Visualization for Casual Debugging and System Awareness in a Ubiquitous Computing Environment

Meredith Ringel Morris
Stanford University
353 Serra Mall
Stanford, CA 94305 USA
merrie@cs.stanford.edu

ABSTRACT

We describe a visualization of the status of information flow in a ubiquitous computing setting. We explain the semantics of the visualization, and discuss how this display has assisted novice users of our ubicomp environment with debugging and system understanding.

Keywords
Visualization, debugging, ubiquitous computing.

INTRODUCTION

The iRoom [5] is a ubiquitous computing environment that allows interactions between various devices, including large, wall-sized displays, standard PCs, laptops, and PDAs. A central tuple-space server, the Event Heap [4], coordinates communication between these disparate devices. Each machine posts messages (events) to the Event Heap server, and also checks the server for messages directed to it. When there is a technical breakdown in such a complex, multi-machine environment, it is difficult to pinpoint the cause of the problem – was the problem with the machine that sent the message, with the server receiving the tuples, or with the intended message recipient? We have developed a visualization of the status of the Event Heap server, which has improved system awareness and has been used for debugging by casual users of our ubicomp space.

Software visualization techniques, such as those described in [1] and [2], have been shown to assist in program comprehension and debugging; however, the target user of most software visualizations is the person who has written the code for the program being visualized. In a ubiquitous computing scenario, where a large number of computers affect users' environments, providing transparency of system status can be beneficial. In such situations, it may be desirable that end-users of the ubicomp space, rather than the original system programmers, be able to observe and, if necessary, correct the state of the system. Thus, the target audience of the Event Heap Visualizer is the users, rather than the developers, of the iRoom.

VISUALIZATION

The Event Heap Visualizer is an abstract visualization that displays the current state of the iRoom's Event Heap (Figure 1). The communicative aspects of the visualization are:

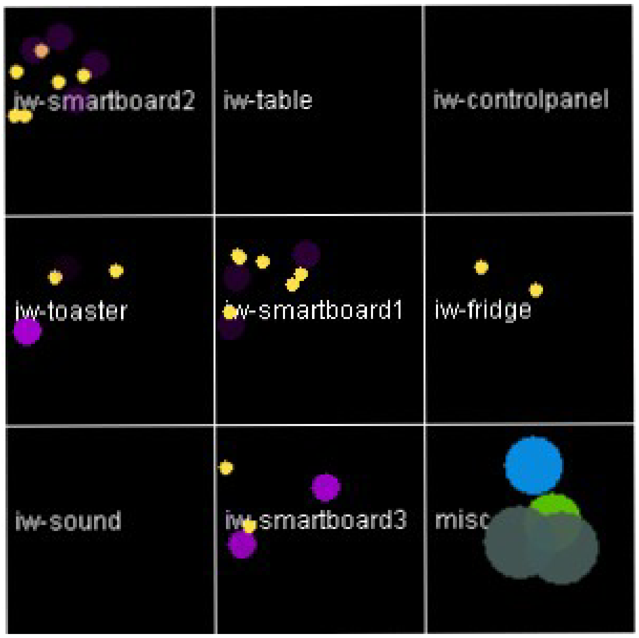

Figure 1. The Event Heap Visualizer uses the color, size, transparency, and position of abstract shapes to convey the state of the iRoom's tuple-space server, providing non-experts with awareness of system status and debugging insight.

- **Shape.** Each circle represents a single event (tuple) that has been posted to the Event Heap server.
- **Position.** The location of each circle indicates the machine that posted the event. Regions are labeled with machine names.
- **Color.** All events with the same type (e.g., "light controller events" or "multibrowse events") are displayed in the same color; events of different types are shown in distinct colors.
- **Size.** The diameter of the circle indicates the "time to live" value for the event it represents.
- **Transparency.** The longer the event remains on the server, the more transparent it becomes.

AMBIENT PRESENTATION OF SYSTEM STATE

Our observations of the Event Heap Visualizer's impact on end-user debugging of the iRoom were conducted over a two-week period during which the visualization was present as part of an ambient display projected onto the room's conference table [3]. Ambient displays are displays that provide peripheral awareness of non-critical information. As an ambient display, the visualization was not meant to be the focus of attention for iRoom users; however, even though it was relegated to the periphery, it was effective in its goal of communicating system status information to end users. By presenting the visualization in an ambient display format, it was unobtrusive, and did not force itself upon users who were not interested in involving themselves in maintaining and improving the ubicomp space.

SYSTEM AWARENESS AND DEBUGGING

This display has increased system awareness and understanding, and has helped identify the cause of technical breakdowns in the iRoom. Several incidents illustrate this benefit:

In one instance, a user attempting to multibrowse (send a special kind of event) with the same machine as both the source and recipient of the message, noticed that her computer had frozen. Initially, she thought her own machine was broken, but a glance at our visualization indicated the true nature of the problem – an infinite number of circles were appearing located near the spot on the display that represented her machine. The realization that her action caused an infinite number of events to be posted to the server identified a previously unknown bug in the iRoom's infrastructure.

Another aspect of the iRoom's infrastructure was improved when a user observed that a certain category of events were displayed as circles with an unusually large diameter, indicating that their default time to live was exceptionally high, which caused the events to remain on the server too long, potentially clogging it.

On another occasion, users of the iRoom who were not involved in designing or maintaining the infrastructure gained insight into the system design as a result of observing the Event Heap Visualizer. Several users who were observing the visualization noticed that executing a single "multibrowse" action caused two circles to appear, prompting a discussion of whether sending two messages in response to a single event was the most efficient way to design that particular application.

CONCLUSION

The Event Heap Visualizer has been an important factor in making our ubicomp space easier to use and maintain. We believe that visualizations for system understanding can be particularly beneficial in ubiquitous computing environments where the plurality of machines and transience of users makes traditional debugging difficult.

REFERENCES

1. Baecker, R., DiGiano, C., and Marcus, A. Software Visualization for Debugging. *Communications of the ACM*, 40(4) (April 1997), 44-54.
2. Bassil, S. and Keller, R. Software Visualization Tools: Survey and Analysis. *Proceedings of the Ninth International Workshop on Program Comprehension*, 7-17. Toronto, Ontario, Canada, May 2001.
3. Morris, M.R. Benefits and Challenges of Tabletop Peripheral Displays. *Adjunct Proceedings of UbiComp 2004*, in press.
4. Johanson, B. and Fox, A. The Event Heap: A Coordination Infrastructure for Interactive Workspaces. *Proceedings of WMCSA 2002*, 83-93.
5. Johanson, B., Fox, A., and Winograd, T. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive Computing* 1:2 (April – June 2002), 67-75.