

Automated Usability Testing

Elizabeth Chang and Tharam S. Dillon

School of Computer Science and Computer Engineering
La Trobe University, Bundoora, Melbourne, Australia
chang@latcs1.lat.oz.au; tharam@latcs1.lat.oz.au

ABSTRACT: Currently Laboratory Testing for Usability evaluation requires external monitoring and recording devices such as video and audio, as well as evaluator observation of user and user actions. It then requires review of these recordings, a time consuming and tedious process. We describe an automatic Usability Testing and Evaluation tool that we have developed. This consists of a piece of software called AUS (Automatic Usability Software) running at the operating system level that collects all the information related to user actions when s/he is using a particular application. It then displays this information in various useful formats as well as calculates suitable measures of Usability. There is no need with this system for any external recording devices. Patent protection has been applied for in respect of the inventive aspects of this system.

KEY WORDS: Usability Testing, Usability Evaluation, Usability Measurement.

1. INTRODUCTION

A variety of techniques for carrying Usability Evaluation are explained in the literature and these include: (a) Empirical testing [Molich and Nielsen 1990], (b) Inspection [Nielsen and Philips 1993], (c) Comparative Usability measures [Dillon and Maquire 1993], (d) Formal complexity based measures [Thimbleby 1994], (e) The MUSIC methodology [Bevan & Macleod 1994]

Empirical testing consists of testing an implementation of the user interface and software in a fairly controlled situation so as to ascertain the problems that the user is experiencing with the user interface. It can be carried out using three different environments, and these are: (1) Fixed Laboratory Testing, (2) Portable Usability Laboratory Testing, (3) Remote Usability Testing.

Fixed Laboratory Testing involves a test room and a

separate observation room, which are linked together usually through a one way mirror. Drawbacks of the fixed laboratories include: (i) they are relatively expensive, costing typically around US\$50,000 to \$100,000. (ii) the test is performed away from the work environment and could be artificial. (iii) it is expensive to shift staff observation and test staff to test site.

Portable Laboratory Testing moves equipment to the user site so the testing can be carried out in the user's company close to or at the actual work site where the user is likely to be using the software [Dorward 1994, Rowley 1994]. Advantages of Portable usability testing include, (a) it is much cheaper costing about US\$20,000 for a Portable Laboratory, (b) it involves less cost for the corporation as no people are moved to the test site, (c) the tests involved are carried out in a more realistic environment. This approach still relies on extensive video recording which has to be reviewed, a time consuming process. Further, there are only a limited number of things that are actually captured by

the video camera. In addition it still has the problem that test personnel and test equipment have to be sent to the site where the test subjects, are in order to carryout the testing.

Remote Usability Testing, uses computer networks and modem connections to monitor what the user is doing, so, neither the Usability tester nor the test subjects have to travel. However the disadvantages are that the Usability tester cannot observe the test subject's reactions, nor does s/he have access to the dialogue that the user is encouraged to enter into to provide some feedback.

All these Usability laboratory tests have been found to be very effective in actually picking up specific useability problems that relate only to the designated tasks. There could still be Usability problems with the User Interface which are not covered by the designated tasks. Furthermore, these approaches do not give an overall figure of merit either to the whole user interface or to significant aspects of the user interface. Thus it might uncover a specific problem but it does not give you an overall measure of goodness of the Usability of the user interface.

These difficulties with Usability Testing have meant that whilst it might be utilised for some expensive projects, it is not in widespread use for a large proportion of the software that is being produced today. However, the need to make software user friendly is becoming greater as time passes. There is clearly, therefore, a need for a more convenient approach to Usability Testing that would lead to more widespread Usability Testing.

2. NEW PROPOSAL

In the proposed method, a piece of software know as the Automated Usability Testing System is installed in the user's computer together with the software that is actually tested.

This Usability Software is automatically linked into the operating system. it runs as part of operating system. It works quietly in the background of the users' application. The user will not notice that there is Automatic Usability Testing software running. It is like many other tools in Windows, such as Print Manager, because it works in the background. You can continue working with your applications while the Print

Manager manages the printing queue to print many files frøm many applications. AUS can run in several ways. ie: click the icon or automatically start with the StartUP program. The Usability software opens and closes when the Windows Operating System comes on and off. It always does an auto save data at every time interval (which can be specified) and protects the data when computer is turned off. AUS is only run and used by UI analysts or UI evaluators, not by the user. AUS automatically collects all data concerned with UI analysis and Usability evaluation.

This Usability software should be able to monitor all the user actions when s/he is carrying out the specified series of tasks. It should be able to store all the information that it has gathered allowing the test personal to gather various statistics, or frequencies. The Automated Usability Testing software should also permit test personal to replay user actions and hence be able to review a user's test session, at normal speed and fast speed. This Usability Testing software must also be able to generate the required Usability Metrics.

3. SPECIFICATION OF AUTOMATED USABILITY SOFTWARE (AUS)

The AUS should be capable of eventually capturing all user interactions with the system through any user interface device. The Automated Usability System needs to be able to:

- Monitor multiple applications that the user is interacting with at the same time;
- Monitor multiple windows rather than just current activated window.

For the purpose of the prototype system, the following function should provide:

(I) UI data collection needs to occur:

- ⇒ Keystrokes within application; within each window and within each time interval.
 - Input and output from a port
 - Help system use
 - Delete (string, number..) function use
 - Select, searching or find (a menu, submenu or a item in a multiple choice field) function use;
 - Modification to what they have entered;
 - Retrieve function use;
 - Copy function use;

- Edit function use;
 - Undo function use;
 - Cancel function use;
 - Resizing of windows;
 - Move windows;
 - Scroll button use;
 - Network UI data transfer (Sound receiver, Video Recording)
 - ⇒ Mouse movements within application; within each window and within each time interval
 - Left mouse key down-up
 - Right mouse key down-up
 - Middle mouse key down-up
 - Double click
 - Drag and Drop
 - Mouse travel distance
 - (II) The following information needs to be stored for later retrieval by the AUS:
 - Keystrokes with window number and timer
 - Mouse location with window number and timer
 - System Messages with timer
 - (III) The Automatic Usability Software (AUS) needs to carry out the following analysis :
 - ⇒ Task time measurement data are collected as below:
 - Total task finishing time;
 - Total computer idle time (unproductive time or wasted time);
 - Total computer user dialogue time;
 - Total computer response time;
 - Average operation time;
 - ⇒ Error measurements determined include:
 - Number of dialogue errors;
 - Number of user requests from or to the system;
 - Number of user requests that failed due to user error;
 - Number of user requests that failed due to system error (eg not enough memory);
 - Number of times the application was terminated by user (user control to quit the system);
 - Number of times the application crashed (out of user control);
 - ⇒ Statistical Data analysis of the following type is carried out:
 - Fuzzy Logic Based Metric
 - Standard deviation among 4-100 users
 - Normal distribution fits to data
 - Learning curve (Correlation and Regression)
 - FFT and Spectral Analysis
 - Best Case /Worst Case Analysis
 - (IV) The Usability Results Output should display:
 - Historical keystrokes (virtual keys, extended keys, and system keys) within each predefined time interval;
 - Total density of mouse clicks within each predefined time interval;
 - Density of mouse clicks in each window opened within a selected time interval;
 - Total number of application windows opened (created) (eg 20 WD opened);
 - Times when each window is opened (activated or deactivated) (eg WD1 opened 10 times);
 - Times when each menu (system menu, pop up menu) is selected;
 - Times when each button is pressed (system button, speed button, dialogue button)
 - Time of every user action
 - Time of task completion
 - User action playback with selected speed, pause, fast forward, backward, stop and continue. (This allows a complete playback of all user actions carried out on the system within a specified time interval automatically).
 - Overall Usability figure of merit (Fuzzy Model)
 - Statistical graph and metric displays
 - A playback operation of all user interactions with the system at different speeds of display
 - User action sequence diagram
 - Mouse movement pattern
 - Three categories of user file creation and choice (Novice, Intermediate, Expert)
 - Up to 24 window partitions of the Screen for visual comparison of all outputs for up to 24 users.
- The AUS has two main components and these are:
 (1) UI data Collection; (2) Usability analyser and UI metrics. For the purposes of the prototype system, the following implementation has been done:
- Collect the messages user sends to the application;
 - Collect the messages the system sends to the user;
 - Collect every user action for user performance playback;

- Collect historical keystrokes (virtual keys, extended keys, and system key) and time stamps;
- Collect density of mouse clicks and mouse click time stamps;
- Collect density of mouse clicks in each window (WD) opened and open time stamps;
- Calculate total number of windows opened (created) (eg 20 Windows opened);
- Calculate the number of times each window (WD) is opened (activated or deactivated) (eg WD1 opened 10 times);
- Calculate times of each menu (system menu, pop up menu) being selected;
- Calculate times of each button being pressed (system button, speed button, dialogue button);
- Compare Mouse density for up to 24 users
- Compare Key Strokes for up to 24 users using Bar and Line graphs
- Draw mouse movement diagram.
- Output overall figure of merit

4. SYSTEM DISPLAY AND RESULTS

This section presents some typical user input screens, results screens and displays obtained using the Automated Usability Testing prototype system. The program can be placed in any Window group. It can be executed by double clicking on the AUS icon. Figure 1 shows AUS running in a minimised fashion, like PrintManager in the background of any user's application software.



Figure 1. AUS is running in a minimised fashion.

The main control of the AUS Prototype is shown in Figure 2. It allows the evaluator to view (i) the Statistical Distribution of Keystrokes with timer; (ii) Mouse clicks with timer and each window opened and

times of each window opened; (iii) Playback of user actions; (iv) overall figure of merit. It also allows the evaluator to pre-set the test time and data store time interval as well as transferring the UI data file across platforms. The File name is automatically assigned by a Date and Time and no user control is needed to run this program. The time is never duplicated. So when the user, for some reason reboots the system, AUS will continue working with a new time stamp on the file. The default recording time is 240 minutes (4 hours). This recording time can be reassigned by the evaluator for upto hundreds of hours. When the 4 hour limit is reached, AUS is closed automatically.

If the 4 hour limit is not reached, but the user wishes to take a break or simply to reboot the system, this poses no difficulties for AUS. It automatically saves the intermediate results gathered in every predefined time-interval, such as every 30 seconds or every 1 minute or 5 minutes. This time interval can be assigned by the evaluator. It depends on the type of the software you are testing. For a big system, there is no need to save every 30 seconds, but for some small software, it is necessary.

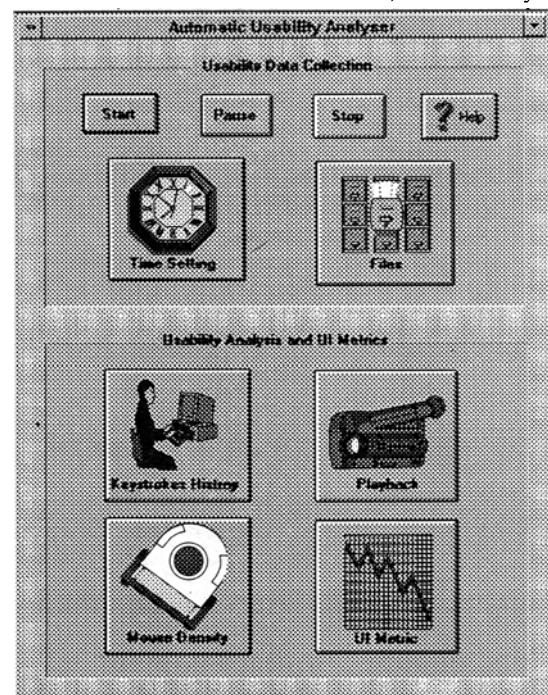


Figure 2. AUS main menu

Once the user finishes using the computer, the evaluator can either copy or transfer the data file to disk or other machine (Figure 3) and the test results at their own time or then can just directly run AUS on the user's machine. AUS has great potential to summarise the UI metric. We carried out some sample tests with the prototype. The following screen dumps from AUS are the results of testing Peter Coad OO Tools which are used for drawing OO conceptual models. The user had 3

to 4 hour of learning time with this tool before we started the recording. We asked the user to draw four different diagrams using this Tool from the projects: (i) Authors and Referee; (ii) Course Booking System;(iii) JCE Company Database; and (iv) HP Call Dispatcher.

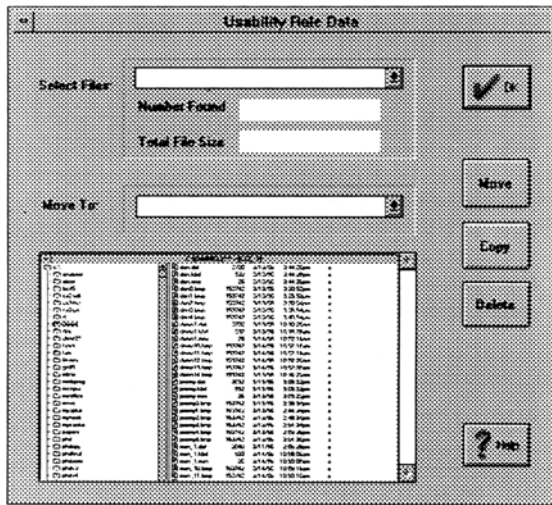


Figure 3 UI data transfer or backup window.

Figure 4 shows at the 30 minutes time, the user's finished a single task and the keystroke statistics and the mouse clicks. Again, the time interval can be set by evaluator. The Arrow buttons here are to switch the time forward and backward. As you can see, that there are not many different variety of keys that the user used. This is because the test software is an OO Diagramming Tool which is a piece of modern software, that should not need too much keyboard use, rather it should use the mouse.

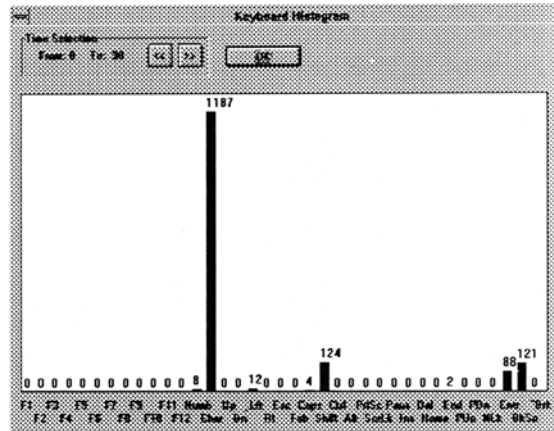


Figure 4. Keystrokes for a single task.

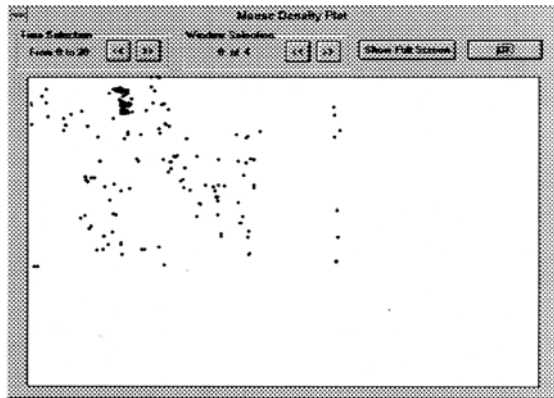


Figure 5. Mouse click density for a single task.

Figure 5 shows the total mouse clicks on doing one predefined task, and finishes within 30 minutes. From the mouse density, we look for regions where the number of clicks is largest, and search the relevant menus or buttons or any widgets which has caused the user difficulties. The Arrow buttons on Window selection allows the evaluator to switch several window images to match the mouse clicks.

A multiple comparison is shown in Figure 6. The user has to use the same package and enters the four diagrams during four different dates and times. The number of Ctrl, Shift and BackSpace ,keystrokes reduce as time goes on.

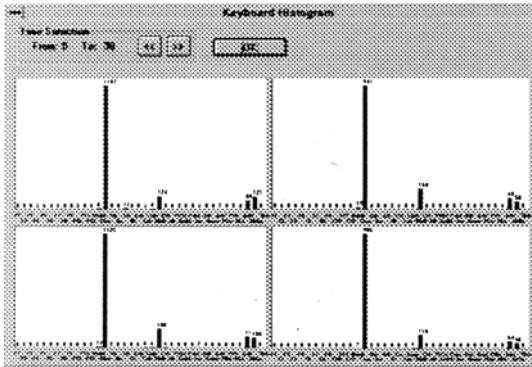


Figure 6. Multiple window comparison of keystrokes.

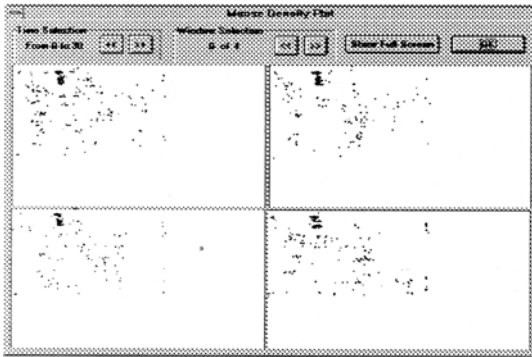


Figure 7. Multiple window comparison of mouse clicks

Figure 7, shows the user entered the four conceptual diagram using OO tools at different times. The mouse clicks show that the user's skills have improved and the mouse clicks density is getting lighter.

Because we capture each window opened. So we can match the mouse clicks with each particular window or menu or button. Figure 8 show that the OO Tool main window image is brought up. The database shows that there are 182 mouse clicks on menu called "Drawing". The database stored the exact number of clicks on a particular window, or a menu, or a button.

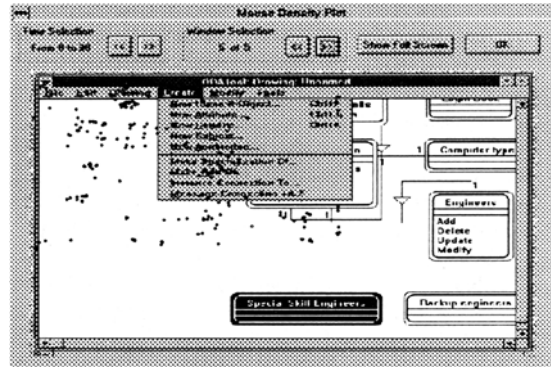


Figure 8. Match mouse clicks with menus/ sub-menus.

Figure 9 display the mouse travel pattern. Multiple windows can be shown comparing different users doing the same tasks, or one user doing one task on different software packages.

Figure 10 shows the menu used for playback users operation or actions. It is like a video recorder. The playback speed can be set to *fast mode*, which is usual if the user used software for two hours, the playback time in *fast mode* is about 10 minutes; the *slow mode*, is the speed at which the user actually worked on the computer and did the predefined tasks. This is a better option than with video, since one can exactly see what the user clicks on. This is not clear in a fast speed playback on video.

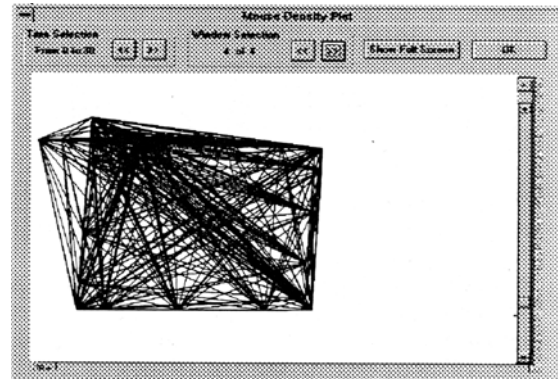


Figure 9, Mouse travel pattern.

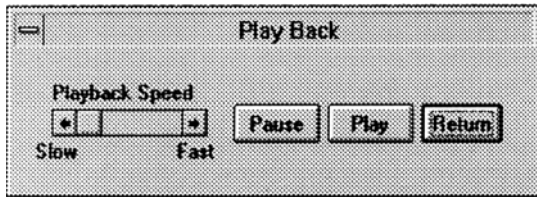


Figure10. User action playback.

Some software utilities such as Word Macro or Excel Macro or Macro-recorder allow a replay of keystrokes, but not the mouse clicks or movements, for a single window or a single application only [Sybex 1993] AUS does not have these restrictions and is capable of playback for multiple windows and multiple applications.

Figure 11, we show the overall figure of merit using fuzzy system approach [Chang, Dillon & Cook 1996].

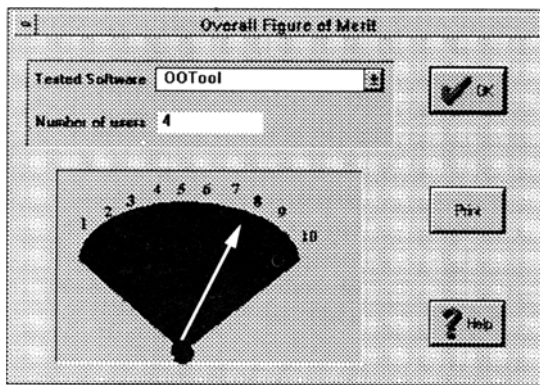


Figure 11. Overall figure of merit.

The ASCII database file for keystrokes can be viewed by any Word document The keystroke statistics for all the 256 virtual keys can be recorded in AUS. In the future, it is intended to include the extended keys, which is another 256 keys, such as Ctrl-Shift-A, etc. All statistical data are saved in ASCII format, which can be viewed in any Word document. Figure 12 shows the example in Figure 4 shows that in the first 30 minutes, there were 705 mouse clicks.

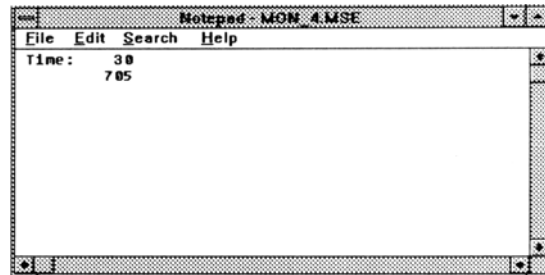


Figure 12. Total mouse click in a single task.

This prototype AUS can also record each window the user opened and number of keystrokes in this window. If this window is opened again and again, it calculates the number of times each window is opened. This is shown in Figure 13.

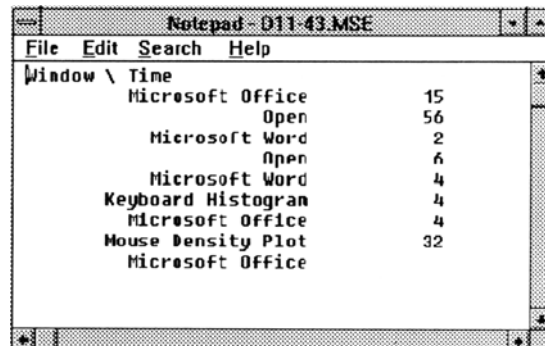


Figure 13. The number of times each window is opened.

The number of clicks is listed in 30 minute intervals. This time interval can be set. The size of the keystrokes database file is a 500 bytes ASCII file for a 30 minutes session and that of mouse density file is 20 bytes to 30 bytes, and mouse clicks and timer and window pop-up information is about 2k-3k bytes. The size of the executable code for the prototype of AUS is only 620K. The dynamic link library that we created which is used at run-time, and does all the major work, is only about 158K. This DLL is designed to be compatible across different platforms for C++, such as X window environment.

5. RECAPITULATION

In this paper, we presented a Computer Aided Usability Testing and Evaluation Software Tool. This overcomes a number of the difficulties associated with more manual methods of Usability Testing.

The AUS is able to permit replay of user actions such as mouse movements, screen opening and menu selection etc. for multiple screens and multiple applications. Some software utilities such as Word Macro or Excel Macro or Microsoft Macro-recorder allow a replay of keystrokes, but not the mouse clicks or movements, for a single window or a single application only. Also, there are many restrictions required when running such tools, such as closing all un-used windows, not moving or dragging a window around or re-sizing the window, not being able to use the mouse, etc [Sybex 1993]. AUS system should be capable of replaying for multiple window and multiple applications.

Note that, several tool developments [Eldh 1996] that are used for keeping track of keystrokes information are for: (1) debugging purposes and, (2) repeated tasks or demonstration

The debugging process uses the recorded key information to determine bugs or errors in the system. The Macro uses the key-in sequences for repeated tasks, such as automatic re-configuration of serial ports to specific settings for use with a variety of external hardware, or to change the screen colours from one set of saved colours to another, or the playback function can be used for demonstrate some flow of presentation. Most Macros are built into the application software to display a sequence that a user has pre-typed information within the software itself. Such tools are not capable of being used or run with any other commercial software. In her survey and evaluation [Eldh 1996], she pointed out that the Tool for use in Debugging and Software Testing is not mature and causes the difficulties for Industry Engineers in in-house software development. The software that we have developed has a totally different purpose and goal and can be used for User Interface Testing as well as Software Quality Testing. It runs with any software or application, tool and prototype, and is capable of

analysing the interaction between multiple windows and multiple applications and between the user and the system at the same time for the purpose of Usability Testing and Evaluation.

REFERENCES

- [1] Bevan, N. & Macleod, M (1994) "Usability measurement in context". *Behaviour and Inf. Tech.*, pp132-145
- [2] Chang E, Dillon T.S., Cook D. "Measurement of Usability of Software Using a Fuzzy System Approach" Invited paper *Proc. of 8th Int. Conf. on Software Eng. and Knowledge Eng.*, pp69-76, Lake Tahoe, USA 1996.
- [3] Dillon, A and Maquire, M. (1993) Usability Measurement- Its Practical Value to the Computer Industry. *Proc. of ACM/IFIP Human Factors in Comp. Sys., INTERCHI 4,93* pp 145-148
- [4] Dorward, F (1994) "The advantages of portable Usability Testing" *Proc. of OZCHI'94* pp125-126
- [5] Eldh, S. (1996) "The Challenge of Test Automation- a Process Change" Research report, Ericsson Hewlett-packard Telecommunications AB, Sweden.
- [6] Molich, R and Nielsen, J. (1990) Heuristic evaluation of User Interfaces *Proc. of ACM Human Factors in Comp. Sys. CHI'90 4*, pp249-256
- [7] Nielsen, J. and Philips, V. (1993) Estimating the relative Usability of Two Interfaces: Heuristic, Formal, and Empirical Methods Compared. *Proc. of ACM/IFIP Human Factors in Comp. Sys.*, 93 pp214-221
- [8] Nielsen, J. (1993) Enhancing the Explanatory power of Usability Heuristics. *Proc. of ACM/IFIP Human Factors in Comp. Sys. INTERCHI'93 4*, 93 pp152-158
- [9] Rowley, D.E., "Usability Testing in the Field: Bringing the Laboratory to the User" *Proc. of ACM/SIGCHI Human Factors in Comp. Sys. CHI'94* pp252-257
- [10] Thimbleby, H. (1994) Formulating Usability. *SIGCHI Bulletin APRIL 94* pp59-64
- [11] Treu, S. (1994) "User Interface Evaluation -A Structured Approach" pp 175-195