A comparison of usability techniques for evaluating design

Ann Doubleday, Michele Ryan, Mark Springett , Alistair Sutcliffe
Centre for HCI Design,
School of Informatics,
City University,
Northampton Square,
London EC1V 0HB, UK
Email: M.Ryan@uk.ac.city
+44-171-477-8993

ABSTRACT
We report on a series of experiments designed to compare usability testing methods in a novel information retrieval interface. The purpose of this ongoing work is to investigate the problems people encounter while performing information retrieval tasks, and to assess evaluation methods by looking at the problem focus, the quality of the results and the cost effectiveness of each method. This first communication compares expert evaluation using heuristics [15] with end user testing [24].

Keywords
Usability, Evaluation, Heuristic evaluation, Information retrieval, User interface design.

INTRODUCTION

From the time that user interfaces were no longer dictated by the underlying software, and technology became available to design computer applications with usability in mind, publications and methods, guidelines and methodologies have proliferated on how to design usable interfaces [4; 9; 11; 12; 15; 18; 19; 22; 8; 5]. Why then are applications still so difficult to use? Some reports have suggested that industry is reluctant to use design and evaluation techniques from the HCI community [2]. This is reported as being partly due to the expertise required to use some methods, and partly because of the amount of time and resources required to perform evaluation. A number of established methods require considerable expertise and effort to apply [*e.g.* 5; 9].

For example, the method described in [9] demands the generation of several augmented and general transition network diagrams, among other representations. This is unlikely to be a trivial task for the typical design professional. Some more contemporary approaches, [*e.g.* 16] require the presence of human factors specialists or evaluation experts. Also, industry tends to rely on field reports to test products, and assess new product versions. The viability of HCI in formative and summative design activities may only be accepted if a tangible improvement to current practice is offered. Therefore a method for evaluating designs and assisting usability must either be an inexpensive addition to existing practices or a means to integrate the goals of usability with those of efficient software development. This paper tests two approaches to assessing usability with the theme of swift and effective evaluation in mind. These are heuristic evaluation [15] and end user testing following Wright and Monk's guidelines [24]. We assess the strengths and weaknesses of the methods, and, by separating task specific from general usability problems, attempt to identify the kind of task or problem each optimally addresses.

The paper starts by showing how the Esprit project INTUITIVE (INTeractive User Interface Tools In a Visual Environment) interface attempts to support the cognitive effort involved in information retrieval tasks. Next we provide a brief overview of a number of evaluation techniques in current use. The next section introduces the experimental session by describing how heuristic evaluation and end user testing were applied to INTUITIVE. We assess how well each lived up to its theoretical claims and how we rate their comparative success. Included in this is an attempt to identify those usability problems which could be predicted theoretically, such as the system introducing a requirement for user action not present in the user's task model, and to distinguish faults in the users' perception of the task from general difficulties in operating techniques that might be applied to any task.

INTUITIVE AND THE INFORMATION RETRIEVAL TASK

We identify four principal subtasks which are normally performed in the following sequence: (i) navigating the dataspace and choosing the entities to interrogate, (ii) forming the query by selecting and constraining attributes from the chosen entities, (iii) summarising the results of the query to enable the user to choose a subset for full viewing, and (iv) presenting the chosen items in full. For each task we illustrate how the INTUITIVE interface provides support for the user.

Navigation and exploration

In some way the retriever has to match his search needs to the contents of the database. Frequently his knowledge of both is imprecise. The system needs to show the user what is in the database and, if possible the connections between database items. It also needs to enable the user to select entities for queries simply and rapidly. In INTUITIVE, the dataspace is represented visually by entity nodes with non-directional links showing the relations between them. See figure 1. Any entity or linked group of entities can be selected for querying by point and click.

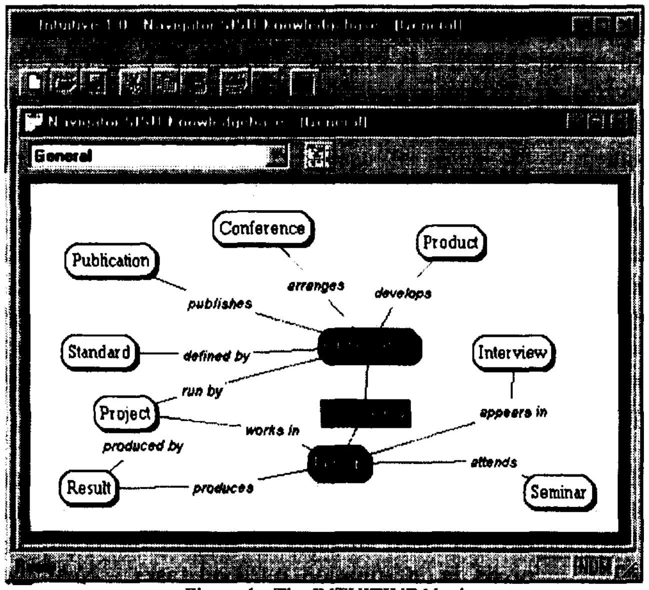
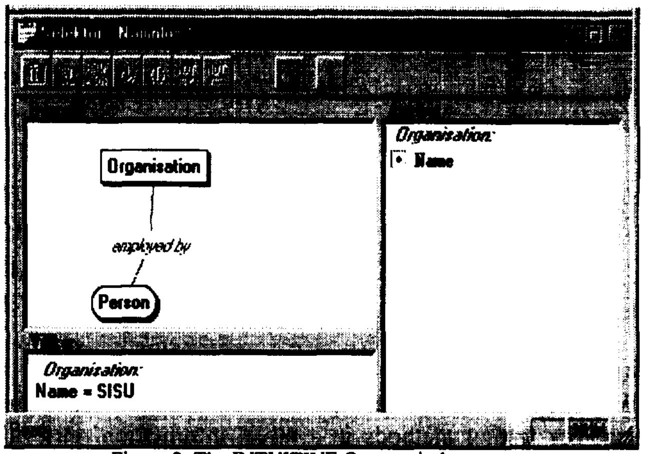


Figure 1. The INTUITIVE Navigator

Query formation

The user's precise needs have to be translated into a language the DBMS can understand, in a form that the user can understand and if necessary modify. In INTUITIVE, the DBMS language is SQL, which is difficult for the average information retrieval user to code with syntactic accuracy. Therefore, the translation to SQL is done internally, the user's interaction being entirely graphical: selection from an interactive list of attributes for the selected entity(s); the selected attributes can then be constrained by pointing and clicking. See figure 2. The developing SQL text is shown below the graphical interaction, when the user is satisfied the query is submitted from the Query>Browse menu option.

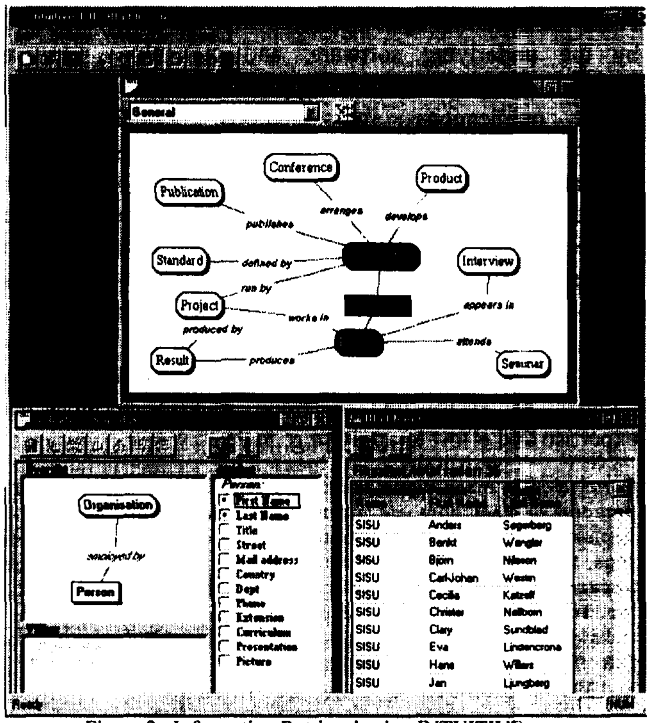


Organisation:
Name = SISU

Figure 2. The INTUITIVE Query window

Previewing the retrieved information

In all but the simplest cases, the quality and quantity of retrieved information is unpredictable. Many systems provide a preview facility to permit user selection of a subset of the items. INTUITIVE returns from its database search with a list of retrieved items, spatially linked to the query enabling the user to select and retrieve items by point and click. See figure 3.



SISU	Anders	Segerberg
SISU	Benkt	Wangler
SISU	Björn	Nilsson
SISU	Carl-Johan	Westin
SISU	Cecilia	Katzeff
SISU	Christer	Nellborn
SISU	Clary	Sundblad
SISU	Eva	Lindencrona
SISU	Hans	Willars
SISU	Jan	Ljungberg

Figure 3. Information Retrieval using INTUITIVE
The top window shows the INTUITIVE navigator, bottom left shows the query window, bottom right shows the results window.

Presentation of retrieved items

As far as the user is concerned, the items should just appear, in comprehensible chunks, displayed in surroundings appropriate to their medium, visibly connected to each other and to the originating query. In INTUITIVE, selected items for output appear in the most appropriate application - .doc files in Word, .avi files as runnable videos, with the windows arranged logically.

USABILITY EVALUATION

Overview

Method developments in usability assessment include standards and guidelines [8; 4; 19], observation and monitoring techniques [7; 23], attitude questionnaires [10], checklists [18], and techniques for expert evaluation [15; 16]. Some methods, notably performance monitoring, yield statistical data which can show where most problems occur but do not diagnose problems or recommend solutions. Most, however, are aimed at providing designers with more incisive diagnosis which can inform re-design. The quality of the information yield may be critical, given that selecting redesign solution may miss its target or cause secondary problems [21]. This paper assumes the goal of effective redesign through usability data collection.

On INTUITIVE we employed a combination of standards [8], user testing in which end users were videotaped and timed performing increasingly complex tasks on the toolset under control conditions [17], and expert evaluation in the form of Nielsen's heuristic checklist [15] by HCI expert evaluators. All the techniques were applied to finished or near-finished systems. This first communication compares heuristic evaluation with end user testing.

EXPERIMENTAL DETAILS

Nielsen's heuristics

Heuristic evaluation is intended to find the usability problems in a user interface design, in order to address them during the iterative design cycle. Nielsen [15] developed a set of heuristics to assess an interface according to recognised usability principles, such as users' language, consistency, minimising memory load. He has shown [14] that errors are found faster if evaluators are experts in the domain or in user interaction, maximally if in both. In principle, evaluators decide on their own how to proceed with evaluating the interface, but it is recommended that they go through the interface at least twice. The first pass is to is to get a feel for the flow of the interaction and the general scope of the system, the second is to focus on specific interface aspects.

In our case, we used five members of staff from the HCI design Centre at City University who were HCI experts, but most of whom had very little knowledge of any specialist aspects of information retrieval. The recommended procedures were followed: after a familiarisation session including demonstration and explanation, the evaluators did a two pass supported walkthrough using a set of seven increasingly complex information retrieval tasks, provided by us. The evaluators worked at their own pace and the whole session took a little over an hour, but no time limits were applied.

End user testing

Both to obtain statistically valid data and because end users need a structured session, the end user testing was of necessity task based. We devised data access tasks of increasing technical complexity. It has been widely demonstrated that information retrieval is an iterative, exploratory process [13; 1] so our experiments also tested how easy it was for users to modify and refine queries.

Twenty subjects, all undergraduate or postgraduate students, took part in the study. After completing a questionnaire on their level of computer, Microsoft, database and SQL experience, and being shown a short demonstration, each user was given a set of information retrieval tasks using a personnel database. The first three tasks required users to find straightforward information from the database, and the remaining tasks involved increasingly complex queries including zero results. Users worked at their own pace for up to 30 minutes. They were videotaped and asked to provide continuous verbal commentary on their thoughts and actions, following the protocol analysis procedure of Ericsson and Simon [7]. Each user then completed a satisfaction and usability assessment questionnaire, rating such items as flexibility, appeal, ease of avoiding and correcting errors, on 1-7 Likert scale. To elicit their post test memory of the task, users were asked, without looking at the system, to describe the steps needed to send a query to the database.

A de-briefing session was carried out for a sample of the sessions. This provided an opportunity for the evaluation facilitator to go through the usability problems the user had encountered.

Both task-based and performance-based analyses were performed on the quantitative data captured from the video tapes, observations and memory test, which included time taken to complete the tasks, number of tasks completed, and those correctly answered, error frequency, number of times user accessed help, and the task steps missed. Statistical analysis is published elsewhere [20], this paper concentrates on the tasks and implications for re-design.

RESULTS

Cost-effectiveness

The purpose of this research is to help users to decide on the most cost-effective way of evaluating their interface. It is our contention that simply counting and matching errors found throws little light either on the quality of the interface or the quality of the evaluation technique. It seemed worthwhile to investigate a little more deeply, to attempt to categorise errors and then see whether the different methods varied in their effect on the type of errors found. We need to look at a number of factors, for instance: clarifying the error count metric, the effort and cost needed to achieve the results, any difference in support for the developer attempting to cure the problems, and the quality of the interface at the end of the operation.

The error-count metric

As a simple count, we note that five HCI experts identified 86 usability problems, as against twenty end users who enabled us to identify 38. However, the definition of 'error' is imprecise, both as expressed by expert evaluators and as deduced (by investigators) from the end users' videos and debriefing. Boundaries between 'errors' can be fuzzy and overlapping; one person's stated error may comprise several symptoms reported by others as separate errors. Such a composite error would reduce the total count (and therefore the apparent quality of the method) but might actually throw more light on what caused the symptoms. Also, we note that the 38 problems found by end user testing were not a subset of the 86 heuristic errors, so if there had been no user testing, many genuine problems would have gone undetected. It is obvious too, that the more errors there are in an interface, the more there are to find - the important, and much harder, measure, is how *few* the technique has *missed.*

Comparative costs

Approximate comparative costs are:

- For heuristic evaluation a total of 33.5 hours; 6.25 hours of (probably expensive to hire at consultancy rates) expert time and 6.25 hours of evaluator (usability professional) time taking detailed notes. Transcription of the HCI experts comments and analysis took another 21 hours.
- For end user testing a total of 125 hours; Twenty end users, totalling 25 hours with a supporting 25 hours of evaluator time, answering queries, taking notes and monitoring the video camera plus 75 hours for statistical analysis.

For the end user testing we would expect to use people who would actually work on the final system, either clients or employees, and would estimate that the 1.25 hours each of them spent filling out questionnaires, performing the tasks and debriefing were a minimum. Twenty end users, totalling 25 hours with a supporting 25 hours of evaluator time, answering queries, taking notes and monitoring the video camera, could possibly be reduced which would also reduce the 75 hours for statistical analysis. User-centred design techniques [3] require that users should be aware of, and committed to, new design, so this time should be considered in that respect.

Ease of problem fixing

The complexity of fixing an error will depend on how accurately it has been reported, how thoroughly the cause has been understood, and on implementation factors, such as whether it is a bug that can be fixed in the interface software, a dependency on the environment or operating system that needs a work round, or a fundamental design flaw. Some errors may just go away with use - we could define another characteristic, that is, the *learnability* of an error. In general it is to be expected that HCI experts should be better at accurate, thorough reporting and identifying causes.

Effectiveness of problem fixing

The frequency with which an error is encountered seems to have no correlation with the trouble it causes to the user. Fixing as *many* usability problems as possible is all to the good, but from the user's point of view the *quality* of improvement will also depend on the *severity* of the error. An interface problem which produces the wrong answer, or stops the user proceeding at all, is far more critical than one which merely requires him to be more accurate with the mouse, which is easily *learnable.* From the problems we identified, the 'Don't know how...' type is the most severe, it results in the user stopping and asking for help, or abandoning and restarting his task.

We therefore, as a rule of thumb, define the severity of a user error as:

- *High* if the user could not continue without external help, or performed the task wrongly.
- *Medium* if the user continued after pausing for a significant time, or tried alternatives successfully.
- *Low* if the user continued with a pause for thought.

Problems encountered by end users but *not found* by heuristic evaluation

Nielsen [15] claims that the heuristic method using HCI experts will rapidly (by about five evaluators) find a high percentage of all errors with an interface. However, in our experiments, 39% (15/38) of usability problems identified by end user testing were *not* noted by the heuristic evaluators. Note that there is a difference in the way we quote the results for the two methods, this is because the experts identify an interface error and *predict* that it will cause user problems, whereas for end-users we identify the symptom and then must *infer* the cause or more general problem. The experts are looking for causes of error and predicting effects. The end users encounter effects, from which we infer causes.

Table 1 includes task specific and Windows manipulation problems shown by the users but missed by heuristic evaluation. Figure numbers in the symptoms column link to INTUITIVE task steps as defined above, navigating the information space (Figure 1), query formation/refinement (Figure 2), results presentation (Figure 3). Although the precise symptom may not be illustrated the reader can link the symptom to task step screen.

Symptom	**User problem**	**Freq**	**How severe**	**Cause**
Repeated re-submission of the same query while its results are on the screen (Fig 3).	1. Can't relate results to the query.	6	High	Inadequate feedback - not flagged as success.
	2. Result set of zero mis-understood.	5	High	
Stuck on creating query (Fig 2).	1. Don't know how to constrain a query.	6	High	Inadequate cues provided.
	2. Confused by the interface feature (*eg.* greyed out or positional item).	6 (2)	High	Non-standard use of interface features.
Repeated attempt to delete query clause (Fig 2).	User model assumes *delete* functionality.	2	High	System lacks *delete* function-ality.
Can't make sense of results (Fig 3).	Results window obscured.	6	High	Screen organ-isation confusing.
Repeated mouse clicks on Results window (Fig 3).	Assumes results window is interactive.	1	Low	No system error faulty user model.
Can't proceed - stuck with open dialogue box (Fig 2).	Dialogue box not part of user model.	1	Med	OS[2] function-ality not part of user model.
Attempts to interact with the wrong window (Fig 3).	Change of window focus dis-orientating.	1	Med	OS function-ality not part of user model.
Missed menu/menu option hot spot (Fig 1,2).	Manipulation	2	Low	OS feature. Operation difficult.
Difficulty in moving window (Fig 1,2,3).	Manipulation	5	Low	OS feature. Operation difficult.
Difficulty in stretching windows sideways (Fig 1,2,3).	Manipulation	1	Low	OS feature operation difficult.
Difficulty in double-clicking (Fig 1,2).	Manipulation	1	Low	OS feature. Operation difficult.
Can't view scrollable window contents (Fig 3).	Don't understand scroll bar arrows.	2	Med	OS feature. Concept difficult.
Difficulty in closing windows (Fig 1,2,3).	Manipulation	1	Low	OS feature. Operation difficult.

Table 1. Usability errors identified by end user testing but not by heuristic evaluation.

[2] Operating System

Apart from Windows manipulation problems, which the HCI experts using Nielsen's heuristics probably regarded as part of the environment and for which they were not explicitly looking, the errors in this table are characterised by the *manner* in which INTUITIVE failed to support users' expectation of the task: poor cueing and feedback and lack of expected functionality. The repeated submission error is totally absent from the expert list, although it was very widely encountered as well as being of high severity. Indeed the uncovering of this problem alone would be sufficient to justify the effort of an additional alternative to heuristic evaluation. As for the next most severe - query constraint - the experts did identify something amiss but did not predict the manner of the user problem.

Errors identified by HCI experts using heuristics and by end users

In order to see what might have caused the HCI experts to miss the errors in Table 1, it is instructive to compare a small representative sample of the errors experienced by users which *were* identified by at least 3 experts (Table 2). These errors are less concerned with *task* performance and more with the *features* of the interface.

Error identified by heuristic evaluation	No of experts	User symptom	No of users	How severe
The query window does not provide a complete view of the query - only one entity at a time (Fig 2).	3	Attempts to toggle entities in order to see them all at once.	9	High
Menu terminology confusing (Fig 1,2).	5	Wrong menu option selected.	10	High
Attribute selection unpredictable (Fig 2).	3	Wrong hotspot selection in attribute list.	15	High
		Repeated right hand side selection on attribute list.	1	Low

Table 2. Usability errors identified by user testing and by heuristic evaluation.

Problems identified by heuristic evaluation but *not* by user testing

From tables 1 and 2, and from the fact that the HCI experts were using a heuristic checklist, we would expect that errors identified by experts but not encountered by end users would be less about the task and more concerned with interface features. About 40% (34 out of 86) of problems identified by heuristic evaluators were not identified by end users. Table 3 shows a subset of typical problems. Note that there might have been many more of these but for a preliminary demonstration given to end users. In order to give the end users some understanding and confidence in using the system, they were given a demonstration which covered many of the aspects identified by the heuristic evaluators as potential problem areas. These have been excluded from the comparative data in this paper.

Error identified by heuristic evaluation	Predicted User Problem	Freq	How severe	Heuristic /cause
No arrows indicating the direction of relations (Fig 1,2).	Relationship between entities not fully understood.	1	High	Predicted user model not supported.
Clicking on text list un-expectedly produces a pop up menu (Fig 2).	Clickable items not clearly identifiable.	1	Med	Non-standard interface feature.
Having submitted the query, the user should be able to cancel the operation.	User model assumes query can be cancelled.	2	Med	All actions should have undo.
Reformulate dialogue box has no cancel option.	User model assumes reform-ulation can be cancelled.	2	Med	All actions should have undo.
Slow response time to display results (Fig 3).	user model confused by system waiting.	1	Med	Immediate feedback to all actions.
Need to be informed of delay information during database search (not after).	Confusion while system waiting and may need to cancel.	1	Med	Immediate feedback and need to cancel.
Can't pick out items in the results sets to construct a query (Fig 3).	User model infers iterative query.	2	Med	System does not support user model.
There should be a clean up command.	General expectation.	1	Med	Missing UI require-ment.

No undos /cancels available.	General expectation.	2	Med	Missing UI require-ment.
Help needed.	General expectation.	2	Med	Missing UI require-ment.

Table 3. Usability errors identified by heuristic evaluation but not by user testing

The errors in Table 3 are all genuine interface faults and should be corrected, even if there has as yet been no visible user problem.

DISCUSSION

Advantages and disadvantages of heuristic evaluation

Whereas we can agree with a very large part of Nielsen's attitude to usability as described in Nielsen [15], we have more difficulty with the implication that a distillation into ten heuristics will act as an adequate guide to evaluators. Heuristic evaluation does find usability problems that fail HCI principles (*e.g.* lack of feedback, lack of visibility, obscured visibility, and inappropriate terminology). It is also supposed to find where the *task* flow is not simple and natural, but this is much more difficult in practice.

Nielsen implies that the *number* of usability errors found by a set of evaluators using his method can usefully be quoted as a proportion of all possible errors in the interface (defined for the total of errors findable by his evaluators). As mentioned above, 39% (15/38) of usability problems identified by user testing evaluation were *not* identified by heuristic evaluators, who totally missed at least one error which if uncorrected, would certainly have caused significant problems. We consider whether more or different evaluators would have found more of the user problems, whether the checklist could be improved, whether different instructions (a different role) should have been given to the evaluators, or whether the errors missed are intrinsically unfindable by this method.

Heuristic evaluation problems can be subjective

Heuristic evaluators identify usability problems which do not comply with predefined principles, but effectively they are in the position of using their judgement to do this and therefore it is a subjective process. Nielsen states that novice users are poor evaluators, HCI experts are 1.8 times as good, and domain and HCI experts 2.7 times as good. Our evaluators were HCI experts and some were Windows aware. They were not specialists in information retrieval and so could misunderstand the task problem, inferring an erroneous user model. For example, one heuristic evaluator noted that there was no automatic selection (for output) of constrained attributes, thereby implying that there should be. However, once one actually thinks this through, automatic attribute selection is not the best solution for many users' tasks.

Heuristic evaluation problems are often not distinct

A recurring problem found in attempting to analyse the heuristic evaluators' problems is that they are often non-distinct. For example, three heuristic evaluators noted, "there are little or no shortcuts available", whereas a fourth heuristic evaluator noted, "it would be handy to have a shortcut for Query>Browse". Is this one problem or two? It is estimated that a good proportion of problems identified were in fact subsets of other problems.

Observation vs. Immersion

Heuristic evaluators were, naturally, observing the interface and were not absorbed in using the system to perform a task. In contrast, end users were visibly absorbed, using the system to perform specific tasks. The difference in emphasis is one of the reasons why the heuristic evaluators failed to identify some of the end users' task based problems (see Table 1).

We *suggested* similar tasks to the experts as were given in a structured way to the end-users. However, because it is not task based, Nielsen has no system for evaluating every aspect of an interface for performing complex tasks. Heuristic evaluators usually carry out two passes on the interface but they frequently explore new directions and new tasks as they work and often go from a general problem "feedback present but not always comprehensible" to the specific "no feedback if user selects a relation, what does this mean?". This often means that the problems identified are not matched to a specific task and can be non-specific. The vaguer problems such as "system doesn't provide good recovery from errors", "no active guidance even at a simple level", "cryptic error messages, but the system doesn't blame the user" all need to be investigated further. It is possible that a better choice of tasks would have helped. But the HCI experts using heuristics and the end users were given the same tasks.

Terminology

Heuristic evaluators are not always precise in their terminology. There was an interesting confusion between help and feedback. One heuristic evaluator noted that there was "not much online help, and it wasn't context sensitive". In fact there was no online help, but there was feedback which this evaluator interpreted as help.

The Heuristics checklist

It is difficult to know how far the heuristics guided the evaluators, even though they were given a detailed synopsis of the meaning of each heuristic.

Completeness - Of the errors found by our HCI experts using heuristics, relatively few were tagged against one of the heuristics. Some heuristics were not mentioned explicitly at all. It is possible that if the checklist is not fully comprehensive, then by focusing in detail on one aspect, attention is deflected away from another.

Evenness - The heuristics themselves are very varied in level and precision. Some heuristics are simple and precise (*e.g.* Provide feedback and Help/documentation), whereas others are imprecise and difficult to check for completeness (*e.g.* Prevent errors). Some problems can be missed altogether, such as task related errors caused by the operating system (see Table 1).

Advantages/disadvantages of user testing evaluation

User testing is good at assessing the system in action, at identifying problems users experience while performing real tasks. Of the six highest user severity errors (two of which were amongst the most frequent errors) three were completely missed by the expert evaluators, and two were found by only one evaluator. However, observation of the video data indicated that users (especially female ones) tend to blame themselves rather than the interface, however much one tells them that it is not they but the interface being tested. Their post-test questionnaires are excessively favourable. They do not explicitly criticise, the interface problem must be inferred from the presenting symptom, with additional verbal explanation provided if you're lucky. As they do not identify causes, it is not possible to predict associated symptoms. User testing is uneven in operation and will miss any feature not encountered in the tasks. It is thus very dependent on the quality of the experimental tasks. It found fewer errors at greater cost than heuristic testing.

The fact that about 40% of all problems identified by heuristic evaluators were not identified explicitly in our user experiments suggests either a lack of thoroughness in the composition of user tasks, that users learnt to perform the task in spite of the interface error, or that the manifestation of a predicted error was noted as something different. It cannot be assumed that problems not found by user testing are not problems. The likelihood is that they will occur in use over extended periods. We look in more detail at the errors the end users failed to find.

Absent or inadequate Checklist items

Users users don't access help even if they are struggling. Only 2 users (out of 20) tried to access the help button. This may be because general experience of online help is that it is not helpful.

Experts identified the lack of undo, cancel and clean up facilities throughout the system. Although the specific instances did not map precisely onto user actions, the general inability to modify and cancel the effect of actions perceptibly worried users.

The HCI experts using Nielsen's heuristics noted that feedback was present but not always clear; again the specific instances were not matched, but feedback problems were encountered throughout user testing, indeed some of the user errors, not identified by heuristic testing, were due to inadequate feedback. Expert evaluators were good at finding poor terminology and lack of clarity. These caused users to stumble for a short time, but were then learnt, so were not flagged as errors. However, they would cause continuing irritation and possible errors over extended periods of use, so must be identified and corrected.

Task specific interface problems

It is the proper purpose of heuristic evaluation to find task specifics like the lack of directional arrows on relational links. Although these gave no problems in the tasks presented, they could be predicted to confuse the user task model and cause problems with more complex tasks.

Users need information on the time to access the database when submitting the query, not on return. This is the sort of criticism that users just don't voice, and is difficult to infer, so it is unlikely that it would be picked up by observing users alone.

All the items relating to query construction and modification need to be arranged visibly and interactively around the query window. Heuristic evaluators identified lack of such screen organisation as a general problem. Catching it by observing users would be time consuming, because it is experienced as a series of sub-problems.

SUMMARY AND CONCLUSION

Although heuristic evaluation and user testing share the same aim, that is, to identify the usability problems in a system, the actual results produced by each technique are quite different in kind. For example, observing users showed that they frequently chose the wrong menu option to move from the Navigator to a Query window. But we do not know from observation alone whether this is on account of poor menu terminology, disorientation due to moving away from the Query window and searching the menu, or for some other reason. Heuristic evaluation explicitly stated that the terminology of the menu option in question was unclear, noted that the menu is located too far away from the query, and recommended that a button for this task should be available on the Navigator in addition to the menu option.

This suggests that one difference between the techniques is that end user testing may indicate the *symptom* of the problem (*i.e.* the observed problem, for example, users chose the wrong menu option), whereas heuristic testing goes some way to identifying the *cause* of the problem (*i.e.* the reason for the observed problem, for example, poor terminology, loss of window focus), and furthermore can suggest a possible solution the problem (*i.e.* provide a button on the Navigator tool to deal with loss of window foucs problem). We can cite another example. Heuristic evaluation noted that only one entity's constraints are visible at any one time and that users need a view of the whole query. End users were observed toggling entities, the toggling activity enables users to view all of the query components.

The point we wish to make is that observation alone gives little information as to the cause of the problem, it deals primarily with the symptom. Not understanding the underlying cause has implications for re-design as a new design may remove the original symptom, but if the underlying cause remains, a different symptom may be triggered. User testing fails to deal with the underlying cause and is time consuming. Also, only simple tasks can be given in a reasonable experimental time, and many problems show up only in repeated or complex tasks. Heuristic evaluation to some extent identifies the underlying cause of the problem as the evaluator articulates which parts of the design fail the principles and how. This is particularly true when using HCI (as opposed to domain) experts who should be aware of the underlying psychology.

In summary, heuristic evaluation provides causal categories so it can help analyse observed usability problems, but observation of novices is still vital as many problems are a consequence of the users' knowledge, or lack of it, when interacting with a system. Heuristic evaluators cannot place themselves in users' shoes, hence they miss errors.

It is a truism that lack of precision takes longer and costs more. Usability testing is costly and time-consuming partly because the techniques are so general. In order to fully assess an interface it is necessary to use a variety of techniques: there might be whole classes of error missed by any one. But, by optimising each evaluation technique to focus on the aspects it does best, we hope to achieve better interfaces more cheaply in less time.

ACKNOWLEDGEMENTS:
This work was partially supported by the European Union's Esprit programme in Project 6593 INTUITIVE, whose partners are CAP Gemini Innovation, INRIA, IBERMATICA, SISU, BRAMEUR, Lloyds Register and City University.

REFERENCES

1. Bates, M. The design of browsing and berrypicking techniques for the online search interface. Inline Review, pp 407-423, (1989).

2. Bellotti, V. Implications of Current Design Practice for the Use of HCI Techniques, in: People and Computers IV, D. M. Jones and R. Winder (eds.), Cambridge University Press, 13-14, (1988).

3. Bevan, N. INUSE European Usability Support Centres. Telematics Applications Project IE 2016. Information Engineering Usability Support Centres. Principal author: Nigel Bevan of the National Physical Laboratory. Main Editor Dr. J Kirakowski of the Human Factors Research Group, University College, Cork. June 1996.

4. Brown, C. Human Computer Interface Design Guidelines. Ablex, New York, (1988).

5. Card, S., Moran T., Newell. A. The Psychology of Human-Computer Interaction, Laurence Erlbaum Associates, (1983).

6. Carroll, J. M., Koenemann-Belliveau, J., Rosson, M., and Singley, M. Critical Incidents and Critical Threads in Usability Evaluation in Proc. People and Computers viii, J Alty et al (eds), Cambridge University Press, 279-292, (1993).

7. Ericcson, K. A. and Simon H. A. Protocol Analysis. MIT Press, (1984).

8. ISO, International Standards Organisation, Draft International standard 9241, Ergonomic Requirements for office work with visual display terminals, Part 11, Usability and parts 12-16 on user interface design guidelines, ISO or National Standards Organisations, (1995).

9. Kieras, D. E. and Polson, P. A. An Approach to the Formal Analysis of User Complexity, International Journal of Man Machine Studies, 22, pp 365-395, (1985).

10. Kirakowski, J. & Corbett, M. SUMI: the Software Measurement Inventory. British Journal of Educational Technology, 24, 210-212, (1993).

11. Lewis, C., Polson, P., Wharton, C., and Reiman, J. 'Testing a Walkthrough methodology for Theory-based Design of Walk-up-and-use Interfaces', Proc. CHI-90, J R Chew and J Whiteside (eds), ACM press, 235-241, (1990).

12. Monk, A., Wright, P., Haber, J., and Davenport L. Improving your human computer interface. Prentice Hall (BCS Practitioner series), (1993).

13. Morris, R. C. T. Towards a User-Centered Information Service, Journal of the American Society for Information Science, vol. 45(1), pp20-30, (1994).

14. Nielsen, J. Getting Usability Used. Human Computer Interaction, Interact, '95, pp3-12. (eds.) K Nordby and P Helmersen, (1995).

15. Nielsen, J. Usability Engineering. Academic Press. New York, (1993).

16. Polson, P., Lewis, C., Reiman, J., and Wharton, C. Cognitive Walkthroughs: A Method for Theory Based Evaluation of User Interfaces. International Journal of Man Machine Studies, Vol, 36, pp365-395, (1992).

17. Preece, J. Human-Computer Interaction. Addison-Wesley Publishing Company, (1994).

18. Ravden, S. and Johnson, G. Evaluating Usability of Human Computer Interfaces. Ellis Horwood, (1989).

19. Smith, S. L and Mosier, J. Guidelines for Designing User Interface Sofwatre. Mitre Corporation Report MTR-10096, Bedford, MA, (1986).

20. Sutcliffe, A. G., Ryan, M., Springett, M., and Doubleday, A. Model Mismatch Analysis: Towards a Deeper Explanation of Users' Usability Problems. (City University, 1996).

21. Sutcliffe, A. and Springett, M. From Users Problems to Design Errors: Linking Evaluation. Proceeding People and Computers VII. (eds.) A Monk et al., Cambridge University Press, (1992).

22. Whiteside, J., Bennett, J., and Holzblatt, K. Usability Engineering, our experience and evolutions. in Handbook of Human Computer Interaction, M Helander (ed.) Elsevier North Holland, 791-817, (1988).

23. Wright, P. and Monk, A. F. 'Evaluation for Design', People and Computers V., A G Sutcliffe and L Macaulay (eds); Cambridge University Press, 345-358, (1989).

24. Wright, P., Monk, A., Carey, T. Co-operative Evaluation. The York Manual, version 0.4, July 1989. P Wright & A Monk, Department of Psychology, University of York, York Y01 5DD.