In: *Behavior Research Methods, Instruments, and Computers*. 38(4). 656–659.

RUI—Recording User Input from Interfaces under Windows and Mac OS X

Urmila Kukreja[1], William E. Stevenson[2], and Frank E. Ritter[2,1]

[1] Department of Computer Science and Engineering
[2] School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16802

Draft of 28 October 2005, RUI15.doc

frank.ritter@psu.edu

+ 1 (814) 865-4453

Abstract

Event and timing logs are useful in studying human-computer interaction, evaluating applications, and comparing input devices. RUI (Recording User Input) is a tool that records user computer interface behavior. It is created in the .Net framework with C# for Windows and in the Carbon framework for Mac OS X. RUI runs in the background and works with software that runs under Windows or Mac OS X (10.3 "Panther" and later). We illustrate its use with a human-robot interaction interface, as well as present two simple tests that RUI passes and other timing software should pass: avoiding 0 ms timings and time distributions that follow a gamma (or gamma-like) distribution.

Keywords

event log; user logs; behavior playback; human-computer interaction; human-robot interaction

INTRODUCTION

Timing records of the way users interact with applications can give useful insights into human behavior and an application's usability. Timing records can be useful for comparing pieces of software, and are also necessary for building and testing user models.

Westerman et al. (1996) describe three different ways in which user information can be obtained. The first way, videotaping or a human recorder, is not always feasible. The video recording must then be analyzed, and the human recorder has known limitations. Camtasia by TechSmith is an example of such a tool to record and replay user interaction as a video file. However, timing information cannot be directly obtained and the tool creates logs that are resource intensive because they are videos.

The second method, instrumentation, is the augmentation of an interface such that it records the user's actions. This approach can only be used if the system studied includes logging or can be modified to include logging. This is thus not a general solution, as information collected in this way is limited to software that can be instrumented. For example, commercial applications such as Microsoft Word and many games cannot be instrumented to record user behavior.

The third method is to include an unobtrusive application that exists in the background, which can be used generically across all applications, and that records user behavior and timestamps it. We introduce here such a program, Recording User Input (RUI).

RUI belongs to the set of programs known as keystroke logging tools. Several such tools currently exist but there are few that record both mouse and key events and even fewer that give logs in a form that would be useful to HCI researchers and practitioners. Several of these recording and playback tools can be downloaded from the web, but typically these were developed as malicious "spyware" to provide keystroke logs without timing information.

A tool similar to RUI was developed (Westerman et al., 1996) for the Windows 3.1 platform but it appears to be no longer available. MICELAB (Baccino & Kennedy, 1995) is yet another similar tool, but does not run on modern computers (though their analyses are still helpful in analyzing mouse movements). Another tool, InputLogger, can be used to obtain user interactions across generic interfaces but it works only on the Classic Macintosh (pre-Mac OS X) platform (Trewin, 1998). There are also several large commercial products available.

DESCRIPTION OF RUI

RUI is a keystroke and mouse action logger for the Windows (2000 and XP) and the Mac OS X (10.3 and later) platforms. There are options in RUI's user interface to choose which types of actions to record, including keystrokes and mouse movements. The data collected is stored in a log file as shown in Figure 1, as a list of the timestamp, action, and arguments (if any, such as key pressed or move location). To start and stop recording, hot keys are provided (Ctrl+R to Record, and Ctrl+S to Stop). The amount of data recorded with extensive mouse movements can be as high as 22k / min. If the user exclusively moved the mouse it could be as high as 156k / min. The timestamp for a single keystroke is approximately 19 characters, so less mouse-intensive activities will have correspondingly much smaller log files.

The data collected can be replayed using an included Replay option, and can be played faster or slower than real-time. While replaying the keystrokes, if the observer wishes to see the same results occur in the interface, it is essential that the system be in the same state it was in before the recording started. Otherwise, the mouse moves and the keystrokes are created and performed by RUI but the resulting effects are likely to be different. It is worth noting that very fast-paced interfaces with variable system time lags may lead to difficulties with playback.

<Insert Figure 1 about here.>

Implementation of RUI for Windows

RUI for Windows is developed in the .Net framework using C#. One key feature of this application is that it uses events to capture keystrokes and mouse clicks. Thus, whenever there is a change in the state of the mouse buttons, the position of the mouse, or the state of any of the keys on the keyboard, an event is raised. To record mouse moves, a thread runs in the background checking if there is any change in the position of the mouse. If there is a change in the position of the mouse, a mouse move event is raised. The fastest rate at which RUI collects mouse movement data is 10 ms (100 Hz). This rate can be adjusted to be slower to create smaller log files.

Mouse activity is recorded with .Net using the MousePosition and MouseButton properties of Windows Forms. However, these forms will record activity only while the focus is on the application developed with .Net. To record generic mouse activity while working on other applications, the Control.MouseButtons and Control.MousePosition classes were used. To capture global keystrokes, a Win32 API function, GetAsyncKeyState was used. The System.DateTime function of the .Net framework was used to calculate the elapsed time between subsequent actions.

User actions are replayed using the log file, setting the mouse position based on recorded values using the Cursor.Position function. To create mouse clicks and keystrokes on generic applications, mouse_event and keybd_event functions of the User32.dll are used.

Implementation of RUI for the Macintosh

RUI for Mac OS X is developed in C using the Carbon Event Manager APIs of the Carbon development framework. An event handler that listens for raw key down, mouse movement, and mouse click actions is installed, and is automatically called when such an event occurs. Inside the event handler routine, the event that occurred and its type is determined, and appropriate information similar to that used by RUI for Windows is logged into a text file. Playback of events is done using the CGPostKeyboardEvent and CGPostMouseEvent calls of the Quartz Services API, which provides access to low-level features of the Mac OS X window server.

Use of RUI Data

The data collected using RUI can thus be used to measure response times of the user interacting with a wide range of interfaces and the time course of their behavior. In addition, it is possible to obtain information about user interaction and preference of different input devices can be obtained, for example, the time to switch from mouse to keyboard. We describe an example application of RUI applied to a complex interface.

Data collection in robot studies is a difficult task (Scholtz, Young, Drury, & Yanco, 2004), and thus makes a nice test environment. The Windows version of RUI was used to collect data in a human-robot interaction study.

RUI provides a rich set of data that has been successfully used to examine user interaction with the ER1 (www.evolution.com/er1/) robot's interface over sessions of up to four hours. The robot interface with its controls is shown in Figure 2. The ER1 robot is a simple three-wheeled robot that carries a laptop as its controlling computer. The carried computer can drive the robot or remote use is possible by networking with another computer through a wireless connection. This figure illustrates a complex interface that RUI can gather data from. The data were studied using playback and other analysis tools, and gave details about response times, number of errors, and even mouse down durations across individuals as they did a variety of tasks (Kukreja, 2004). The data collected with RUI helped examine differences across novice and expert robot users and to understand several aspects of learning and visual memory. A cognitive model of user interaction was created in the ACT-R cognitive architecture (Anderson & Lebiere, 1998) and the model was validated against human data gathered using RUI (Kukreja, 2004).

<Insert Figure 2 about here.>

A simple test of the quality of the timing data is shown in Figure 3. The histogram represents the times between actions over a 15-minute period of a single user interacting with the ER1 human-robot interface. Mouse movements, mouse clicks, and keystrokes are included.

The histogram shows that the distribution is right skewed (mean greater than the median, a long tail to the right, the usual case with response time data). Response time distributions should follow a positively skewed distribution like a gamma (or a Wald, Weibull, or ex-Gaussian). Van Zandt and Ratcliff (1995), Van Zandt (1995), and Luce (1986), to note just a few reports, provide further discussions on this topic, some arguing for other, related distributions.

The distribution can be seen to approximate two Gamma distributions. The first peak at 10 ms is for the mouse movement distribution, dominated by the 10 ms update during movements. The second, shallower peak, around 175 ms, is for keystrokes.

Most importantly, a possible artifact that is missing is a spike of 0 ms response times. Such a spike would indicate that the timing program is not being called often enough and is batching the time stamping actions, although, with rapid keystroke actions (which this does not include), simultaneous keypresses from both hands are possible but unlikely.

There could be other types of problems with RUI, but it passes these two tests related to the distribution of response times (gamma distribution and no spike at 0 ms). The use of the log scale helps show the keystroke and mouse click data without being distorted by the frequent mouse movement data generated every 10 ms when the mouse is moving. A similar curve is generated by the Macintosh version.

<Insert Figure 3 about here.>

SUMMARY

RUI provides a lightweight tool to record user behavior on Windows and Mac OS X machines. It records mouse movements, mouse button presses, and keystrokes. The resulting logs are readable by humans and other software, and RUI can play them back for the analyst to see the details of the behavior as it emerges.

RUI has been used to study human-robot interaction, which was briefly described here as an example application. The resulting logs have been checked for correctness with two simple tests.

RUI will provide a useful logging tool for researchers interested in recording behavior on most interfaces because it does not require instrumenting the interface to study how users interact. Because it is available without cost and its source code is available on the web, RUI might have a longer life than previous comparable tools. Users are likely to find other uses for it. For example, the logs can be used for other analyses such as plotting paths of users' mice, and computing where the mouse is typically located.

Availability

The software is freely available at http://acs.ist.psu.edu/RUI, including the source code, installation instructions (a README), example log files, and a movie of its application. RUI for Windows requires that the .Net framework be installed. Details and the code for .Net are available at msdn.microsoft.com/netframework/downloads/framework1_1/ . RUI for Mac OS X works without system modification.

Acknowledgements

The Windows version of RUI was part of Urmila Kukreja's MS thesis in Computer Science and Engineering. This project was partially supported by ONR Contract N00014-02-1-0021 and N00014-03-1-0248. Richard Carlson, Georg Jahn, Jong Kim, Diana Rösler, and two anonymous reviewers provided useful comments that improved this paper.

REFERENCES

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum.

Baccino, T., & Kennedy, A. (1995). MICELAB: Spatial processing of mouse movement in Turbo-Pascal. *Behavior Research Methods, Instruments, & Computers, 27*(1), 76-78.

Kukreja, U. (2004). *Towards model-based evaluations of human-robot interfaces.* Unpublished MS thesis, Department of Computer Science and Engineering, The Pennsylvania State University.

Luce, R. D. (1986). *Response times: Their role in inferring elementary mental organization*. New York, NY: Oxford.

Scholtz, J., Young, J., Drury, J., & Yanco, H. (2004). Evaluation of human-robot interaction awareness in search and rescue. In *ICRA 2004*.

Trewin, S. (1998). InputLogger: General -purpose logging of keyboard and mouse events on an Apple Macintosh. *Behavior Research Methods, Instruments and Computers, 30*(2), 327-331.

Van Zandt, T. (1995). Analysis of response time distributions. In J. Wixted (Ed.), *Stevens' handbook of experimental psychology* (Vol. Volume 4: Methodology in experimental psychology, pp. 461-516). New York, NY: John Wiley & Sons.

Van Zandt, T., & Ratcliff, R. (1995). Statistical mimicking of reaction time data: Single-process models, parameter variability, and mixtures. *Psychonomic Bulletin & Review, 2*(1), 20-54.

Westerman, S. J., Hambly, S., Alder, C., Wyatt-Millington, C. W., Shrayane, N. M., Crawshaw, C. M., & Hockey, G. R. J. (1996). Investigating the human-computer interface using the Datalogger. *Behavior Research Methods, Instruments, & Computers, 28*(4), 603-606.

Figure captions

Figure 1. An example log of RUI showing the user moving the mouse, clicking, moving again, and then typing "hello."

Figure 2. The ER1 robot interface.

Figure 3. Histogram of the time between actions for 15 minutes of user behavior with the ER1 human-robot interface.

```
er.txt - Notepad
File  Edit  Format  View  Help
00:00:02:874    Moved   97      321
00:00:02:884    Moved   97      319
00:00:02:894    Moved   98      316
00:00:02:914    Moved   98      315
00:00:02:954    Moved   98      314
00:00:02:964    Moved   98      313
00:00:03:575    Pressed Left
00:00:03:835    Released Left
00:00:04:126    Moved   98      311
00:00:04:136    Moved   102     309
00:00:04:146    Moved   112     303
00:00:04:166    Moved   132     295
00:00:04:176    Moved   168     279
00:00:04:196    Moved   215     260
00:00:04:206    Moved   265     242
00:00:04:216    Moved   313     232
00:00:04:236    Moved   349     230
00:00:04:246    Moved   371     235
00:00:04:256    Moved   389     245
00:00:04:276    Moved   400     257
00:00:04:286    Moved   406     269
00:00:04:296    Moved   411     286
00:00:04:316    Moved   411     299
00:00:04:326    Moved   406     312
00:00:04:336    Moved   400     324
00:00:04:356    Moved   392     340
00:00:06:379    Key     h
00:00:06:509    Key     e
00:00:06:709    Key     l
00:00:06:860    Key     l
00:00:07:991    Key     o
```

Figure 1. An example log of RUI showing the user moving the mouse, clicking, and then typing “hello.”

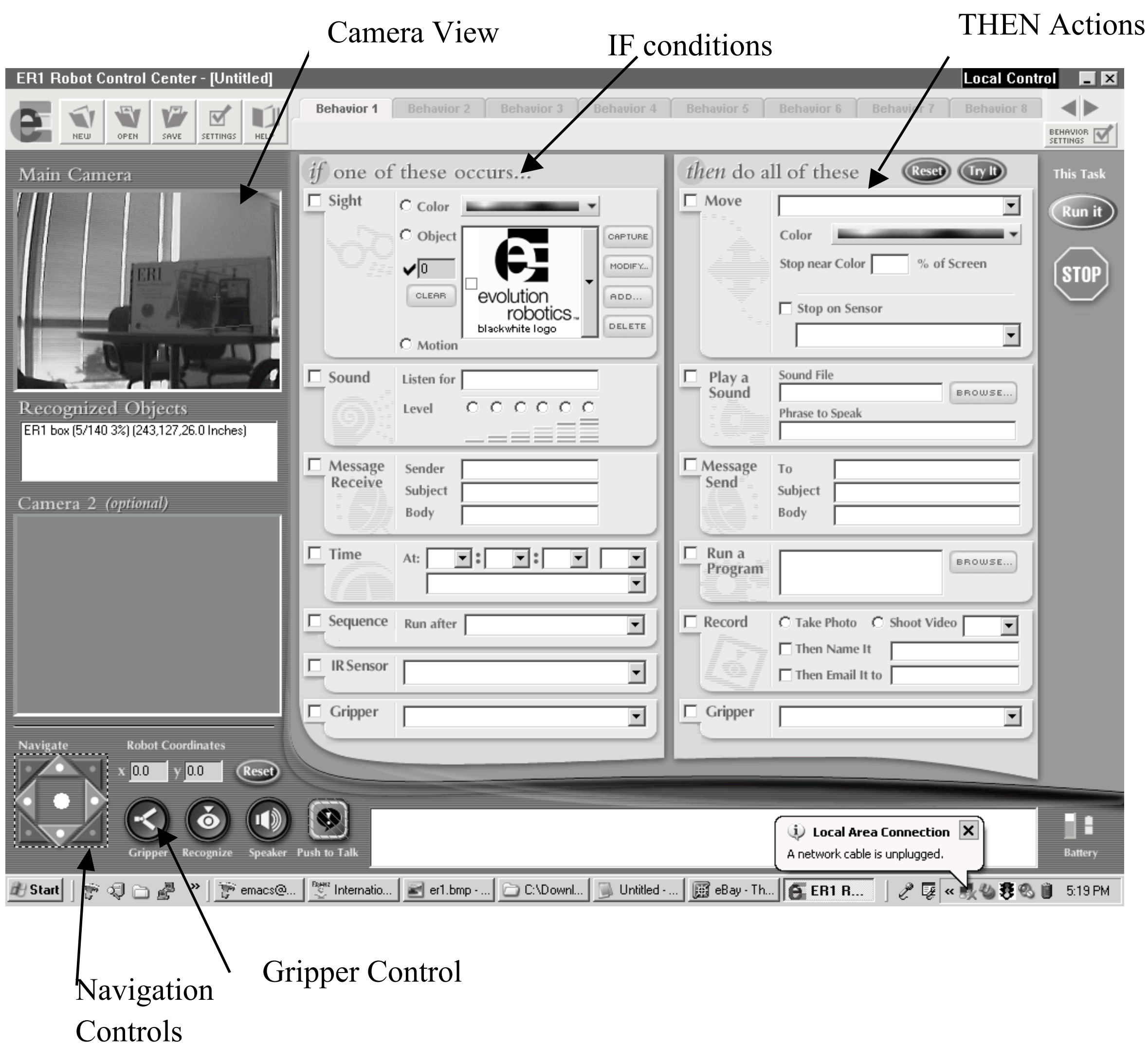

Figure 2. The ER1 robot interface.

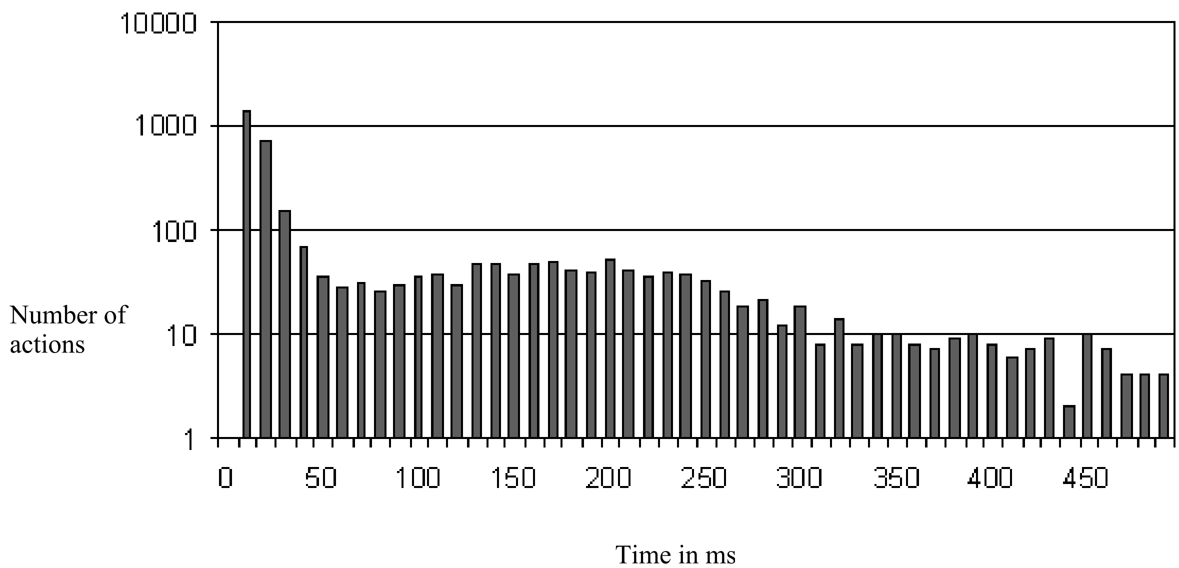

Figure 3. Histogram of the time between actions for 15 minutes of user behavior with the ER1 human-robot interface.