Using Surveillance Software as an HCI Tool

Blaise W. Liffick
Laura K. Yohe
Department of Computer Science, Millersville University
Millersville, PA, 17551, USA

Abstract

Laboratory equipment (both hardware and software) for conducting experiments, usability studies, and field studies in the area of human-computer interaction (HCI) is typically complex, bulky, expensive, and intrusive. Recent strides in the development of surveillance software offer the prospect of a non-invasive, inexpensive, and largely automatic way of capturing data from user activities that could be useful to HCI professionals, researchers, and educators. This project investigates this possibility.

Keywords: Human-computer interaction, usability study, surveillance

An article that appeared in the Wall Street Journal in March 2000 described two low-cost surveillance software packages that were finding success on the market (McCarthy 2000). This article coincided with a problem that is common to small human-computer interaction (HCI) labs: finding non-intrusive, inexpensive ways to monitor computer users during experiments, usability studies, or field tests. Equipment and software being marketed for usability labs currently costs tens of thousands of dollars, a sum beyond the reach of small labs or companies (UserWorks 2001). Surveillance software, however, is relatively inexpensive (a few hundred dollars). We wondered whether such software might be used in place of much more extensive (and expensive) lab setups which typically use several video cameras and observation-recording software.

One of the classic problems with observing users can be referred to as the panopticon effect, Hawthorne effect (Preece *et al* 1994) or the Heisenberg effect. Panopticon was a term used by Jeremy Bentham in 1787 to describe a type of prison constructed in the round, with cells on the outside of a central core that is manned by a guard (Johnson 2001). The cell wall facing the guard is a one-way window, so that the guard can see into each cell, but the prisoners could not see the guard. Bentham conjectured that it would not be necessary to always have a guard present in order to ensure good behavior on the part of the prisoners, since the prisoners could not verify when the guard was indeed present—they would adopt behavior on the assumption that they were being observed. The Hawthorne effect refers to a study done in 1939 in a Hawthorne, Illinois manufacturing plant. The Heisenberg effect comes from physics and, simply put, states that the act of observing a particle changes the behavior (velocity or direction) of that particle. All terms apply to the problem of trying to observe users without affecting how they behave. Such problems are nothing really new, and can be traced back to the use of "efficiency experts" more than 100 years ago (Edgar 1997).

HCI researchers and practitioners typically use video cameras to observe both the user and the computer's monitor during use. While it is possible to hide the cameras to a certain extent, and while some experts believe that users forget about the cameras in a short amount of time (Shneiderman 1998), there are times when setting up several cameras is difficult (such as in the field) or at least expensive. Furthermore, while a camera can faithfully record most of the actions visible on the monitor, it is not recording the actual actions of the user, i.e. what keys the user pressed, when the mouse was clicked, which mouse button was clicked, etc. While some of this could certainly be deduced by watching a videotape, to do so would also be extremely time consuming (and, therefore, expensive). Automated data collection software has largely been custom built and is not widely available, so videotaping is still the preferred way of recording user interactions for most HCI professionals (Drury *et al* 1999).

This paper, then, describes an experiment conducted to determine whether surveillance software could be used as an inexpensive, non-intrusive, automatically logging tool for HCI work.

1. THE PROJECT

Two software products were selected to test as possible HCI tools. One of the products proved to be too buggy to use (inquiries to the company were ignored), and was dropped from the study. The product selected, Silent Watch by Adavi (Adavi 2001), has received substantial amounts of press in the past year, so appeared that it might be a reasonable place to start our project. Since this current project is more of a feasibility study, it was decided that the initial study could successfully be completed with a single product, leaving a comparative study of multiple products to a future project.

Silent Watch clearly fit the first two criteria we were concerned about, cost and invisibility to the user. The product costs about $225 per package, which includes a license for monitoring up to four machines. Although software must be installed on the machine to be monitored, an option in Silent Watch allows it to be placed into "stealth" mode, so that the user has no way of discovering that the software even exists on their machine. This is perfect for overcoming the Heisenberg effect, a considerable concern in HCI studies.

The major portion of the project, of course, is evaluating the software's performance for logging user activity.

Silent Watch software consists of two parts: the viewer and the client. The viewer software is loaded onto the test conductor's computer to observe the test subject's machine. The client software is loaded onto the test subject's computer to allow the viewer machine to pick up the image of the subject's machine as well as any use of the keyboard and mouse. Transactional information is transmitted from the client to the viewer system over an existing local area network.

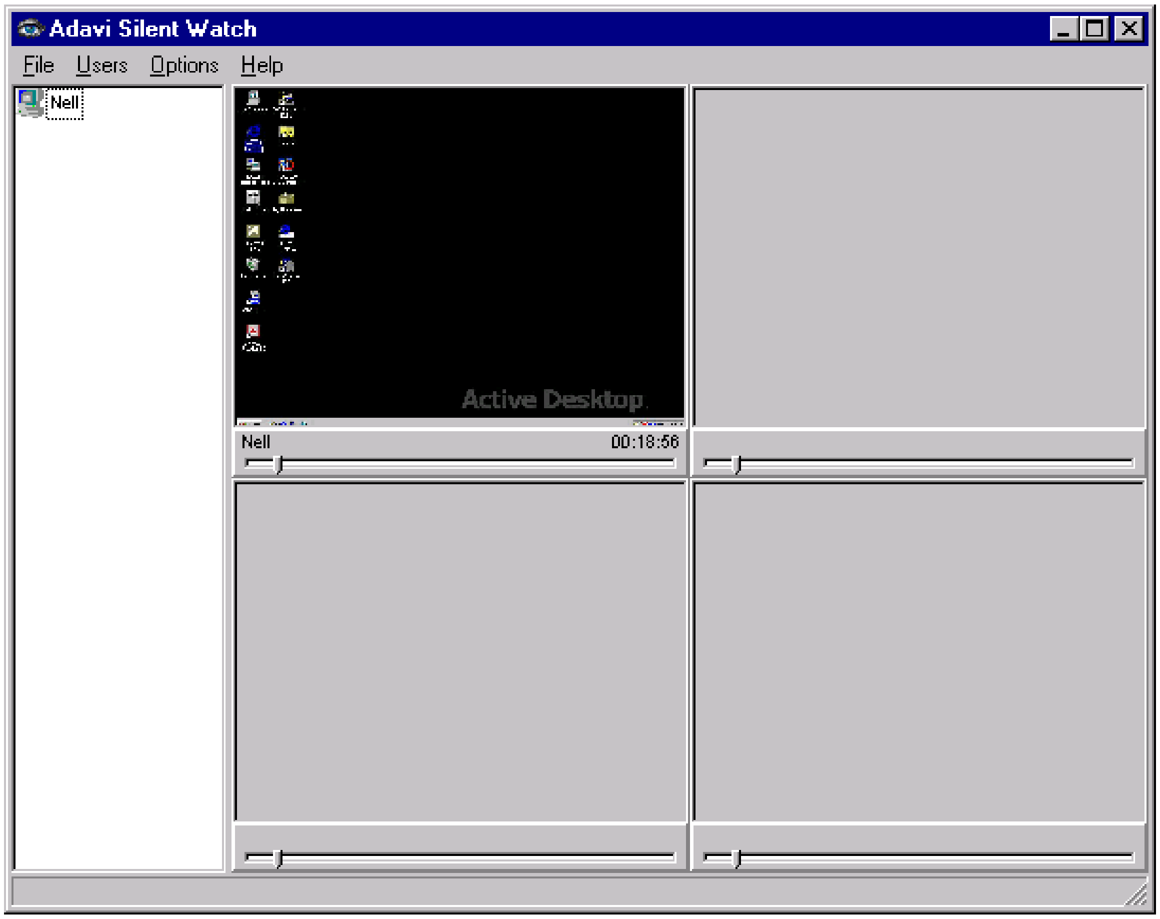


Figure 1. Main Window of Viewer

Figure 1 shows a typical display of the viewer system. The viewer can be configured to monitor up to 49 users simultaneously. The figure shows the viewer configured for four users, with only one currently active (called Nell). The client software records keystrokes and other data (such as URLs) and forwards these records on to the viewer system, which then updates the viewer display as well as its own activity files. The amount of time between snapshots of a client system is settable, with the minimum time of 3-second intervals.

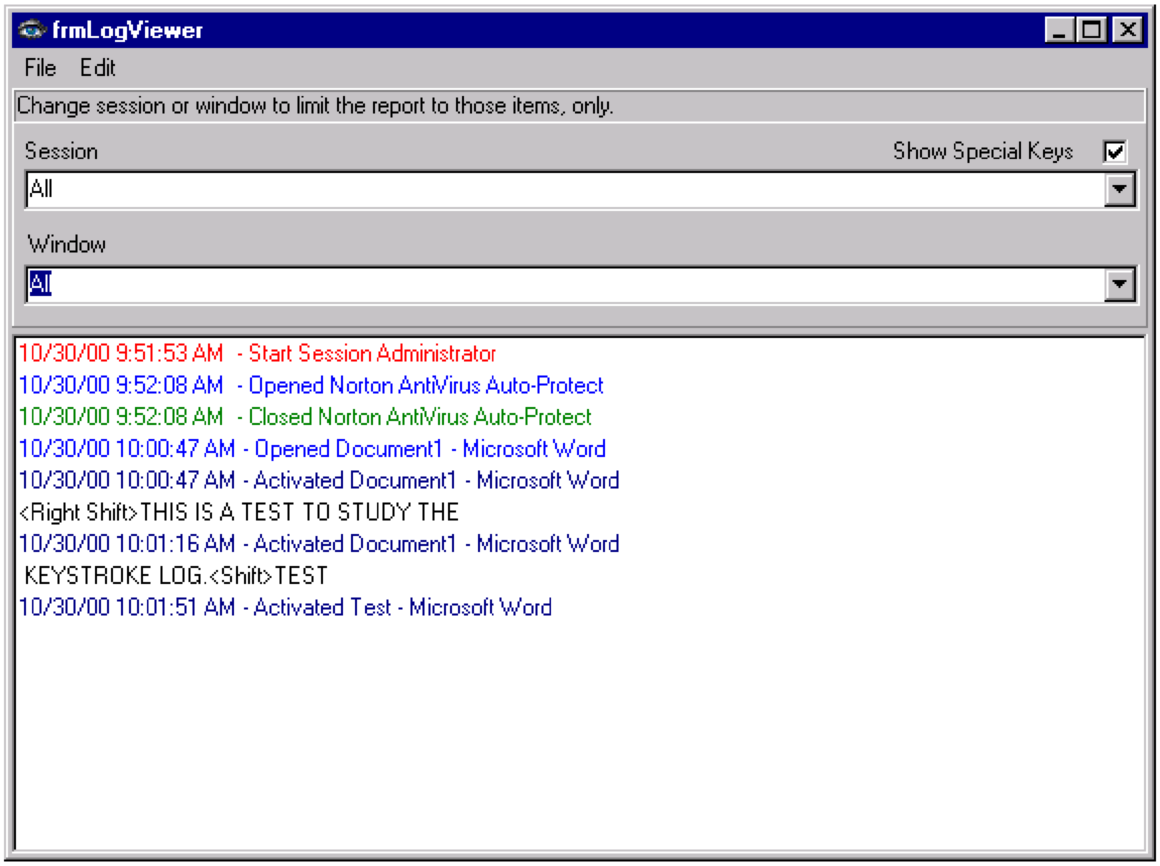
Log Files

The question we hoped to answer was whether Silent Watch's log files provided enough detail that could be analyzed to provide meaningful information for a usability study. Initial testing showed that the two likeliest useful files were the keystroke log and the URL log.

The keystroke log displays all characters entered from the keyboard, including erasures made with the <backspace> key and special keys such as <enter>, <shift>, and function keys (see Figure 2). In addition to displaying what has been typed, it provides a time/date stamp down to the second and states when a user:

1. Logged onto a machine, which is indicated by "Start Session" and the user login name,
2. Opened an application, which is indicated by "Created" and the name of the application,
3. Activated an already open application, which is indicated by "Activated" and the name of the application, and
4. Closed an application, which is indicated by "Closed" and the name of the application (Adavi, 2000).

The URL log specifically displays the Uniform Resource Locator (URL), i.e. website addresses, for every Internet site the machine has accessed or attempted to access. It continues to log this information even when the viewer machine is off. A sample of the URL log viewer can be seen in Figure 3. It can be saved and printed by using the options in its file menu. For every address, there is a time/date stamp down to the minute that shows when the site was accessed.



Change session or window to limit the report to those items, only.

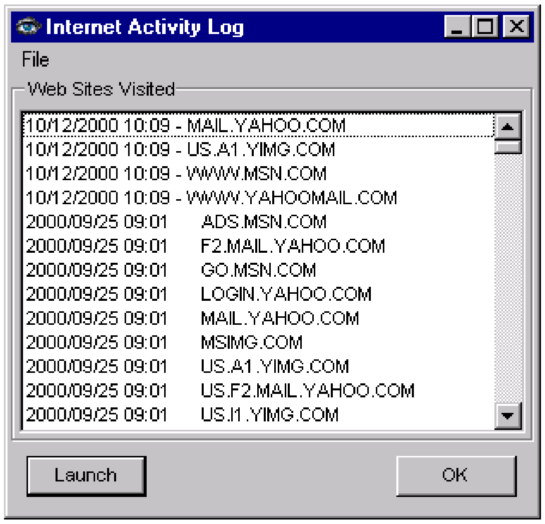
Session: All

Show Special Keys: ✓

Window: All

```
10/30/00 9:51:53 AM  - Start Session Administrator
10/30/00 9:52:08 AM  - Opened Norton AntiVirus Auto-Protect
10/30/00 9:52:08 AM  - Closed Norton AntiVirus Auto-Protect
10/30/00 10:00:47 AM - Opened Document1 - Microsoft Word
10/30/00 10:00:47 AM - Activated Document1 - Microsoft Word
<Right Shift>THIS IS A TEST TO STUDY THE
10/30/00 10:01:16 AM - Activated Document1 - Microsoft Word
 KEYSTROKE LOG.<Shift>TEST
10/30/00 10:01:51 AM - Activated Test - Microsoft Word
```

Figure 2. The Keystroke Log



```
10/12/2000 10:09 - MAIL.YAHOO.COM
10/12/2000 10:09 - US.A1.YIMG.COM
10/12/2000 10:09 - WWW.MSN.COM
10/12/2000 10:09 - WWW.YAHOOMAIL.COM
2000/09/25 09:01     ADS.MSN.COM
2000/09/25 09:01     F2.MAIL.YAHOO.COM
2000/09/25 09:01     GO.MSN.COM
2000/09/25 09:01     LOGIN.YAHOO.COM
2000/09/25 09:01     MAIL.YAHOO.COM
2000/09/25 09:01     MSIMG.COM
2000/09/25 09:01     US.A1.YIMG.COM
2000/09/25 09:01     US.F2.MAIL.YAHOO.COM
2000/09/25 09:01     US.I1.YIMG.COM
```

Figure 3. The URL Log

The Experiment
In order to test the Silent Watch software, we created a dummy usability experiment, ostensibly to study the differences between using a mouse, trackball, and touch pad as a pointing device. The intent was to pretend we were conducting a comparative experiment on these devices while using the Silent Watch software to monitor the users' interactions. The keystroke and URL logs would then be analyzed to determine whether they provide any useful information regarding the interactions. Specifically, we were interested in whether the logs would show each user's actions accurately and with sufficient detail required of a real usability study.

Twenty-two computer science juniors and seniors in an HCI course were divided into three groups, one for each pointing device. Each student was given a set of written protocols to follow in order to accomplish specific tasks. Protocols were composed for four different activities: word processing, spreadsheet use, Internet browsing, and using a graphing calculator. Figure 4 shows the protocol for the Internet Browsing task, along with a typical resulting keystroke log. In this example, note that although the keystroke log records most of the activity of the protocol steps, it doesn't do so discretely – most of the actions have been recorded as part of the first time-stamped event in the keystroke log.

The test trials were conducted in the typical fashion for a usability study. The test conductor randomly assigned each of the students to one of the three test groups. One at a time the subjects were given one of the four protocols to complete, as quickly as they could. The tests were conducted in an HCI lab that was more or less isolated from other activities. The test conductor observed each test trial while monitoring the progress of each on a nearby system (which contained the Silent Watch Viewer software). Although the tests were not conducted in as rigorous a way as would be normal for an actual usability study, the conditions were close enough to an actual test as needed for our purposes. The results of the protocols themselves are unimportant to this project - it doesn't matter how quickly or accurately the test subjects were actually able to accomplish their given tasks. Although we did some simple analysis of the test results, the purpose of doing so was to see how well the log files that are automatically recorded by Silent Watch captured the actions of the test subjects.

The Results
The results were basically as expected, a mixture of success and failure, and a promise of potential improvements that would produce a very useful tool.

First, the additional overhead of the Silent Watch software does not appear to be noticeable to the user, such as by adding in unexpected delays as a result of logging activity. This was true even though the systems the tests were conducted on were very slow by today's standards (200 MHz Pentium II's). So the software lived up to its promise of stealth. This is important for two reasons: (1) users are not reminded (although they are initially told) that they are being monitored by occasional delays of the system during their activities, and (2) the use of the software does not appear to add noticeable overhead time for completing tasks, so that timed tests would be reasonably accurate.

Next, the keystroke log does a pretty good job of recording what the user has entered at the keyboard, including any special keys and function keys. Although all entered text is displayed in all upper case, the actual characters entered can be deduced by following the use of special keys such as <shift> and <caps lock> (it even differentiates between the left and right shift keys). Time stamps for launching and closing applications are recorded to the second, adequate perhaps for many kinds of usability studies. It was not difficult to calculate the time between certain events using these time stamps. In addition, certain types of errors, especially typos, are easily determined. It is easy to imagine a program to assist in such mundane analysis, e.g. by counting the number of times the <backspace> character appears in a log file or calculating the time between two events. It is also not very difficult to match up elements of the keystroke log with the protocol steps, where that's possible. For instance, Figure 4 shows how some of the logged data matches the given protocol.

The URL file is less useful. Although it contains the time a particular site is entered, and records even the launch of auxiliary components of a website (note the launching of go.msn.com in Figure 3, for instance),

1. Double click on the Internet Explorer/Netscape Browser icon on the desktop
2. Type www.yahoomail.com in the address slot of the browser and hit enter
3. Type in user name and password
4. Click on check e-mail
5. Click on compose
6. Send the message to self
7. Type "This is a test" in the text box
8. Click on the "Send" button
9. Type www.wunderground.com in the address slot of the browser and hit enter
10. Type "17601" in the text box for fast forecast and hit the <enter> key
11. Type www.adavi.com in the address slot of the browser and hit enter
12. Click on the Download tab
13. Fill in the text boxes but do not click on the "Download" button
14. Type www.yahoo.com in the address slot of the browser and hit enter
15. Search for some topic by typing the topic in the text box
16. Type www.millersville.edu in the address slot of the browser and hit enter
17. Click on the Max icon
18. Click on the "Personal Information" button
19. Type in user identification and password and hit enter.
20. Type in password again and hit enter
21. Type www.citibank.com in the address slot of the browser and hit enter
22. Select United States from the country list
23. Select Credit Card Account Online from Products/Services List (automatically takes you to the desired site)
24. Type in user name and password and hit enter
25. Click on the link for unbilled activity
26. Click on the "x" in the upper right corner of the window to close it

Figure 4a. A Typical Test Protocol for Internet Browsing

```
1.  10/30/00 10:09:28 AM - Activated http://www.msn.com/ - Microsoft Internet Explorer
     WWW.YAHOOMAIL.COM<Enter><Shift>LEORAH3779<Tab>******<Enter>
    <Shift>LEORAH3779<Right Shift>@YAHOO.COM<Shift>THIS IS A
    TEST.WWWIW<Backspace><Backspace>.WUNDERGROUND.COM<Enter><Num 1>
    <Num 7><Num 6><Num 0><Num 1><Num Enter>WWW.ADAVI.COM<Enter>
    <Shift>DR. <Right Shift>ROY <Shift>ROGERS<Tab><Right Shift>ASSOCIATE
    <Shift>PROFESSOR<Tab><Shift>TRIGGI<Backspace>ER
    <Shift>UNIVERSITY<Tab><Shift>PO <Right Shift>
    BOX 1002<Tab><Shift>DEPARTMENT OF <Right Shift>COMP <Right
    Shift>SCI<Tab><Shift>TEMPE<Tab><Shift>A<Right Shift>Z<Tab><Num 1>
    <Num 7><Num 5><Num 5><Num 1><Tab><Shift>UNITED <Right Shift>STATES OF <Right
    Shift>AMERICA<Tab>717-123-4567<Tab><Shift>
    ROGERS<Right Shift>@CS.TRIGGER.EDU<Tab>WWW.YAHOO.COM<Enter>LOGGING
    SOFTWARE<Enter>WWW.TRIGGER.EDU<Enter><Num 1>
    <Num 7><Num 3><Num 6><Num 0><Num 2><Num 9><Num 9>
    <Num 4><Tab><Num 0><Num 3><Num 0>
    <Num 7><Num 7><Num 9><Num Enter><Num 0><Num 3><Num 0><Num 7>
    <Num 7><Num 9><Num Enter>WWW.CITIBANK.COM<Enter>
23. 10/30/00 10:16:49 AM - Activated https://www.accountonline.com/CB/Login.dcl -
     Microsoft Internet Explorer
     <Shift>LEORAH3779<Tab>********<Enter>
26.  10/30/00 10:17:45 AM - Closed Unbilled Activity - Microsoft Internet Explorer
```

Figure 4b. The Keystroke Log for the Internet Browsing Test
The numbers to the left of the time stamp indicate which operation in the protocol the logged event corresponds to.
Note: Passwords in the log were replaced with "*," but the log does record them in plain text.

this information is only marginally useful, except, perhaps, in analyzing how a particular website launches, and how long it takes to complete the overall launch.

On the negative side are a number of shortcomings that would need to be addressed before this particular product would be particularly useful for HCI work:

1. The time stamps are not fine enough. In the keystroke log they are only down to the second. Down to the tenth of a second would make this feature much more useful for usability research, as many actions can happen much faster than in one second. In the URL log, time stamps are only to the minute. Again, an order of magnitude improvement in the accuracy of the time would be a great improvement.
2. Mouse clicks are not recorded. Where and when mouse buttons are activated is often important to HCI work. They need to be recorded, again probably down to the tenth of a second.
3. The keystroke log is perhaps not complete enough for work on web applications. For instance, the field into which particular data is being entered is not recorded.
4. The URL log does not record all web pages visited, only when each website is entered. It would also be helpful to record whenever a link (in whatever form: menu, button, embedded link, etc.) is activated (including the time), although if all web page addresses were recorded with a fine enough time stamp, this might be deduced.
5. The mouse icon is not visible on the Viewer screen, which makes it difficult for an observer to follow the user's actions. Actions such as mouse clicks must be deduced by, for instance, observing that a particular application has just been launched or a menu been displayed.
6. There is no ability to save the screen snapshots from the Viewer. This means that the screen would need to be recorded to a video tape for later analysis, an additional step and expense of equipment.
7. The minimum three second refresh cycle for data transmitted between the client (subject) system and the viewer (test conductor) system is probably a bit too long. One second would be preferable, and with today's fast machines should be possible without degrading the performance of the client system. A one-tenth cycle would be preferred.

3. CONCLUSIONS

This study succeeded in identifying weaknesses of one surveillance software package when attempting to use it as an HCI tool. It further showed, however, that surveillance software has the potential for such use. The weaknesses described above are felt to be within the technical limits of software development and the capabilities of current hardware.

4. FUTURE RESEARCH

An extensive comparative study of other surveillance software packages might prove instructive by perhaps identifying other products that already address some of the concerns indicated above. Ultimately it is hoped that we could assist companies such as Adavi to further develop their surveillance products so that they would provide an inexpensive tool for HCI professionals and educators.

5. ACKNOWLEDGEMENTS

Special thanks to Adavi, Inc., for their donation of the Silent Watch software for this project. Also, the laboratory equipment used in this study was made possible by National Science Foundation grant #DUE-9551245.

6. REFERENCES

Adavi, Inc. www.adavi.com. June 2001.

Drury, Jill, Tari Fanderclai, and Frank Linton. "Automated Data Collection for Evaluating Collaborative Systems." SIGCHI Bulletin. Volume 31, Number 4 (October 1999): pp. 49-52.

Edgar, Stacey. *Morality and Machines.* 1997. Jones and Bartlett Publishers. Sudbury, MA.

Johnson, Deborah. *Computer Ethics, 3rd Edition.* 2001. Prentice Hall Publishing. Upper Saddle River, NJ: p. 113.

McCarthy, Michael J. "You Assumed 'Erase' Wiped Out That Rant Against the Boss? Nope". Wall Street Journal (03/07/00): p. A1.

Preece, J. *et al*. *Human-Computer Interaction.* 1994. Addison-Wesley Longman. Reading, MA.

Shneiderman, Ben. *Designing the User Interface, 3rd Edition.* 1998. Addison-Wesley Longman. Reading, MA: p. 131.

UserWorks, Inc. Sales Literature. June 2001. www.userworks.com.