited more users). Unfortunately, collecting more data from more users only increased the amount of noise in the data, recording use of approximately 2000 unique commands.

Proposed solution:

- **Conduct a small, controlled pre-study.** Since data mining techniques require a large amount of data, it may have been more effective to first refine hypotheses through a small, more controlled pre-study before conducting a much larger field study. For example, since our goal was to cluster commands based on users, we could have predetermined groups of users who perform distinctly different tasks (e.g., mainly Java™ developers or mainly Web services developers) and chosen only a few users from each group to participate in the pre-study. By collecting quantitative and qualitative data and comparing differences in high-level tasks and command usage between the groups, we may have been able to build a preliminary model of command clusters to validate in the larger usage study.

4 Conclusion

We have identified methodological decisions and contextual constraints that negatively impacted the quality of data collected in a field study with 90 users of an open-source, extensible software application. In hindsight, a few changes to the study may have increased its chance of success: improved recruitment of users through active involvement of at least one member of the open-source project (rather than an outside researcher), and more aggressive community outreach; inclusion of a lightweight qualitative data collection mechanism to provide additional context for interpreting the usage log data, even at the expense of interrupting the user; development of an experimental version of the application to fix ambiguities in command naming, thus reducing noise in the data.

Acknowledgements. We thank Jin Li, Lawrence Mandel, Arthur Ryman, and Craig Salter of the IBM Toronto Lab for their input and help in running this study. We also thank IBM Centers for Advanced Studies and NSERC for funding.

Note. The opinions expressed are those of the authors and do not necessarily represent the opinions of IBM. IBM is a registered trademark of International Business Machines Corporation in the United States, other countries, or both. Java is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both. Other company, product or service names may be trademarks or service marks of others.

References

1. Murphy, G., Kersten, M., and Findlater, L. (2006). How are Java software developers using the Eclipse IDE? IEEE Software, 23(4): 76-83.
2. Lethbridge, T.C., Sim, S.E., Singer, J. 2005. Studying software engineers: Data collection techniques for software field studies. Empirical Software Engineering, 10, 311-341.
3. Linton, F., Joy, D., Schaefer, H.-P., & Charron, A. (2000). Owl: A recommender system for organization-wide learning. Educational Technology & Society, 3(1), 62-76.
4. McGrenere, J., and Moore, G. (2000). Are we all in the same "bloat"? Proc. of GI 2000, 187-196.
5. Shneiderman, B. (2003). Promoting universal usability with multi-layer interface design. Proc. of the 2003 Conference on Universal Usability, 1–8.