EVALUATING EVALUATION: AN EMPIRICAL EXAMINATION OF NOVEL AND CONVENTIONAL USABILITY EVALUATION METHODS

Jonathan Crellin and Jenny Preece

People And Computer Interaction Systems Research Group (PACIS), Computing Department, Faculty of Mathematics, The Open University, Walton Hall, MILTON KEYNES, MK7 6AA.

Software designers and software users have different views of software. Designers see all the parts that make up a system, and which are usually hidden from the users, who only see the interface. Conventional usability evaluation methods strive for objectivity in their measurement of user/system interaction, yet it is often the subjective experience of using a system which is important. Whiteside et al (1988) propose contextual research as a way of getting hold of this type of data, but a number of problems present themselves. It is difficult to observe users in context without intruding and altering the nature of the interaction, especially as much of the interesting parts of an interaction are internal to the user, and not available for direct observation. Methods such as 'thinking aloud' verbal protocols (which may make such phenomena explicit) are both intrusive, and likely to alter the phenomena being observed.

The PROTEUS method represents a potential solution, midway between formal, empirical methods (for example measuring behavioural performance on bench mark tests), and direct observation of users in their normal environments. In the PROTEUS method a number of direct observations of user behaviour are made by a shell program, which controls and monitors the presentation of a number of prototypes for a system. Questionnaire data is collected interactively. Users provide their own evaluation of the systems through a process of construct elicitation, using an automated repertory grid program (the Construct Elicitation System). In this way both objective, behavioural data is gathered, as well as subjective evaluative data elicited from users in an open ended manner.

During recent co-operative working with an industrial partner, the PROTEUS method was evaluated with a number of other usability evaluation techniques. Keystroke models of the stimulus systems were prepared. A keystroke log of the interaction as recorded. Video records of the interactions were kept, and survey data was collected. This material has allowed a comparison of the different methods to be made, and prescriptive recommendations as to their use to be drawn up.

1. INTRODUCTION

1.1 Designers and Users

Software designers and software users have different views of software. Designers see all the parts that make up a system, those parts which are usually hidden from the user, who only sees the interface. Conventional usability evaluation methods strive for objectivity in their measurement of user/system interaction, yet it is often the subjective experience of using a system which is important. Grudin (1989) refers to problems of formal approaches. Grudin defines a formal approache as one which concentrates for its analysis on the form of the interface in isolation. Focusing on the interface in isolation is very attractive proposition (perhaps especially to system designers) as the interface is always readily available, and never complains whatever you do to it (unlike users). Formal analysis of the interface in isolation holds the possibility of automation of some aspects of evaluation. It is possible to assess an interface before it has anything other than a theoretical existence. Unfortunately a focus on the structure of the interface in isolation, without a consideration of users and their tasks, often leads to interfaces which are difficult or unpleasant to use.

1.2 Empirical Methods.

Empirical approaches are those which test the interface in an experimental (or more often semi-experimental) setting. They involve studying users using the interface, or prototypes of parts of the interface. Often formal tasks are set for users, and user performance on different versions of the prototype are compared. Occasionally more than one solution for a particular aspect of an interface are prepared as prototypes, and performance on these are compared. Early

prototyping of the interface, and frequent iteration remain central to such techniques. This approach involves defining measurable behavioural goals before testing takes place. These operationalised goals form the yardsticks by which progress to a final interface design is measured. These goals are usually set and defined by an expert, usually a human factors engineer, based on ergonomic and cognitive science knowledge about how people react to interfaces. In the empirical approache the criteria for design are no longer focused on purely the formal aspects of an interface, but also include observation of people using the interface. In the empirical approach the 'agenda' of interface evaluation issues remains clearly in the hands of the evaluators. Gould et al (1987) provide a clear account of such a procedure for developing a complex interface.

1.2. Contextual Research

Empirical evaluation has been very successful in designing usable interfaces, but as the gap between the knowledge and experience of the evaluator and the knowledge and experience of the user increases, there is an increasing sense that evaluators may not always be addressing the right issues. Particularly the aesthetic aspects of interface design are less adequately catered for in empirical evaluation. Such aspects will undoubtedly be of increased importance as the gap between the client (who orders the software) and the user (who actually has to use it) narrows, with the increased use of micros, and off-the-shelf software. This suggests that the important data that needs to be collected during evaluation is that related to the experience of using the interface. Whiteside et al (1988) propose contextual research as a way of obtaining this type of data. Contextual research is an attempt to understand the use of the software in ecologically valid situations. It involves replacing the empirical positivist approach which views the user as another object in the human computer system, with a user as subject focus. There are a number of methods which are associated with contextual research. These involve the collection of verbal thinking-aloud protocols, direct observation (usually video recording) of subjects using software in a natural setting, and open ended, or ethnographic interviewing. Whiteside's focus is on recording experience as it happens, to avoid such phenomena as post-hoc rationalisation. This suggests the pre-eminence of direct observation and verbal protocol recording.

Unfortunately the methods associated with the contextual approach present a number of problems. It is difficult to observe users in context without intruding and altering the nature of the interaction, especially as much of the interesting parts of an interaction are internal to the user, and not available for direct observation. Methods such as 'thinking aloud' (TA) verbal protocols (which may make such phenomena explicit) are both intrusive, and usually alter the phenomena being observed.

1.4. Another Approach

If the process of using an interface provides information about the usablility of the interface, then it appears that techniques derived from knowledge elicitation may be appropriate, Briggs (1987). Although the user is not an expert in interface design, he is certainly an expert on his own experience. Techniques derived from knowledge elicitation can extract experience knowledge.

In knowledge elicitation questioning after the event forms the basis of many techniques, (although recording of actual problem solving is also used). Methods of collecting experience can be based on questioning after the event. The disadvantage of these methods is that post-hoc rationalisation may occur and conceal the actual processes that took place. On the other hand collecting data after the event is much less likely to interfere with the processes as they take place. Open ended questioning involves asking none-leading questions of the user. It is not a particularly straightforward skill to acquire, since it requires the interviewer to ask questions in a way which is not influenced by the replies of the interviewee. Computer presentation of questions can be more appropriate than a human interviewer for presenting open ended questions. The interviewee may be less influenced by the interview when typing into a machine (with the possibility of erasing text) than when answering a human interviewer. Ethnographic techniques involve asked the interviewee questions in the interviewee's own terms. This approach originated in the context of social anthropology, and would usually involve a researcher going native in order to understand the intricacies of the experience of life in another society. In the context of human computer interaction an ethnographic based approach will involve asking questions using an interviewees own terms (for example "What do you mean by a 'sticky' interface?").

Open ended and ethnographic approaches are relatively suitable for automation. In essence they involve repeatedly asking the interviewee the same question, or variations on the same question, collecting the replies, and then feeding these back to the interviewee for clarification and elaboration.

The repertory grid Kelly (1955), which has been used in various automated knowledge elicitation tools, Boose (1985), appears to provide a suitable approach to the extraction and representation of this experience knowledge. Similar techniques have also been used in the evaluation of sets of complex artifacts in marketing, Stewart et al (1981), and in evaluating architecture, Honikman (1976) In this study the PROTEUS tool was used. PROTEUS (Crellin (1990)) allows the on-line collection of repertory grid data through the Construct Elicitation System.

1.5. A classification of evaluation approaches.

Evaluation approaches can be distinguished partly by the different methods of data collection employed, the situation in which data is collected, and also by the different ways that the data is analysed. For example the empirical based

approach and the contextual approach both may collect verbal protocols. The empirical approach would then classify the data according to a pre-defined categorisation, probably based on a cognitive psychological abstraction. The contextual approach would be more open to the natural structure of the verbal data, and seek a method of classification according to categories which arose naturally from the data.

This study involves the collection of data by a variety of means, employing methods associated with formal approaches, empirical approaches, and contextual approaches, (including a novel experience elicitation method). Although it is usual to define the different approaches of interface evaluation as distinctly different, the boundaries are usually rather less distinct. In this study the data was collected in a semi-formal fashion, midway between a contextual and an empirical laboratory setting. The laboratory like features of the setting were that the task examined was not entirely real, subjects were asked to explore the system until they understood how to operate it. However formal benchmark tasks for subjects were not set during the early stages of the study. A subsequent study of the interfaces using prescribed benchmark tasks was made to compare actual performance and the keystroke analaysis prediction of expert performance. Subjects were not using the software in their normal working environment however they were allowed to consult with other people if they got stuck. Data was collected from a TA verbal protocol, video observation, keystroke logging, and using a knowledge elicitation technique. Formal methods used were a prediction of expert performance from keystroke analysis, and a prediction of complexity and consistency from a BNF analysis of the different stimulus interfaces. This paper reports on the strengths and weaknesses of the methods employed for data collection, rather than on actual conclusions about the stimulus interfaces used.

2. METHODOLOGY

2.1. The Task

The evaluation task is calculating the cost of a telephone call, using different versions of a prototype application. Each user is required to use all the different versions. Most people are familiar with telephone usage so that the task domain should not be unfamiliar to any user. However the method of call charging employed by BT is not particularly obvious, and so many people will not be familiar with exactly how telephone charges are calculated. BT charge calls by measuring the number of units used in each call. Each unit costs 5.06 pence. Some calls are more expensive than others, due to the distance called, or the amount of equipment used. For example overseas calls are typically much more expensive than inland calls and calls to mobile phones are particularly expensive. BT cope with this by allotting different amounts of time per unit to each distance band, hence a Local call may use one unit every 360 seconds, and a call to Hong Kong may use a unit every 3.5 seconds. Additionally BT vary the amount of time per unit according to the time of day, so that busy periods are more expensive. Inland calls usually have three charge rates, cheap, standard and peak, however some overseas calls have only one charge rate, and many have just two. British Telecom, (1987).

In this study the functional part of each version of the prototype is clearly separated from the interface. There is considerable scope for variation in the way a user enters call parameters (distance and time of day). Also the way a user calls the different application functions provides considerable scope for interface variation. Each version has a radically different type of interface. The underlying function of each version is identical. It is simply a timer which increments the cost of a call by the cost of a unit, as each unit is completed.

The different interface versions were implemented on a Mac Plus computer using MS BASIC, which provides a rapid prototyping environment with full access to the Macintosh ROM Toolbox. Where relevant the final interfaces conform to Macintosh interface guidelines, and the final package of prototypes, PROTEUS, CES, and Mac system files fit on a single Macintosh boot-up disc.

The following interface issues were explored in the different prototypes.

* Depth versus breadth in menu structure.
* The use of icons.
* Mouse versus keyboard input.
* User event based versus system directed.
* The amount of information displayed on the screen at any one time

2.2. The PROTEUS shell

The PROTEUS shell represents an integrated environment in which interfaces, evaluation and help systems can be easily accessed. System usage data is collected by the shell. The shell also presents on-line questionnaires and rating scales when programmed to do so. More details of the system are available in Crellin (1990).

2.3. Formal Measures

Keystroke level recording was performed by using a Macro writer application which collects a real time record of system usage. This recording is suitable for replay in synch with the video data, and also provides a text description of the actions performed. This text description is suitable for statistical analysis (for example in evaluating the relationship between predicted and actual 'expert' level performance). This type of analysis can be performed automatically

2.3.1 BNF description

A BNF description of each interface of the system was developed along the lines of Reisner []. The BNF description was used to derive two metrics measuring 'string simplicity' (how many actions in a task) and 'structural consistency' (are semantics reflected in syntax) which are described more fully in the paper. This leads to a number of predictions, based on an elementary cognitive model. These were compared against experimental data and user's subjective opinions.

2.4. The Construct Elicitation System

The Construct Elicitation System is an on-line system for eliciting and recording discriminating constructs between the different interfaces. It is described in detail in Crellin (1988, 1990).

2.5. Subjects

The subjects chosen for the experiments came from industry, having a broad range of backgrounds and experience. Clerical, technical and managerial users were represented, with experience of computers ranging from novice to very experienced.

2.6 The set up for the video protocol collection.

Instructions to subjects. To verbalise "...what you want to do next...". This collection of data is at the intention level, and can therefore be compared to their actual behaviour, highlighting areas where the users misunderstand the systems. Users were asked to verbalise their thoughts: to explain what and how they were going to tackle the tasks given them, and also to communicate their emotional reactions to the interfaces.

3. RESULTS

3.1 Video Protocol Data

The video protocol data was analysed in a bottom up manner, in sympathy to a contextual research approach. One of the clearest modes of categorisation that arose from the data was a classification of the type of error made. Slips were errors where the individual mistyped a command, or miss-selected a button or menu item. Mapping errors were where the individual did not map the systems commands to the underlying task. Task errors were where the individual had an inadequate understanding of the task. Error classification arose from the availability of intention statements from the subjects verbal protocols, and the replay of system events in synchronisation. The verbal protocol data varied considerably between subjects, both in the number of comments, and what the focus of the comments were. Some subjects simple reported their physical actions, whereas other verbalised their intentions, and overall goals. Obviously the latter were more useful than the former.

Initially the video data appeared to add little to the analysis, however a number of observations arose exclusively from the video data. Inter-subject variations in body language suggested difference in attitude to the task, and during execution of the task the strength and focus of attention of the subject could be observed. Additionally, the clarity of screen display was also apparent from the posture of subjects.

3.2 The Construct Elicitation Data

The data from the Construct Elicitation System has been analysed using the FOCUS algorithm, Shaw (1980), Jankowicz and Thomas (1982). Results from the analysis are displayed as two binary trees per subject. The binary trees show the similarity matchings for the constructs elicited from subjects, and for the interfaces as the subject saw them. In this study the two menu interfaces which required selection by typed number are always seen as very similar, even though their appearance on the screen is very different, and the depth of the menu structure is quite different. One of these menu interfaces is very similar in appearance to a menu interface where selection of the menu items is by ticking checkboxes, however the similarity between these two interfaces is not so universally observed. This observation suggests the relative importance of the input device, over more abstract features of the interfaces such as menu depth.

The construct trees make more apparent the number of distinct ways an individual is using to distinguish between the different interfaces. More details of the analysis methods for the CES data can be found in Crellin (1990).

3.3 Discussion of Results

For usability evaluations to be viable within the working environment of a small company such as Brameur it must be relatively cheap; that is, be quick to carry out and not require expensive specialist equipment or expertise. It should also not disrupt employees normal working patterns unreasonably. In addition the information derived from the assessment must provide a good overall picture of the usability of the prototypes in terms of their use within the company. In particular there must be clear indications, which designers can act upon, of where usability problems occurred and what improvements are needed.

The suitability of the methods described in the previous section can, therefore be examined with three questions in mind:

1) What equipment, time and expertise was required to collect the data and how much disruption did employees incur?

2) What kind of data was collected, how was it analysed and what equipment, time, and expertise was required for the analysis?

3) What were the main findings from the data?

Tables X.1 to X.6 summarise the results and observations relating to each of the questions.

	Setting Up			Running		Comment
	Equipment	Time	Expertise	Reliability	Disruption	
Video	Portable Video Equipment	1 hour.	Positioning and co-ordinating with interaction log.	Good.	More than desirable.	Initial expense or hire fees.
Audio	Integral with video, although external micro-phone essential.	20 minutes	Checking sound quality.	Good.	More than desirable also 'thinking aloud' is disruptive to colleauges.	initial expensee or hire fee.
Keystroke Log	Software in the users system so nothing extra. System requires additional RAM.	5 minutes	Needs co-ordi-nation with video.	Poor, kept run-ning out of memory without warning and recording was lost.	Much more than desirable, but more when working prop-erly.	Could be made automatic with no efect on users, very low cost.
System Usage Log	Software on users system.	None.	Automatic.	Excellent.	None.	Low cost.
Construct Elicitation System	Program called in normal way by user.	None.	None.	Excellent.	A seperate ac-tivity which must be carried out by user. No disruption dur-ing evaluation task itself.	Low cost.
Keystroke/BNF	None.	None.	None.	N/A	N/A	Time and ex-pertise required to decompose task.

Table X.1: Analysis of data collection techniques

Table X.1 shows that, in our opinion, video recording was far from unobtrusive. In fact, we remain to be convinced that contextual research, in which video is used, is as ecologically sound as the overall philosophy suggests it should be. Some disturbance to the normal working environment is seems to be unavoidable.

	Data Type	Analysis	Equipment	Time	Expertise	Comment
Video (Observation)	Record of users body language. (Qualitative).	Differences noted in relation to events in interaction (Plus log and verbal protocol).	System proto-type. Good definition cam-era which will operate in nor-mal room light-ing. Good play-back, pause and search facilities.	X2 per record-ing (approx 1 hour per user).	Previous experi-ence helpful.	Sparse source of data, but valuable in conjunction with verbal protocol of interaction.

Audio (verbal protocol).	Protocol of users thoughts and actions. (Qualitative).	User perceived difficulties, and user intentions noted.	System prototype. Directional microphone, recording onto audio track of video.	X2 per recording (approx 1 hour per user).	Previous experience helpful.	Thinking aloud is difficult for users. Video and S.Log provides context.
System Keystroke Log	Record of all keystrokes, mouse actions. (Qualitative but can be interpreted quantitatively).	Problems noted. Timings can be taken and metrics applied. Number of types of errors recorded.	System prototype. Appropriate monitoring. Ideally real-time playback, and text description of events should be available.	Depends on nature of analysis. Qualitative analysis X2 per recording.	Intimate knowledge of prototypes being tested.	Rich and detailed data about user behaviour. Less useful without intentional context provided by audio and video.
Construct Elicitation System	Numerical ratings data, textual construct labels.	Computer aided cluster analysis. Graphical display of analysis.	Several prototypes of system. Appropriate elicitation software.	Varies.	Experience of technique.	provides a stand-alone analysis, which can provide a user interpretation of other data.
System and Usage Log	Quantitative record of time spent on each prototype.	Supplements other data.	Several prototypes of system. Appropriate software shell.	Short amount of time.	Little required.	
Keystroke and BNF	Quantitative helps determine possible expert performance.	Formal using prescribed model	None		Previous experience useful.	

Table X.2: Analysis of data analysis techniques

An important conclusion that we drew from our data analysis experience was that video recording, audio recording and interaction logging are individually impoverished forms of data collection. However, when analysed in conjunction 'the sum is far greater than the individual parts'. The data becomes very rich and this justifies the time, expense and disruption to work that occurs during its collection.

The data analysis reports for all five users are too long to include here. However, tables X.3 and X.4 show summaries of the findings for two users. From these two examples the information obtained from the different kinds of data is obvious. Notice particularly the information obtained from the video records of the body language of each user; this which was later supported by audio data in the case of user AA. Also notice the level of detail in the interaction log for AA using the command language interface. From this log we can see that AA did not know the syntax of the command language nor how to get the 'help screen'. Three distinct problems can be detected: incorrect use of the delimiter; lack of understanding of paramenters and the syntax for setting them; confusion between the use of the underscore to set parameters and hyphens in parameter names. A number of typing errors are also apparent.

From the rest of AA's data we gain a clear picture of how his confidence grew along with his knowledge of the task until he understood it completely and became bored. AA's data, like that of most of the subjects, showed that the command language interface had poor usability and that there was little to choose between the others; possiblely because the underlying functionality of the stystem was very limited. The logs of the usage also supported these findings, with most subjects spending longer on the command language interface or not completing the task.

The CES data, however, indicated distinct differences in the way that users themselves viewed the interfaces even though they appeared to perform on all but the command interface with similar competence. Consider, for example, the total range of data available for each of the two subjects.(ADD DISCUSSION IN HERE)

Observation (Video, Audio, Interaction log).	Usage Log	Construct Elicitation System

Checkbox	Body language stiff. User doesn't understand task requests help.	Used twice: 186 and 108 seconds. Total 294 seconds.	
Command Language	Syntax causes problems, omits delimitter, reads error message, adds delimiter. Doesn't set parameters, comment "I havn't got a clue what I am doing". Eventually find the HELP screen. Works out how to set parameters. Confuses underscore with hyphen. Does not complete the task.	Used twice: 654 and 199 seconds. Total 853seconds.	
Icon	Now understands the task well. Puzzled when the checking on the desk icon returns him to the shell program. Disturbed by macro breaking down.	Used twice: 110and 75 seconds. Total 185 seconds.	
Menu-A	Complete quickly. Now understands task and how to use menus so no challenge.	Used twice: 176 and 99 seconds. Total 275seconds.	
MenuB	Decides to experiment with changing the order in which the parameters are set.	Used twice: 153 and 77 seconds. Total 230seconds.	
Dialogue	Didn't complete the task.	Used twice: 184 and 52 seconds. Total 236 seconds.	
Macintosh	Likes this one "When you go to rate you can see which ones are available".	Used twice: 158and 72 seconds. Total 230 seconds.	

Table X.3: The main findings from the data for user AA (use Phil's? but still need the CES data.)

An interesting point to notice in the CES data is how the users' apparent understanding of the way that this data collection technique works seems to affect the quality, though not necessarily the content, of the data that is collected. User BB (i.e. Mike) for example, selected concepts with well discriminating poles and to describe these concepts very clearly. User AA CES data by comparison shows less discrimination; probably because he was less sure about how to do this task. In addition to the data relating to the use of the prototypes, we performed keystroke level and BNF analyses for each of the prototypes. Tables X.5 and X.6 respectively show these results.

This technique [keystroke analysis] Card et al (1988) allows one to predict the time it takes an expert user to perform a given task on a comparison of predicted time it takes an expert user to perform a given task on a computer system, and can be applied at the design stage. Applying this approach yielded the following results:

Comparison of predicted times (s) for interfaces:

Mac	Icon	Menu A	Com Lang	Menu B	Dialog	Check Box
43.4	42.7	32.8	30.8	30.6	23.7	21.4

Users were give two trials with the TTS. The data for this last trial has not yet been fully analysed, but assuming that users can now be considered fairly expert at using the system it would be interesting to see to if there is any correlation between the time spent by them on the system and the above figures. Certainly, from the above it would appear that the check box interface would be an expert's choice.

Applying Reisners metric for string simplicity to the stimulus interface results

Dialog	Check Box	Comm Lang	Icon	Menu A	Menu B	Mac
7	5	13	6	5	5	6

Following Reisner's assumptions this would imply that the Command Language would be the most difficult to learn and the menu interface the easiest. From the available data it would appear to be true that the Command Language

interface is the most difficult, not only to use, but also to learn to use. This is, however based only on two sessions with the system, and it is not clear whether this is enough to assess learnablity.

The other metric (structural consistency) is perhaps not applicable to the stimulus interfaces because of their limited functionality. Examination of the BNF rules shows that all interfaces have a high degree of structural consistency.

4. DISCUSSION

4.1. The strengths and weaknesses of the different methods of analysis

Jonathan, I've done a bit on this but it needs more work and a bit of pepping up!

In this paper we have described a case study evaluation of the use of different evaluation techniques. The case study was carried out with the help of users from a small software company and the data collection was done on their premises and integrated in with other tasks which the emplyees perform during their normal working day. We have therefore argued that, although not a pure contextual evaluation, our study has far more in common with the philosophy of contextual research than with scientific and engineering paradigms. Many data collection and analysis techniques can be used in either paradigms; it depends on how they are applied, where the study occurs and the role and relationship of users and researchers. Indeed we have discussed how observational data collected within a contextual paradigm can be analysed in a quasi-experimental or empirical way.

Our aim in this study has been to evaluate a number of well-known data collection techniques, a novel ethnographic evaluation tool, and two analytical techniques within the context of a small company. As a focus for this evaluation we have examined the usability of seven different protoype interfaces.

The conclusions that can be drawn about the interfaces are that, on the basis of the video, audio, interaction and usage logs, it is clear that the command language interface has very poor usability - all the users had difficulty as we expected. However, there is little to discriminate between the usability of the other interfaces at the level of analysis that we have performed. More detailed analyses of numbers of errors will be performed but it is unlikely that many differences will be detected. This leads us to suggest that providing users understand the task that is to be performed and know how to map this onto the design of the system, then carefully designed interfaces of any type are likely to have similar usability. Although, this is a speculative suggestion and needs to be examined further, it is in accordance with a report by Whiteside et al. who carried out a similar study of a number of fully implemented systems. (paper in interacting with computers). According to the analytical analyses, however, it appears that expert users would perform better on (ADD). However, as many authors have pointed out (e.g.), it is questionable whether performance metrics for 'time to complete a task' should be given the deciding vote on which form of interface to be adopted.

Given the lack of discrimination between the techniques, the deciding factor surely ought to be users' own opinions of the systems. In the next part of our study we shall replay our data and invite the users to analyse it with us in a similar way to Wright and Monk's participative evaluation (Wright and Monk, 1989) and the contextual research methods of Whiteside et al. (1988) and Holtzblatt (1989). However, the CES data falls within that same general and it showed that ADD.

From carrying out this evaluation of the interfaces our own opinions and those of the Brameur employees suggest that if ways of collecting a video record could be be improved so that data collection is less intrusive and the interaction logging software perfected so that it worked smoothly, these techniques would be valuable for assessing either the usability of prototypes or the effects of tailoring software within the normal work environment of small companies. The intrusiveness of the video recording could be reduced by using lower quality equipment without a tripod and accepting a recording of lower technical quality. Since the video is only really useful for providing context, giving information about body language and making data analysis more palatable this reduction in quality would be acceptable. The 'think aloud' technique is not suitable for use in work situations since it disturbs colleagues and is embarrassing for users. Sound is valuable for collecting details of users requests for help, comments and discussions with other colleagues. It is probably worth investing in a sensitive lapel, or high qualilty remote microphones, which would be even more preferable. In addition, the video, sound recording and interaction logging must be easy to synchronise. Since the system usage log is unobtrusive, automatic and provides information about timings it is worth collecting even though the data is of limited value.

The CES system is easy and cheap to run as a researcher is not required to collect the data and the first part of the analysis is done by the computer. However, some experience is necessary to interpret the output. ADD

One general conclusion from this case study is the obvious strength of mixed methodologies providing that disturbance to users is minimal. Table X.7 provides a summary of the strengths and weaknesses of the techniques used in this evaluation.

Data Collection		Data Analysis	
Strengths	Weaknesses	Strengths	Weaknesses

Video	None	Intrusive. Recording equipment required	Provides context, can be used to stimulate post-hoc user comments.	Playback equipment required
Audio	None	Thinking aloud is very intrusive	Provides context.	Playback equipment required
Keystroke logging	No additional eqipment needed. Very unobtrusive.	Present software not very reliable.	Good real time record of user behaviour. Adds informatiion to video and audio recordings. Text description a useful resource.	Can be time consuming to anlayse.
Usage Logging	Unobtrusive	None	Useful extra information	None, but rather impoverishe data on its own.
Construct Elicitation System	Can be used by users on their own.	Users need to understand construing if high quality data is to be collected.	Initial quantitative analysis carried out by computer. Graphic representation of data provides basis for qualitative analysis.	Sometimes difficult to work out what users really mean. Some experience is needed to interpret data.

Table X.7: Summary of the strengths and weaknesses of the techniques used in the case study.

In the remaining part of this work we shall discuss the observation data with the users and also inviting them to comment about the evaluation process itself. The next phase of the work will be in two parts. In one part we shall focus on the development of observation techniques and forms of data analysis suitable for use in and for small companies. The other part of the work will be concerned with making the construct elicitation task itelf more obvious to users so that it is easier for them to specify concepts with their poles. This should, in turn, result in better data being collected and easier analysis. Having completed these phases of the work we shall then evaluate the use of the techniques by a number of small companies.

The command Language interface offered more information about problem solving in the task, the menu driven interfaces were too immediately obvious. Virtually all errors were made in use of the command language interface. Unfortunately the command language interface was probably the only solution to the interface problem that almost certainly would not be used.

4.2. The advantages of mixed methodologies

The verbal data and the system logging data were of little use without each other. The video data appeared to add little to the formal analysis, but added information about general attitude that would not be available from a verbal protocol alone. Additionally the video data made the analysis of the verbal data less ambiguous. However it is possible that the video data may distract observers from the contents of the verbal protocol on some occasions.

4.3. The unique features of the Construct Data.

It was quite clear from the video protocol and system logging which was the worst interface. It was not as easy to classify the remaining interfaces using this data. The formal analysis measures provided a clear ranking for the predicted performance of the different interfaces. The Construct Elicitation data also provided a better comparison of the subtler differences between the interfaces, although this comparison did not explicitly rank the different interfaces.

ACKNOWLEDGEMENTS

The evaluation of usability evaluation techniques was sponsored by Brameur as part of ESPRIT Project 1257, Muse. The authors wish to thank the members of the project for their support and, in particular, Dr. J. Hemesley, Director of Brameur and Mr. M. Kelley organiser of the Muse Project at Brameur.

PROTEUS and the CES system is part of Mr. J. Crellin's Ph. D. work which is supported by The Open University and supervised by Mr. D. Benyon and Dr. J. Preece.

We should also like to thank the employees of Brameur who participated in the study.

REFERENCES

Boose, J. H., (1985) A Knowledge Acquisition Program based on Personal Construct Theory, Int. J. Man-Machine Studies, 23, 495-525.

Briggs, P., (1987) Usability Assessment for the Office: Methodological Choices and their Implications, in, Psychological Issues of Human-Computer Interaction in the Workplace, (Frese, M., Ulich, E., and Dzida, W), North-Holland, Amsterdam .

British Telecommunications PLC, (1987) International Telephone Guide, ITG6 RES November, Collier and Searle Ltd..

Card, S. K., Moran, T. P. and Newall, A., (1980) The Keystroke-Level Model for User Performance Time with Interactive Systems, Communications of the ACM, 23, 7.

Crellin, J. M., (1988) Personal Construct Psychology and the Development of a Tool for Formative Evaluation of Software Prototypes, in: Proceedings of the Fourth European Conference on Cognitive Ergonomics, (Green, T. R. G. et al. ed.), Cambridge.

Crellin, J.M., PROTEUS: an approach to interface evaluation, this volume.

Gould, J. D., Boies, S., Levy, S., Richards, J. T., and Schoonard, J., (1987) The 1984 Olympic Messaging System: A test of behavioural principles of system design, Communications of the ACM, 30, 9.

Grudin, J., (1989) The Case Against User Interface Consistency, Communications of the ACM, 32, 10.

Honikman, B., (1976) Construct Theory as an Approach to Architectural and Environmental Design, in The Measurement of Interpersonal Space Vol 1, (P. Slater ed.) Wiley, London.

Jankowicz, D. and Thomas, L., (1982) An algorithm for the cluster analysis of repertory grids in human resource development, Personnel Review, 11, 4, pp.15-22.

Kelly, G., (1955) The Psychology of Personal Constructs, Norton, New York.Stewart, V., Stewart , A., and Fonda, N. (1981) Business Applications of Repertory Grid, Mcgraw Hill (UK) Ltd.

Reisner, P., Formal Grammar and Human Factors Design of an Interactive Graphics System, IEEE Transactions in Software Engineering, 7.

Shaw,M., (1980) On Becoming a Personal Scientist Academic Press.

Stewart, V., Stewart , A., and Fonda, N. (1981) Business Applications of Repertory Grid, Mcgraw Hill (UK) Ltd.

Whiteside J., Bennett J., Holtzblatt K., (1988) Usability Engineering: Our experience and evolution, in: Handbook of Human Computer Interaction, (M.Helander ed.), Elsevier Sciences Publishers, Amsterdam.