Mining Usability Information from Log Files: A Multi-Pronged Approach

Michelle Potts Steves & Emile Morse
National Institute of Standards & Technology (NIST)
100 Bureau Drive
Gaithersburg, MD 20899
(301) 975-8239
{msteves, emile.morse}@nist.gov

This position paper has been prepared for a workshop concerned with issues and experiences surrounding logged data for on-line communities. We briefly describe relevant elements of a field study of a group of welding engineers using a collaborative system conducted at NIST. The log data captured during the field study is sequential, as it is ordered by time. We believe this ordering will be important to preserve during analysis. Therefore, we present our experiences and issues encountered during the analysis of our data in the context of a categorization of techniques for analysis of sequential data by Hilbert and Redmiles [2].

Group Description

The core welding research team is composed of five people with six roles divided among them. Four of the five team members are located at the same physical facility where the actual welding takes place, the welding testbed. Additionally, guest researchers may participate in experiments; they are not necessarily physically co-located with the core group. Each of the participants in the welding collaboratory has access to a Teamwave Workplace[1] (TW) client from his desktop computer as well as computers in the welding testbed. TW is a shared, room-based collaborative system with a WYSIWIS (what you see is what I see) whiteboard backdrop. "Rooms" in TW provide boundaries for data groupings and user interactions, and a metaphor for easing the transition in groupware [1]. Doorways provide portals to other rooms. Data organization within rooms is configurable by its occupants in how they organize various tools housing their data, documents, and graphics. The TW system provides for synchronous and asynchronous user interactions, but importantly these interactions are in the context of relevant data.

The work in this experiment was characterized by several full group meetings (for planning and coordination) interspersed with periods of individual activity (asychronous work) and smaller coordination meetings of two or three team members around the "hand-off" of output from a task used as input for another task.

Collected Data

The native version of TW produces a server-based log-file that contains information about the identity of users entering the distributed application, the identity of the rooms through which users navigate, file uploads, and message passing between users. This set of interactions was deemed too rudimentary for capturing the type of data needed for usability analysis. Since the source code for the application was not available, we contracted with the vendor to instrument the software for additional logging. Issues surfaced during the subsequent negotiation that are pertinent to the workshop focus. For example, the usability analysts tended to want more interactions logged than the vendor was willing to incorporate, primarily because of performance implications. Logging at a fine granularity would potentially adversely impact system performance. The outcome of the negotiation between the vendor and the analysts was the creation of a second log that contained additional information such as the content of chat sessions; the combination of user ID, room ID and tool ID when a new tool was created; and, the combination of user ID, tool ID and appropriate annotations when a tool or its content was modified (e.g., message added to a Message Board or content of a PostIt™ modified).

The information collected in the server logs was supplemented by questionnaires and interviews. The latter data is clearly qualitative, while log data has a more quantitative nature. Over the course of the study, which lasted 2.5 months, 4015 interactions were appended to the custom log file and 3605 events were created in the base server log.

We found two kinds of issues emerging during our data analysis: data issues and analysis-technique issues. The data issues centered around granularity and not knowing what to log before the analysis started. Before starting the experiment, we expected to use exploratory data analysis to zero in on interesting user interactions and behaviors.

[1] Any commercial product identified in this document is for the purpose of describing a collaborative software environment. This identification does not imply any recommendation or endorsement by NIST.

This strategy worked fairly well. However, during the analysis phase, we found that there were some interesting tracks of investigation that could not be fully explored because of insufficient data granularities logged to support deep levels of investigation. The analysis-technique issues centered around being able to abstract the logged data to meaningful levels, putting the data in context (with the users' goals and experiences), and matching the logged data (and its abstractions) to qualitative evaluation data (e.g., user satisfaction questionnaire data).

Data issues background

User interface (UI) events are generated in the normal operation of window-based user interface systems. UI events include mouse clicks, button pushes, data input in selected input fields, and so on. The capture of UI event data is generally possible since all window-based systems produce these events. Additionally, because UI events correspond to user interactions with the application, they are regarded as a potentially fruitful source of information with repect to application usage and usability. However, because of the low-level nature of UI events, the logging of these events produces large quantities of data, and that data may be difficult to abstract to useful levels during analysis. [2]

Sanderson & Fisher [5] in their description of exploratory sequential data analysis (ESDA) discuss the range of event types in human-human interactions. These authors note that the granularity of capturable events range from eye movements and gestures, which change many times per second, to turn-taking and topic changing, which may occur over intervals ranging from several seconds to about 10 minutes. Meetings last between a minute and several hours, while projects may extend for months or even years. The enormous variability in the granularity and duration of these events from a user-centric perspective lead to a consideration of the parallel set of computer-centric, UI events that are possible to capture.

Hilbert & Redmiles [2] have studied how user interface events can be categorized to yield high-level usability data. These authors describe the types of abstractions in user interactions in a hierarchy whose levels are physical events, input device events, user interface events, abstract interaction events (e.g., providing values in input fields), domain/task related events, and goal/problem-related events. Attention to these many levels at one time is as difficult as addressing all the levels of human-human interaction.

Regardless of whether a human- or computer-centric yardstick is used to measure the scale of the problem, it is clear that the data set created from interactions of a user with a computer system is likely to be very large. In addition, multi-user, collaboration systems will necessarily experience a combinatorial explosion due to potential and actual interactions among the multiple users.

Data issues to explore

1. Logging of events generally requires access to the application code. This is not always possible. When it is, there is an important issue related to performance of the application vs. logging of events. Logging huge quantities of events can slow application performance, which in turn can affect usability and user satisfaction with the application. Therefore, some discretion is warranted. Additionally, logging huge quantities of events can generate very large log files; this may be a concern in some situations.
2. What should be logged? Or, what granularity of events should be logged with respect to the anticipated analysis? This is a very hard question. It is easier to answer the question of what should have been logged.
3. What are the differences between what can be logged in window-based applications vs. what can be logged in browser-based applications?
4. Is it possible to generalize and standardize log-file formats so that common tools can help analyze them? One technology that suggests itself is eXtensible Markup Language (XML) [6] for tagging log data so other software tools can interpret it. Currently, there are few tools specifically for analyzing log data files. There are very general tools such as spreadsheets and statistical analysis packages, and there are very application-specific tools, e.g., the CollabLogger [4] for visualizing a specific log-file format. If log-file formats could be generalized and standardized, then a set of tools could be produced for the analyst's arsenal. This notion was explored last June at the WETICE workshop, Evaluating Collaborative Enterprises [3].

Analysis techniques, issues and our experiences

Hilbert and Redmiles group and characterize techniques for analyzing sequential data, focusing on those techniques that have been applied to domain of Human-Computer Interaction (HCI) [2]. In the following section, we give Hilbert and Redmiles' descriptions and our related experiences and issues for each technique category.

*"**Techniques for synchronization and searching.** These techniques allow user interface events to be synchronized and cross-indexed with other sources of data such as video recordings and coded observation logs. This allows searches in one medium to locate supplementary information in others. In some ways, this is the simplest (i.e., most mechanical) technique for exploiting user interface events in usability evaluation. However, it is quite powerful."* [2]

We are trying to use a technique that falls under this category, in that we are in the process of constructing a timeline of (significant) events during the course of our 2.5 month experiment. The data is coming from various inputs (data logs, user recounts of events, and analysts' observations). We are attempting to use HyperText Markup Language (HTML) to present the data, but are running into problems with being able to appropriately portray different levels of event abstraction.

*"**Techniques for transforming event streams.** Transformation involves selecting, abstracting, and recoding event streams to facilitate human and automated analysis (including counts, summary statistics, pattern detection, comparison, and characterization)."* [2]

We are using the CollabLogger visualization tool to help us select and abstract various event sequences of interest.

*"**Techniques for performing counts and summary statistics**."* [2]

The CollabLogger has some count and summary statistic reporting facilities that are helpful, but we are also using spreadsheet and database management tools.

*"**Techniques for detecting sequences.** These techniques allow investigators to identify occurrences of concrete or abstractly defined "target" sequences with "source" sequences of event that may indicate potential usability issues."* [2]

We are using the CollabLogger to explore visually what other events happen near target sequences. We are also using a spreadsheet and database management tool to detect the timeframes for selected sequences. The necessity of using other tools in this fashion indicates a problem with the CollabLogger's (and, indeed with many visualization tools') ability to deal appropriately with potentially large amounts of white space.

*"**Techniques for comparing sequences.** […] In all cases, the purpose is to compare actual sequences of events against some model or trace of "ideal" or expected sequences to identify potential usability issues."* [2]

We have also performed expert walkthroughs (a usability technique) and have begun to mine the logs of the two types of approaches to measure the steps performed by the participants in the welding scenario with that of the experts. The synergistic application of database and visual methods is being used for this analysis.

*"**Visualization techniques.** These techniques present the results of transformations and analyses in forms allowing humans to exploit their innate visual analysis capabilities to interpret results."* [2]

A complete description of the CollabLogger visual tool can be found in [4].

*"**Integrated evaluation support.** Evaluation environments that facilitate flexible composition of various transformation, analysis, and visualization capabilities provide integrated support."* [2]

The CollabLogger is a rudimentary example of an integrated evaluation support environment, as it does provide some basic summary statistics. However, we have no plans to significantly augment that aspect of the system at this time.

In addition to issues that fit in the Redmiles framework, there are both related and unrelated issues that are raised by our analysis. Related issues include:granularity of logged events, how to put keystrokes in a context, how to compare quantitative data with qualitative data. Unrelated issues address things like the the ethics of logging chat.

References

[1] Greenberg, S. and M. Roseman. 1998. Using a Room Metaphor to Ease Transitions in Groupware. Research Report 98/611/02, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada.

[2] Hilbert, D.M. and D.F. Redmiles. 1999. Extracting Usability Information from User Interface Events. Technical Report UCI-ICS-99-40, Department of Information and Computer Science, University of California, Irvine.

[3] Knutilla, A., M. Steves and R. Allen. 2000. Workshop on Evaluating Collaborative Enterprises - Workshop Report. Proceedings of the IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 79-85.

[4] Morse, E. and M. Steves. 2000. CollabLogger: A Tool for Visualizing Groups at Work. Proceedings of the IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp. 104-109.

[5] Sanderson, P.M. and C. Fisher. 1994. Exploratory Sequential Data Analysis: Foundations Special Issue on Exploratory Sequential Data Analysis. Human-Computer Interaction. 9:251-317.

[6] XML: http://www.w3.org/XML/