CollabLogger: A Tool for Visualizing Groups At Work

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

Abstract

The CollabLogger is a visual tool that has been developed to support usability analyses of human-computer interaction in a team environment. The participants in the computer-mediated activity were engaged in a small-scale manufacturing testbed project. Interactions of the group were mediated by Teamwave Workplace[1] and the members performed both synchronous and asynchronous activities depending on their availability, project requirements, and due to chance meetings in the collaborative space. The software was instrumented to log users' interactions with the system and each other. The CollabLogger addresses the problem of helping investigators analyze the volumes of log data that groupware tools can generate. Visual tools are powerful when large amounts of diverse data present themselves. The place-based collaboration environment offered by Teamwave Workplace provided a level of organization that allowed us to create a visual interface with which to perform exploratory sequential data analysis.

Keywords: collaboration environments, software usability, visualization, exploratory sequential data analysis

Introduction

The current state of computing environments has changed remarkably from the days of single-user applications to one in which data, applications and users are distributed and interconnected. Increasingly, organizations are creating virtual teams that may be not only working at different locations but might also work asynchronously. At other times these same users may be collocated, working synchronously either by plan or by accident. Computer-mediated workspaces can be populated with a diverse assortment of objects; the states of all these components are subject to change depending on a myriad of actions by the users and the system. Although groupware systems can be instrumented to capture system and user actions, it is clear that the types of interactions in collaborative systems are more complex than in single-user applications, if only due to the multiple users.

Evaluation of human-computer interaction is the purview of usability professionals. They need arsenals of appropriate tools to perform the functions of determining whether an interface is supporting the goals and objectives of the system. The extra complexity of groupware systems involves detection and processing of not only human-computer interactions but also human-human interactions. A powerful tool for performing complicated data analysis is provided by the use of visualization.

Subsequent sections of this paper describe 1) previous, related work in the areas of exploratory sequential data analysis, data logging granularity issues, time-based visualizations, 2) the scenario which led to our development of the CollabLogger, and 3) the logging tool itself. Finally some conclusions and directions for future work will be discussed

Background and Related Studies

Pertinent literature related to the current topic comes from two main areas:

- What data can be captured vs. what data is useful to capture in time-based situations, and
- Other studies that have applied a visual approach to analyzing time-based data.

[1] Teamwave Workplace [1]is a commercial product identified in this document for the purpose of describing a collaborative software environment. This identification does not imply any recommendation or endorsement by the National Institute of Standards and Technologies.

Proceedings of the IEEE 9th International Workshops on Enabling Technologies:
Infrastructure for Collaborative Enterprises (WET ICE'00)

Data issues

The work of Sanderson & Fisher [2] on exploratory sequential data analysis (ESDA) discusses the range of event types in human-human interaction. These authors note that the granularity of capturable events range from eye movements and gestures which change many times per second, while turn-taking and topic changing 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 events that are possible to capture.

Indeed, the answer to this question can be found in the work of Hilbert & Redmiles [3]. They 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 (UI) events, abstract interaction events (e.g., providing values in input fields), domain/task related events and goal/problem-related events. Attention to all these levels at one time is as difficult as addressing all the levels of human-human interaction. They suggest that visualization is useful for presenting the results of transformations and other data reduction techniques to leverage the human visual abilities to detect relationships among events.

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 between the multiple users. Such data-rich environments are obvious targets for visual solutions [4].

Time-based visualizations

LifeLines was developed by Plaisant, Milash, Rose, Widoff & Shneiderman [5] as a mechanism for displaying personal histories. They describe medical and juvenile justice scenarios implemented with the interface. Overview, zooming, and filtering are all facilitated in the LifeLines display. The data comprises information about single individuals and it is not apparent how the interface could be extended to cope with data from collaborative settings with many individuals.

Morse & Spring [6] presented a set of visualizations that were based on data gathered using the CASCADE (Computer Augmented Support for Collaborative Authoring and Document Editing) system. Although the environment was collaborative in nature, the data in the visualizations that they describe were filtered by user before rendering. This filtering leads to a situation in which relationships among team members can not be detected.

The Multi-Modal Logger was developed at MITRE [7] to record, retrieve, annotate and visualize data from a variety of sources, including audio and video. The visualization tool that is part of the MML toolkit presents the usability engineer with a timeline on the x-axis and a set of objects along the y-axis. Objects may represent users, audio logs, collaborative tool invocations or other significant components of the team environment. Filtering the data to select only the users yields a display that indicates when each user was active (i.e., creating loggable events) but makes it impossible to determine what the users were doing. On the other hand, if the display is not filtered then inferences about users' activities are possible but are made difficult due to the potentially large number of object (i.e., timelines).

Kimber et al [8] have presented a time-based visualization that handles data derived from audio sources. Their system is capable of detecting boundaries between multiple speakers and other sounds. Their graphical display shows the identities of the speakers and attributions for other sounds (e.g., laughter). It also uses a sliding marker to indicate the current position in the playback of the audio; the marker can be dragged to wind the media to a particular area of interest. In addition, the interface has mechanisms for overview and zoom.

The literature related to data classification leads us to believe that it is important to consider that log data for collaborative activities must be gathered at sufficient granularity to support all possible uses that might be made of the data but, more importantly, that mechanisms need to built into the interface to the data to accomplish dynamic filtering of the data. The work on visualizations validates this approach to the treatment of time-based data, while showing that none of the interfaces developed thus far has provided a full answer to the problems of handling multi-user, interaction data.

Manufacturing Collaboratory Scenario

The CollabLogger was developed to support usability analyses of human-computer interactions in a manufacturing research project at NIST involving an automated gas-metal robotic welding testbed. NIST welding researchers are a geographically dispersed team, working to define interface standards between welding work cell components, controllers and power supplies. To achieve this, a functioning welding testbed has been implemented for testing the interfaces between components, equipment and power supplies. Analysis of

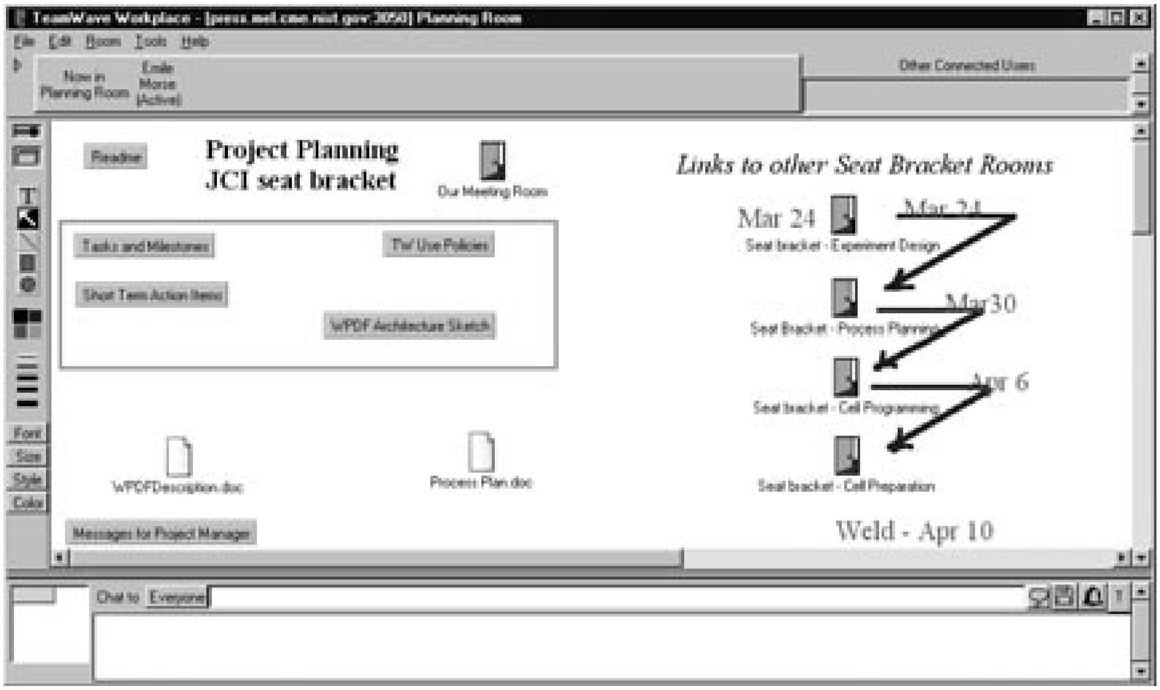

Figure 1: Teamwave Workplace environment for welding project

welds is performed to verify effective operation of interfaces, etc. [9]

The core welding research team is comprised of five people with six roles divided among them and additional guest researchers, who may be distantly located from the physical location of the welding testbed itself. Typically, a welding experiment can be characterized by the following set of tasks:

- Define the experiment (select experimental or developed interface standards and equipment, select the part to be welded -- e.g., automotive bracket, ship plate, select types of welds to be performed, etc.).
- Conduct a synchronous meeting to establish an experiment timeline and deadlines.
- Perform relatively sequential and primarily solitary tasks to set-up the experiment, e.g., procure and prepare the materials, program the robot, test the program using simulation, etc.
- Perform the initial weld(s), with most team members present, some virtually.
- Review the weld data; this is typically performed synchronously by the team if no difficult problem welds are detected. If difficult problem welds occur, typically the welding engineer reviews the weld data asynchronously to analyze the problem and report his finding back to the group.
- Iterate performing welds, varying experiment parameters as defined in the experiment plan until weld results indicate success.

This set of tasks is representative of how many groups work, i.e., in a variety of ways to best achieve their work objective(s). [10]

The CSCW Environment

Each of the participants in the welding collaboratory has access to a Teamwave Workplace[2] client from his desktop computer as well as computers in the welding testbed. Teamwave Workplace (Figure 1) is a shared, room-based collaborative system with a WYSIWIS (what you see is what I see) whiteboard backdrop. Rooms provide boundaries for data groupings and user interactions and a metaphor for easing the transition in groupware [11]. Doorways provide portals to other rooms. Data organization within rooms is configurable by its occupants in how they organize various tools housing their data, such as file viewers, PostIt™ notes, and message boards. FileHolders and ImageHolders are used for linking documents and graphics that may have been created outside of Teamwave Workplace. The system provides for synchronous and asynchronous user interactions, but importantly,

[2] Teamwave Workplace [1] is a commercial product identified in this document for the purpose of describing a collaborative software environment. This identification does not imply any recommendation or endorsement by the National Institute of Standards and Technologies.

Proceedings of the IEEE 9th International Workshops on Enabling Technologies:
Infrastructure for Collaborative Enterprises (WET ICE'00)

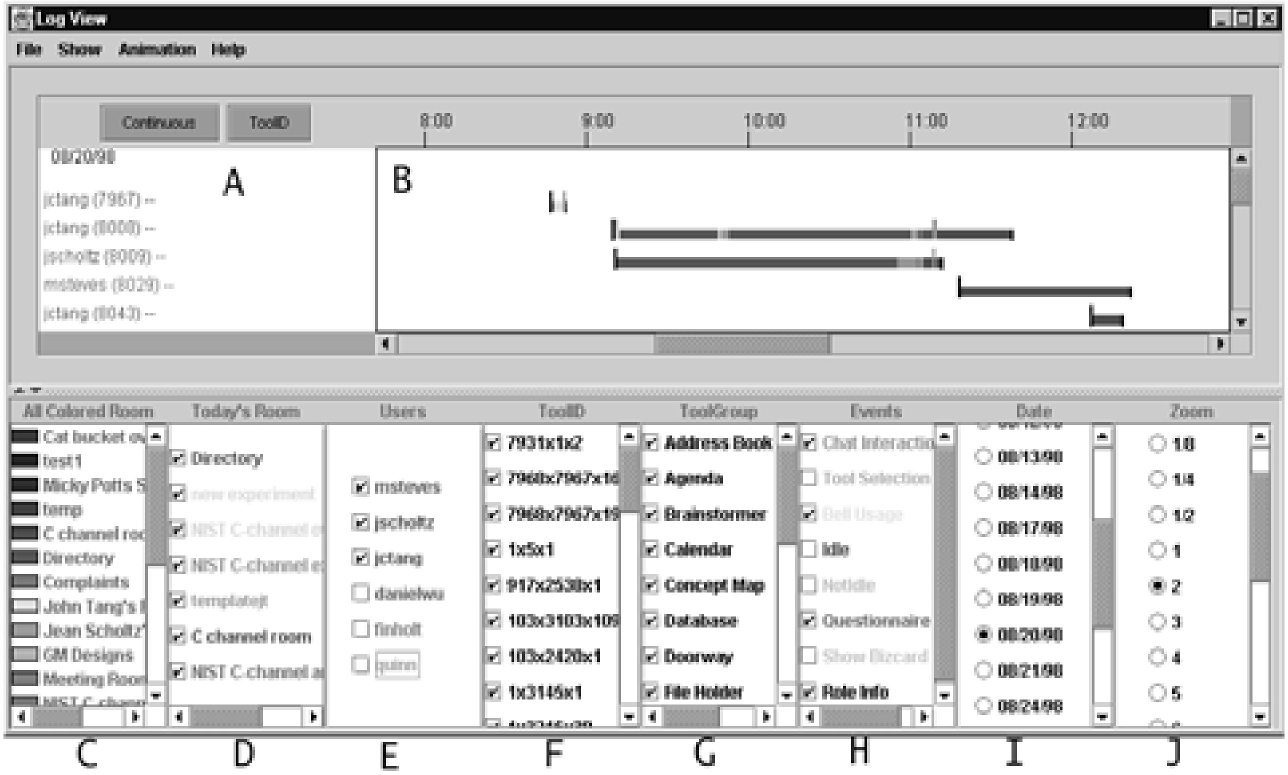

Figure 2: CollabLogger using User ID mode

Figure 3: CollabLogger using Tool ID mode

Proceedings of the IEEE 9th International Workshops on Enabling Technologies:
Infrastructure for Collaborative Enterprises (WET ICE'00)

these interactions are in the context of relevant data [12].

The version of TeamWAVE used in this project has been instrumented to capture information about users, TeamWAVE tools, and rooms. Each event is time-stamped by the server and keyed by user.

CollabLogger Description and Functions

The CollabLogger application (Figures 2 and 3) is written in Java 1.2. The menu bar provides access to specialized functions such as replay and symbol overlay. The leftmost panel (A) provides labels for the timelines shown in the main panel of the display. These labels can be toggled between a user-centric view (Figure 2) and a tool-centric one (Figure 3). The figures show log data zoomed in to a 4-hour time slice. The toggle button for user/tool mode is located immediately above the label list. The other button is a toggle that allows an overview of the entire log or a view of a single day.

The lower portion of the Logger window contains the controls by which the analyst interacts with the display. Checkboxes provide a mechanism to show or hide data in a particular view. Each room is color-coded and the panel C displays the mapping for all rooms in the log regardless of whether the current view has information about a particular room. The panel D shows the list of rooms that were occupied during the current day. Panel E allows selection of a subset of users. Information in the panels F and G is related to tool ID. Each instance of a tool is associated with a unique ID and these are displayed in panel F. Panel G allows hiding of tool information at the level of type of tool. For instance, the analyst may want to view all invocations of the 'Address Book' tool only. Events are shown in panel H and the user can delimit the set of markers by checking boxes next to the event type. Panel I allows selection among time segments based on days found in the current master log. Panel J provides a crude zoom tool.

The main display panel (B) of Figure 2 shows that 5 sessions occurred during the time period under investigation. The bars show which room(s) each user occupied during his/her session. Tick marks colored to map to an event type are overlaid on these bars. The particular session shown here reveals that more than one worker was present in the collaborative environment during most of the time. It also shows that these users engaged in multiple chat sessions. By filtering the data with the controls, it is possible to explore the information space. Inferences can be drawn and hypotheses can be formed that might be subjected to statistical treatment. Usability analysts working on collaborative data sets might be interested in answering the following kinds of questions:

- How often do multiple users occupy the space simultaneously?
- Which rooms are the most frequently visited?
- Are there patterns in the paths that users take within a set of rooms?
- Do particular users appear to be leading the activities?

Although the answers to the above questions could certainly be answered definitively using a statistical approach, visualizations allow the formulation of the questions themselves. Detecting patterns, outliers, densities, and gaps are innate abilities of the human visual system.

Figure 3 gives a different perspective of the data; it shows a list of tool invocations relative to time. An analyst using CollabLogger could manipulate the display by filtering and zooming to investigate:

- Are there combinations of tools that are often used in conjunction with one another?
- What tools are used in a room during joint occupancy vs. single occupancy?
- Are doorways an effective navigation tool? (# of hops, etc.)
- Which communication tools are used the most?

Once again the interface serves as a way to rapidly decide how best to formulate relevant hypotheses.

Summary and Future Directions

Usability analyses of collaborative environments are difficult to perform partly due to the potentially complex interactions between the users of the groupware. Users interact not only with each other but also with information objects and tools provided by the environment. The CollabLogger has been useful in allowing usability professionals to detect subtle interactions, which can then be subjected to more quantitative methods of analysis. One example of a use for the CollabLogger is in characterizing asynchronous interactions. That is, how can you tell when people have created something in a room with the intent to communicate to others later on? The onset of a chat or of desktop conferencing is easily flagged as a synchronous episode, but problems arise when trying to make inferences about the contributions of asynchronous activity to the overall workflow. The logging tool supports detection of trails of tool usage that might prove to be sufficient to infer asynchronous communication.

Use of the visual logging tool has raised a few usability issues on its own. Since the log records cover large time spans, there needs to be a more intuitive way to access the parts of the record that correspond to user activity. Other navigational problems are related to 1) the excessive amount of scrolling that is used in the controls and the main display, and 2) the use of a discrete rather than continuous zoom.

The work on CollabLogger has to this point depended on the fact that the TeamWave Workplace version that was used for these studies was instrumented to provide the kinds of information that were deemed to be useful in usability analyses. Future versions of the logger will need to be decoupled from the underlying collaboration environment in order to allow its application across a broader range of applications.

Acknowledgements

Many thanks to Dr. Ruijun Wu for succeeding in mapping the ideas of a large group of investigators into the application now known as CollabLogger.

References

[1] Roseman, M. 2000. http://www.teamwave.com/

[2] 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

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

[4] Tufte, E.R. 1990. Envisioning Information. Graphics Press, Cheshire, CT

[5] Plaisant, C., B. Milash, A. Rose, S. Widoff, and B. Shneiderman. 1996. LifeLines: Visualizing Personal Histories. CHI 96, 221-227.

[6] Morse, E.L. and M.B. Spring. 1996. Visualizations that Support Group Work. IEEE Symposium on Visual Languages, Boulder, CO, September 3-6, 1996.

[7] Mitre Multi-Modal Logger. 1997. http://www.mitre.org/technology/logger/

[8] Kimber, D.G., L.D. Wilcox, F.R. Chen and T.P. Moran. 1995. Speaker Segmentation for Browsing Recorded Audio, Proceedings of CHI '95, 212-213.

[9] Rippey, W. and J. Falco. 1997. The NIST Automated Arc Welding Testbed, Proceedings of the Seventh International Conference on Computer Technology in Welding.

[10] Gutwin, C. and S. Greenberg. 1998. Design for Individuals, Design for Groups: Tradeoffs Between Power and Workspace Awareness. CSCW 98, 207-216

[11] 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.

[12] Steves, M. and A. Knutilla. 1999. Collaboration Technologies for Global Manufacturing, Proceedings of the ASME International Mechanical Engineering Congress and Exposition (IMECE): Symposium on Manufacturing Logistics in a Global Economy.