FINDING USABILITY PROBLEMS THROUGH HEURISTIC EVALUATION

Jakob Nielsen
Bellcore
445 South Street
Morristown, NJ 07962-1910
nielsen@bellcore.com

ABSTRACT

Usability specialists were better than non-specialists at performing heuristic evaluation, and "double experts" with specific expertise in the kind of interface being evaluated performed even better. Major usability problems have a higher probability than minor problems of being found in a heuristic evaluation, but more minor problems are found in absolute numbers. Usability heuristics relating to exits and user errors were more difficult to apply than the rest, and additional measures should be taken to find problems relating to these heuristics. Usability problems that relate to missing interface elements that ought to be introduced were more difficult to find by heuristic evaluation in interfaces implemented as paper prototypes but were as easy as other problems to find in running systems.

Keywords: Heuristic evaluation, Interface evaluation, Usability problems, Usability expertise, Discount usability engineering, Telephone-operated interfaces.

INTRODUCTION

Heuristic evaluation [17] is a method for finding usability problems in a user interface design by having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics"). Heuristic evaluation thus falls into the general category of usability inspection methods together with methods like pluralistic usability walkthroughs [1], claims analysis [2][3][10], and cognitive walkthroughs [11][19], with the main difference being that it is less formal than the other methods and intended as a "discount usability engineering" [13][16] method. Independent research has found heuristic evaluation to be extremely cost-efficient [8], confirming its value in circumstances where limited time or budgetary resources are available.

The goal of heuristic evaluation is the finding of usability problems in an existing design (such that they can be fixed). One could thus view it as a "debugging" method for user interfaces. The present article extends previous work on heuristic evaluation [4][12][14][17] by looking more closely at several factors that may influence the probability of finding usability problems. A probabilistic approach is necessary in examining the success of a method that is heuristic and approximate. The factors considered below are the expertise of the evaluators, the severity of the usability problems, the individual heuristics, and the activities needed to identify the problems.

EFFECT OF THE EVALUATORS' USABILITY EXPERTISE

Heuristic evaluation was originally developed as a usability engineering method for evaluators who had some knowledge of usability principles but were not necessarily usability experts as such [17]. Subsequent research has shown the method to be effective also when the evaluators are usability experts [4][8]. Unfortunately, usability experts are sometimes hard and expensive to come by, especially if they also need to have expertise in a particular kind of application.

To investigate the effect of having evaluators with varying levels and kinds of expertise, a study was conducted where the same interface was subjected to heuristic evaluation by three groups of evaluators: "Novice" evaluators with no usability expertise, "regular" usability specialists, and "double" usability specialists who also had experience with the particular kind of interface being evaluated.

A Telephone Operated Interface

A "voice response" system is a computer information system accessed through a touch tone telephone. The user's only input options are the twelve buttons found on a regular telephone (the digits **0–9** and the special characters * and #). The system's only output is through speech and sometimes sound effects. This interaction mechanism provides literally hundreds of millions of terminals to any computer system and allows it to be accessed from almost anywhere in the world [6][7].

Because of the variety of evaluators employed in the present study, a printed dialogue was evaluated instead of a running system. The evaluators were given a dialogue that had been recorded from a voice response system which will be referred to here as the BankingSystem. Evaluating an interface on the basis of a written specification is actually a reasonable task, and is one of the strengths of the heuristic

evaluation method. It lends itself to such evaluations as well as to evaluations of implemented systems [14].

The BankingSystem is a telephone operated interface to the user's bank accounts. The user's task in the sample dialogue was to transfer $1,000 from the user's savings account to the user's checking account. The dialogue between the BankingSystem (S) and the user (U) in Figure 1 is took place as the user tried to perform this task. This dialogue has actually taken place, the underlying problem being that the user had not authorized the bank to accept transfers over the phone.

The user can be assumed to be provided with printed instructions stating that the system uses the # key to signify the end of the user's input (in the same way as many other systems use an enter key). As long as the user has not hit the # key, it is possible to correct input by pressing ** (the asterisk key used twice). This option is not used in the dialogue in this example, however. The printed instructions were not evaluated as part of the heuristic evaluation.

For the heuristic evaluation, the evaluators were asked to keep in mind those basic technical limitations of the system which were due to the access through a touch tone telephone and not to include criticism of the very use of 12-button input and speech output instead of, say, input though speech recognition or output through graphical dialogues with pull-down menus, etc.

Even the small sample dialogue in Figure 1 contains a rather large number of usability problems as listed below. Readers who are only interested in the general results of the heuristic evaluation and not the individual usability problems can skip ahead to the section titled *Results of the Heuristic Evaluation.* The list of usability problems is ordered according to a rough estimate of the relative seriousness of the problems. Major usability problems are those that have serious potential for confusing users or causing them to use the system erroneously while minor problems may slow down the interaction or inconvenience users unnecessarily. For each problem a brief phrase in italics indicates what well-established usability principle was broken by the dialogue with respect to that problem. These phrases are taken from the list of nine usability heuristics used in earlier papers on heuristic evaluation [12][17] and are discussed in further detail in [16].

(First there is a short dialogue in which the user is identified by entering an identification number and a secret access code—this is not part of the evaluation exercise)

(1) S: Enter one for account information, three for transfers between your own accounts, ...

(2) U: **3#** *{the user interrupts the system}*

(3) S: Enter account to transfer from

(4) U: **1234567890#** *{savings account number}*

(5) S: Enter account to transfer to

(6) U: **#** *{an abbreviation for the checking account}*

(7) S: Enter amount in cents

(8) U: **100000#**

(9) S: From account number twelve thirtyfour fiftysix seventyeight ninety to account number primary account, a transfer of one thousand dollars is to be made. Press one to confirm, zero to cancel.

(10) U: **1#**

(11) S: You do not have access to use this function.

Figure 1 *A sample dialogue between a user (U) and the BankingSystem (S). The statement numbers are not part of the dialogue but are used to refer to individual statements in the analysis.*

Major Problems

Problem 1. The error message in statement 11 appears much too late. It is not reasonable to have the user enter large amounts of information only to learn that it was all in vain. The error message should have appeared immediately after the user's initial choice of the function in statement 2. One evaluator noted that the current form of the dialogue seems to be designed as a front end to a traditional full screen mainframe system where users fill in an entire screen of data before hitting enter and transmitting all of the data to the computer at once. Even if the bank's main computer system with the account database were indeed a traditional transaction-oriented mainframe, one should not let the properties of the backend computer impact the dialogue when a new interaction medium is introduced. In many cases, the best solution would be to prevent the error from occurring in the first place by only listing those functions in the main menu which the user is authorized to use. *(Simple and natural dialogue).*

Problem 2. Users should not be required to convert an amount of money to cents since only a very mathematically inclined person will find it easy to think of $1,000 as 100,000 cents. This problem can be completely avoided by simplifying the system to allow transfer of whole dollar amounts only. Doing so will also speed up the interaction by eliminating two keystrokes. For transfers between the user's own accounts, whole dollars will be completely adequate. For a bill-payment system, it might still be necessary to have some mechanism for specifying cents. In that case, a solution might be to have users enter first the dollars and then be prompted for the cents. Since the system allows the user to interrupt prompts, a transfer of a whole dollar amount could still be achieved very fast if the user entered **1000##** to signify $1,000. The fact that there were no digits between the two # keystrokes would mean "and no cents." *(Speak the user's language).*

Problem 3. The error message in statement 11 is not precise. It is not clear what "this function" refers to. The problem could be transfers in general, transfers between the two specific accounts, or that the user did not have the $1,000 in the savings account. The system should explicitly state that the user was not allowed to initiate any transfers, thus also avoiding the use of the computer-oriented term "function." The expression "access" is also imprecise as well as being a rather computer-oriented term. An access problem might have been due to system trouble as well as the missing authorization form to allow telephone-initiated transfers. *(Precise and constructive error messages).*

Problem 4. The error message in statement 11 is not constructive. It does not provide any indication of how the user might solve the problem. Users might think that the bank did not want their category of customers to use the transfer facility or that the problem would solve itself if they had more

money in their account. *(Precise and constructive error messages).*

Problem 5. The expression "primary account" in statement 9 is not user-oriented. The system should use user-oriented terms like "checking account." *(Speak the user's language).*

Problem 6. Instead of having the user enter ten digit account numbers, the system could provide the user with a short menu of that user's accounts. There is a much larger risk that the user will make errors when entering a ten digit account number than when entering a single digit menu selection. A menu-based dialogue would probably speed up the dialogue since users would be required to do much less typing and would not need to look up their account numbers. In statement 6, the current design does provide a shortcut by letting the checking account number be the default but this shortcut again involves some risk of errors. Also note that a menu of account names might be difficult to construct if the customer had several accounts of the same type. Assuming that most customers do limit themselves to one of each account, it would still be best to use the menu approach for those customers and stay with the current interface for the difficult customers only: Just because one cannot solve a problem for 100% of the users, one should not skimp out of solving it for, say, the 80% for which a better solution can be found. *(Prevent errors).*

Problem 7. It is very likely that the user will forget to press the # key after having entered menu selections or account numbers. Since the number of digits is predetermined for all user input except for the amount of money, the system in fact does not need a general terminator. The system should only require a # in situations where the input has an indeterminate number of digits and it should then explicitly state the need for this terminator in the prompt. In these few cases, the system could furthermore use a timeout function to give the user a precise and constructive reminder after a certain period of time without any user input, since such a period would normally indicate that the user had finished entering input but had forgotten about the #. *(Prevent errors).*

Problem 8. The feedback in statement 9 with respect to the chosen accounts simply repeats the user's input but ought to restate it instead in simpler and more understandable terms. Instead of listing a ten-digit account number, the feedback message should provide the system's interpretation of the user's input and state something like "from your savings account." By using the name of the account (and by explicitly including the word "your"), the system would increase the user's confidence that the correct account had indeed been specified. *(Provide feedback).*

Minor Problems

Problem 9. The listing of the main menu in statement 1 should reverse the order of the selection number and the function description for each menu item. The current ordering requires users to remember each number as the corresponding description is being spoken since they do not yet know whether they might want to select the function [5]. *(Minimize the user's memory load).*

Problem 10. The most natural order of menu options in this type of system would be a simple numeric order, so the main menu in statement 1 should not skip directly from selection 1 to 3. Users who remember that account transfers were the second option on the list might be inclined to utilize the interrupt facility in the system and simply enter 2 without waiting to hear that the menu choice should have been 3 because there is no option 2 in the system. *(Simple and natural dialogue).*

Problem 11. Feedback on the user's choice of accounts and amounts appears much too late. Normally a lack of feedback would be a "major" problem, but the present design does provide the ** editing facility as well as some feedback (even though it is delayed). *(Provide feedback).*

Problem 12. The options in the accept/cancel menu in statement 9 have been reversed compared to the natural order of the numbers zero and one. Actually it would be possible to achieve some consistency with the rest of the dialogue by using the # key to accept and the * key to cancel. Note that some systems (for instance many British systems) have the reverse convention and use * to indicate the answer yes and # to indicate the answer no. The assignment of meaning to these two keys is more or less arbitrary but should obviously be consistent within the system. The choice between the two meanings of # and * should be made to achieve consistency with the majority of other similar systems in the user's environment. *(Simple and natural dialogue).*

Problem 13. The phrase "account number primary account" in statement 9 is awkward. When referring to an account by name instead of number, the field label "number" should be suppressed. *(Simple and natural dialogue).*

Problem 14. The term "account" in prompts 3 and 5 should be changed to "account number" as the user is required to enter the number. *(Speak the user's language).*

Problem 15. It would probably be better to read out the account numbers one digit at a time instead of using the pairwise grouping in statement 9 since users may well think of their account numbers as grouped differently. The change in feedback method should only apply to the account numbers since it is better to report $1,000 as "one thousand dollars" than as "dollars one zero zero zero." *(Simple and natural dialogue).*

Problem 16. Different words are used for the same concept; "enter" and "press." It is probably better to use the less computer-oriented word "press." *(Consistency).*

The complete voice response system raises several usability issues in addition to the sixteen problems discussed above. One of the most important issues is the voice quality which of course cannot be evaluated in a printed version of the dialogue. Normally one would caution against using the almost identical prompts "Enter account to transfer from/to" (statements 3 and 5) since users could easily confuse them. But the speaker in a voice dialogue can place sufficient emphasis on the words "from" and "to" to make the difference between the prompts obvious.

Results of the Heuristic Evaluation

The BankingSystem in Figure 1 was subjected to heuristic evaluation by three groups of evaluators with varying levels of usability expertise. The first group consisted of 31 computer science students who had completed their first programming course but had no formal knowledge of user interface design principles. These novice evaluators were

	Novice evaluators	*"Regular" specialists*	*"Double" specialists*
Major usability problems:			
1. Error message appears much too late	68%	84%	100%
2. Do not require a dollar amounts to be entered in cents	68%	74%	79%
3. The error message is not precise	55%	63%	64%
4. The error message is not constructive	6%	11%	21%
5. Replace term "primary account" with "checking account"	10%	47%	43%
6. Let users choose accounts from a menu	16%	32%	43%
7. Only require a # where it is necessary	3%	32%	71%
8. Give feedback in form of the name of the chosen account	6%	26%	64%
Average for the major problems	**29%**	**46%**	**61%**
Minor usability problems:			
9. Read menu item description before the action number	3%	11%	71%
10. Avoid the gap in menu numbers between 1 and 3	42%	42%	79%
11. Provide earlier feedback	42%	63%	71%
12. Replace use of 1/0 for accept/reject with #/*	6%	21%	43%
13. Remove the field label "number" when no number is given	10%	32%	36%
14. Change the prompt "account" to "account number"	6%	37%	36%
15. Read numbers one digit at a time	6%	47%	79%
16. Use "press" consistently and avoid "enter"	0%	32%	57%
Average for the minor problems	**15%**	**36%**	**59%**
Average for all the problems	**22%**	**41%**	**60%**

Table 1 *The proportion of evaluators who found each of the sixteen usability problems. "Double" usability specialists had expertise in both usability in general and interfaces to telephone-operated interfaces in particular.*

expected to indicate a worst-case level of performance. Note that they were "novices" with respect to usability but not with respect to computers as such. The second group consisted of 19 "regular" usability specialists, i.e., people with experience in user interface design and evaluation but no special expertise in voice response systems.There is no official certification of usability specialists, but for the purpose of this study, usability specialists were defined as people with graduate degrees and/or several years of job experience in the usability area. The third group consisted of 14 specialists in voice response usability. These "double specialists" had expertise in user interface issues as well as voice response systems and were therefore expected to indicate the best level of heuristic evaluation performance one might hope for.

Table 1 presents the results of the three sets of evaluations and shows that heuristic evaluation was difficult for single evaluators. The above list of usability problems was constructed on the basis of the complete set of evaluations, but no single evaluator found all the problems. Problems 7, 9, 11, 12, 14, and 15 were not included in my own original list of problems but were added after I read the other evaluators' lists. On the other hand, the really catastrophic problems 1, 2, and 3 were found by more than half of the evaluators even in the group without any experience. Just fixing these three problems would improve the interface tremendously.

No group did really well, even though the "double specialists" with both usability expertise and voice response expertise were able to find well over half of the problems on the average. Table 1 indicates that usability specialists are better than people without usability training at finding usability problems and that it helps even more to have usability expertise with respect to the type of user interface being evaluated. The differences between the novices and the regular specialists and between the regular and double specialists are both statistically significant at the $p<.001$ level according to t-tests.

The average performance of individual evaluators may not be acceptable for the use of heuristic evaluation in a usability engineering project, even in the case of the double specialists, but the picture changes when the performance of groups of multiple evaluators is considered. Figure 2 shows the average proportion of the usability problems that would be found by aggregating the sets of problems found by several evaluators. These aggregates were formed in the same way as in previous studies of heuristic evaluation [17]. That is to say, for each group size, a large number of random groups were formed, and for each group, a given usability problem was considered found if at least one member of the group had found it. As can be seen from Figure 2, groups of double and regular usability specialists perform much better than groups of novice evaluators without usability expertise.

For the regular usability specialists, the recommendation from previous work on heuristic evaluation [17] holds in that between three and five evaluators seem necessary to find a reasonably high proportion of the usability problems (here, between 74% and 87%). For the double specialists, however, it is sufficient to use between two and three evaluators to find most problems (here, between 81% and 90%). For the novice evaluators, a group size of fourteen is necessary to find more than 75% of the problems. Using five novice evaluators, which is the upper range of the group size normally recommended for heuristic evaluation, results in the finding of 51% of the usability problems.

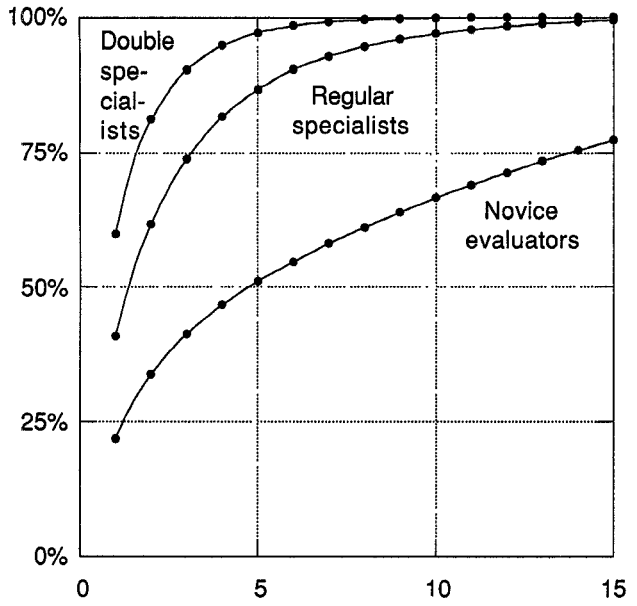

Figure 2 *Average proportion of usability problems found as a function of number of evaluators in a group performing the heuristic evaluation.*

Regular vs. Double Specialists

As mentioned above, the double specialists found significantly more usability problems than did the regular usability specialists. As can be seen from Table 1, the two groups of evaluators actually performed about equally well on many of the usability problems. A large part of the difference in performance is due to the five usability problems for which the probability of being found was thirty percentage points or more higher when the evaluators were voice response usability specialists than when they were regular usability specialists. As outlined below, these five problems were all either specifically related to the use of a telephone as the terminal or were related to the differences between auditory dialogues and screen dialogues.

Problem 9 (read menu item description before the action number) was found by 60% more voice response usability experts than regular usability experts. Even though a similar design issue of whether to list menu selection labels to the left or to the right applies to screen-based menus, the choice would be less crucial for usability. As a matter of fact, screen-based menus are probably better off having the label to the left of the description of the menu item (corresponding to reading the action number before the menu item description) since such a design leads to a uniform, close spacing between the two elements in each line of the menu.

Problem 7 (only require a # where it is necessary) was found by 39% more voice response usability experts than regular usability experts. This problem is much more relevant for telephone-based interfaces than for screen-based interfaces. Actually, the advice to speed up screen-based dialogues by eliminating the need for an enter key wherever possible would probably lead to *less* usable screen interfaces because of the reduced consistency.

Problem 8 (give feedback in form of the name of the chosen account instead of repeating ten digits) was found by 38% more voice response usability experts than regular usability experts. The underlying issue of providing understandable feedback would also apply to screen-based interfaces but the problem would be less serious in such a system because it would be easier for users to understand the ten-digit numbers in their printed form.

Problem 10 (avoid the gap in menu numbers between 1 and 3) was found by 37% more voice response usability experts than regular usability experts. Even though screen-based menus are also more usable when they are sequentially numbered, the numbering is less crucial in the case where the user can see the complete list of numbers simultaneously. A screen-based menu might have a blank line where menu item 2 would normally have been, thus indicating to the user that the number was reserved for a future extension of the system, if that was the reason for omitting the number from the menu. Often, screen menus for non-mouse systems would actually be based on mnemonic characters rather than numbers.

Problem 15 (read numbers one digit at a time) was found by 32% more voice response usability experts than regular usability experts. This problem could only occur in an auditory dialogue and the regular usability specialists would have no prior experience with this exact problem. A similar problem does occur in traditional screen dialogues with respect to the way one should present numbers such as telephone numbers or social security numbers that are normally grouped in a specific way in the user's mind.

These detailed results indicate that the double specialists found more problems, not because they were necessarily better usability specialists in general, but because they had specific experience with usability issues for the kind of user interface that was being evaluated.

In the discussion below of additional factors influencing the finding of usability problems through heuristic evaluation, the results from the "regular" specialists in the BankingSystem evaluation are used since they are the closest to the evaluators used in the other studies that are analyzed.

USABILITY PROBLEM CHARACTERISTICS

Table 2 summarizes six heuristic evaluations. Teledata, Mantel, and the Savings and Transport systems are documented in [17] and the names from that paper are used as headings. For the BankingSystem, the results are given with the "regular" usability specialists as evaluators. The Integrating System was evaluated by "regular" usability specialists and is discussed in [15]. The table only represents those usability problems that were actually found when evaluating the respective interfaces. It is possible that some additional usability problems remain that were not found by anybody, but it is obviously impossible to produce statistics for such problems.

Table 2 also shows three different ways of classifying the usability problems: by severity (i.e., expected impact on the users), by heuristic, and by location in the dialogue. Table 3 then shows the results of an analysis of variance of the finding of the 211 usability problems by single evaluators, with the independent variables being severity, heuristic, and location as well as the system being evaluated and the implementation of its interface. Two implementation categories were used: Teledata, Mantel, and the Banking System were evalu-

Name of interface:	Tele-data		Mantel		Banking System		All paper proto-types		Savings		Trans-port		Inte-grating System		All running systems		All prob-lems	
Number of evaluators:	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3	1	3
All problems (211)	.51	.81	.38	.60	.41	.74	.45	.74	.26	.50	.20	.42	.29	.59	.26	.54	**.35**	**.63**
Severity of problem:																		
Major usability problems (59)	.49	.79	.44	.64	.46	.77	.47	.74	.32	.63	.32	.65	.46	.79	.38	.70	**.42**	**.71**
Minor usability problems (152)	.52	.82	.36	.59	.36	.71	.45	.73	.26	.50	.19	.41	.21	.51	.22	.48	**.32**	**.59**
Applicable heuristic:																		
Simple and natural dialogue (51)	.52	.77	.51	.78	.46	.79	.51	.78	.14	.36	.21	.48	.31	.60	.24	.51	**.39**	**.66**
Speak the user's language (35)	.60	.90	.47	.70	.53	.88	.55	.83	.33	.62	.14	.32	.25	.62	.24	.51	**.41**	**.68**
Minimize user memory load (8)	.44	.81	.94	1.00	.11	.30	.48	.73	.26	.62	.15	.39	.27	.57	.24	.54	**.36**	**.63**
Be consistent (33)	.51	.85	.13	.34	.32	.70	.44	.76	.31	.58	.13	.29	.17	.39	.22	.45	**.31**	**.58**
Provide feedback (21)	.60	.87	.68	.96	.45	.79	.58	.87	.39	.72	.48	.85	.39	.69	.40	.72	**.46**	**.77**
Provide clearly marked exits (9)	.19	.50	.03	.09	•	•	.09	.26	.43	.62	.22	.53	•	•	.32	.58	**.20**	**.40**
Provide shortcuts (12)	.39	.78	•	•	•	•	.39	.78	.33	.67	.19	.48	.29	.63	.28	.61	**.29**	**.62**
Good error messages (25)	.42	.73	.33	.62	.37	.63	.38	.66	.23	.46	.33	.53	.27	.66	.25	.48	**.30**	**.56**
Prevent errors (17)	.50	.86	.17	.39	.32	.70	.29	.59	.19	.45	.21	.48	.37	.79	.22	.49	**.25**	**.54**
Where is problem located:																		
A single dialogue element (104)	.58	.85	.42	.66	.40	.75	.49	.77	.26	.53	.22	.45	.30	.59	.26	.53	**.38**	**.66**
Comparison of two elements (43)	.52	.85	.13	.35	.32	.70	.48	.80	.27	.53	.14	.34	.24	.56	.24	.51	**.31**	**.60**
Overall structure of dialogue (18)	.50	.84	•	•	.53	.84	.51	.84	.33	.67	.24	.50	.21	.53	.25	.54	**.35**	**.66**
Something missing (46)	.35	.67	.33	.51	.10	.30	.33	.58	.29	.55	.21	.49	.41	.71	.30	.58	**.31**	**.58**

Table 2 *Proportion of various types of usability problems found in each of the six interfaces discussed in this article, as well as in the collected set of 211 usability problems from all of them. The proportion of problems found is given both when the heuristic evaluation is performed by a single evaluator and when it is performed by aggregating the evaluations from three evaluators. Bullets (•) indicate categories of usability problems that were not present in the interface in question. The total number of usability problems is listed in parentheses for each category.*

ated as paper prototypes, whereas the Savings, Transport, and Integrating Systems were evaluated as running programs.

Even though Table 2 would seem to indicate that paper interfaces are easier to evaluate heuristically than running systems, one cannot necessarily draw that conclusion in general on the basis of the data presented in this paper, since different systems were evaluated in the two conditions. Earlier work on heuristic evaluation [14][17] did speculate that heuristic evaluation might be easier for interfaces with a high degree of persistence that can be pondered at leisure, and it is certainly true that paper prototypes are more persistent than running interfaces.

Table 3 shows that the system being evaluated had a fairly small effect in itself. This would seem to indicate a certain robustness of the heuristic evaluation method, but this result could also be due to the limited range of systems analyzed here. More studies of the application of heuristic evaluation to a wider range of interface styles and application domains will be needed to fully understand which systems are easy to evaluate with heuristic evaluation.

Major vs. Minor Usability Problems

Previous research on heuristic evaluation has pointed out that it identifies many more of the minor usability problems in an interface than other methods do [8]. Indeed, heuristic evaluation picks up minor usability problems that are often not even *seen* in actual user testing. One could wonder to what extent such "problems" should really be accepted as constituting usability problems. I argue that such minor usability problems may very well be real problems even though they are not observable in a user test. For example, inconsistent placement of the same information in different screens or dialog boxes may slow down the user by less than a second [18] and may therefore not be observed in a user test unless an extremely careful analysis is performed on the basis of a large number of videotaped or logged interactions. Such an inconsistency constitutes a usability problem nevertheless, and should be removed if possible. Also note that sub-second slowdowns actually accumulate to causing major costs in the case of highly used systems such as, e.g., those used by telephone company operators.

The top part of Table 2 compares the proportion of the major and the minor usability problems. A usability problem was

	df	*Mean Square*	*p*	ω^2
Problem severity	1	.842	.001	6.8%
Heuristic used	8	.118	.01	5.0%
Location of problem	3	.047	.37	0.1%
Implementation of interface	1	.747	.07	1.9%
System (nested in Implementation)	4	.123	.03	3.4%
Implementation × Location	3	.159	.02	6.8%
Residual	190	.044		

Table 3 *Analysis of variance for the probability of finding the 211 usability problems when using single evaluators. Other interactions than the one shown are not significant.*

ω^2 *indicates relative effect sizes in terms of proportion of total variance accounted for.*

defined as "major" if it had serious potential for confusing users or causing them to use the system erroneously. Note that the term "serious" was used to denote this category of usability problems in earlier work [12]. Given that the usability problems were found by heuristic evaluation and not by user testing, this classification can only reflect a considered judgment, since no measurement data exists to prove the true impact of each problem on the users. For the Teledata, Mantel, Savings, and Transport interfaces, the major/minor classification was arrived at by two judges with a small number of disagreements resolved by consensus, and for the Banking System a single judge was used. For the Integrating System, the mean severity classification from eleven judges was used. The simple classification of usability problems into only two severity levels was chosen because of this need to rely on a judgment; it was mostly fairly easy to decide which severity category to use for any given usability problem. See [9] and [15] for further discussions of severity ratings.

It is apparent from Table 2 that heuristic evaluation tends to find a higher proportion of the major usability problems than of the minor, and Table 3 indicates that the difference is statistically significant ($p<.001$) and one of the two largest effects identified in the table. Intuitively, one might even have gone as far as to expect the evaluators performing the heuristic evaluations to focus only on the major usability problems to the exclusion of the minor ones, but the results indicate that this is not the case since they find many more minor than major problems in absolute numbers (8.1 vs. 4.1 per system on the average). So the evaluators pay relatively more attention to the major problems without neglecting the minor ones.

Since the interfaces have many more minor than major problems, the minor problems will obviously dominate any given heuristic evaluation, even though the probability of being found is greater for the major problems. Usability engineers therefore face the task of prioritizing the usability problems to make sure that more time is spent on fixing the major problems than on fixing the minor problems.

Effect of the Individual Heuristics

Since heuristic evaluation is based on judging interfaces according to established usability principles, one might expect that problems violating certain heuristics would be easier to find than others. Table 3 indicates a significant and fairly large effect for heuristic. Even so, Table 2 shows that there are few systematic trends with respect to some heuristics being easier.

Considering all the 211 usability problems as a whole, Table 2 shows that usability problems have about the same probability of being found in a heuristic evaluation with the recommended three evaluators for most of the heuristics. Seven of the nine heuristics score in the interval from 54–68%, with the "good error messages" and "prevent errors" heuristics being slightly more difficult than the others. The only truly difficult heuristic is "provide clearly marked exits" (scoring 40%). The practical consequence from this result is that one might "look harder" for usability problems violating the "provide clearly marked exits" heuristic. For example, one could run a user test with a specific focus on cases where the users got stuck. One could also study user errors more closely in order to compensate for the relative difficulty of applying the two error-related heuristics, especially since problems related to user errors are likely to prove especially costly if the system were to be released with these problems still in place.

A contrast analysis of significance based on an analysis of variance for three evaluators confirms that usability problems classified under the "good error messages," "prevent errors," and "provide clearly marked exits" heuristics are more difficult to find than usability problems classified under one of the other six heuristics, with $p=.0006$.

Location of Problems in Dialogue

Even though the specific usability heuristic used to classify the usability problems had some impact on the evaluators' ability to find the problems, it might also be the case that other systematic differences between the problems can help explain why some problems are easier to find than others. Since heuristic evaluation is a process in which the evaluators search for usability problems, it seems reasonable to consider whether the circumstances under which the problems could be located have any influence.

The bottom part of Table 2 shows the result of considering four different possible locations of usability problems. The first category of problems are those that are located in a *single* dialogue element. An example of this category of usability problem is Problem 2 (do not require a dollar amount to be entered as cents) in the telephone operated interface analyzed earlier in this article. To find single-location problems by heuristic evaluation, the evaluator only needs to consider each interface element in isolation and judge that particular dialog box, error message, menu, etc.

The second category consists of usability problems that require the evaluator to *compare* two interface elements. This will typically be consistency problems where each interface element is fine when seen in isolation but may lead to problems when used together. An example from the BankingSystem is Problem 16 (both "press" and "enter" are used to denote the same concept).

The third category contains the usability problems that are related to the overall *structure* of the dialogue. An example from the BankingSystem is Problem 7 (only require a # where it is necessary). Another example would be the need to unify the navigation system for a large menu structure. These problems require the evaluator to get a grasp of the overall use of the system.

The final category of usability problems are those that cannot be seen in any current interface element but denote *missing* interface elements that ought to be there. An example from the BankingSystem is Problem 4 (the error message should have a constructive message appended). Note that the issue here is not that the current error message is poorly worded (that is easy to find and belongs in the category of single-location problems) but that the message ought to be supplemented with an additional element.

As can be seen from Table 3, the difference between the four location categories is not statistically significant. However, the interaction effect between location category and interface implementation *is* significant and has one of the two largest effect sizes in the table. As shown in Table 2, problems in the category "something missing" are slightly easier to find than other problems in running systems but much harder to find

than other problems in paper prototypes. This finding corresponds to an earlier, qualitative, analysis of the usability problems that were harder to find in a paper implementation than in a running system [14]. Because of this difference, one should look harder for missing dialogue elements when evaluating paper mockups.

A likely explanation of this phenomenon is that evaluators using a running system may tend to get stuck when needing a missing interface element (and thus notice it), whereas evaluators of a paper "implementation" just turn to the next page and focus on the interface elements found there.

CONCLUSIONS

Usability specialists were much better than those without usability expertise at finding usability problems by heuristic evaluation. Furthermore, usability specialists with expertise in the specific kind of interface being evaluated did much better than regular usability specialists without such expertise, especially with regard to certain usability problems that were unique to that kind of interface.

Previous results [17] with respect to the improvement in heuristic evaluation performance as groups of evaluators are aggregated were replicated in the new study reported above, and the general recommendation of using groups of 3–5 evaluators also held for the regular usability specialists in this study. For double specialists, a smaller group size can be recommended, since only two to three such evaluators were needed to find most problems. Of course, the actual number of evaluators to use in any particular project will depend on a trade-off analysis on the basis of curves like Figure 2 and the cost (financial or otherwise) of leaving usability problems unfound.

Major usability problems have a higher probability than minor problems of being found in a heuristic evaluation, but about twice as many minor problems are found in absolute numbers. Problems with the lack of clearly marked exits are harder to find than problems violating the other heuristics, and additional efforts should therefore be taken to identify such usability problems. Also, usability problems that relate to a missing interface element are harder to find when an interface is evaluated in a paper prototype form.

The results in this article provide means for improving the contribution of heuristic evaluation to an overall usability engineering effort. The expertise of the staff performing the evaluation has been seen to matter, and specific shortcomings of the methods have been identified such that other methods or additional efforts can be employed to alleviate them and find more of the usability problems that are hard to find by heuristic evaluation.

ACKNOWLEDGMENTS

The author would like to thank Jan C. Clausen, Heather Desurvire, Dennis Egan, Anker Helms Jørgensen, Clare-Marie Karat, Tom Landauer, Rolf Molich, and Robert W. Root for helpful comments on previous versions of the manuscript. The four studies reported in [17] were conducted by the author and Rolf Molich who also participated in the classification of usability problems as major or minor and in relating the problems to the heuristics. The further analyses and conclusions on the basis of this and other data as reported here reflect the views of the author of the present paper only.

REFERENCES

1. Bias, R. Walkthroughs: Efficient collaborative testing. *IEEE Software* **8**, 5 (September 1991), 94–95.
2. Carroll, J.M. Infinite detail and emulation in an ontologically minimized HCI. *Proc. ACM CHI'90* (Seattle, WA, 1–5 April 1990), 321–327.
3. Carroll, J.M., Kellogg, W.A., and Rosson, M.B. The task-artifact cycle. In Carroll, J.M. (Ed.), *Designing Interaction: Psychology at the Human–Computer Interface*. Cambridge University Press, Cambridge, U.K., 1991. 74–102.
4. Desurvire, H., Lawrence, D., and Atwood, M. Empiricism versus judgement: Comparing user interface evaluation methods on a new telephone-based interface. *ACM SIGCHI Bulletin* **23**, 4 (October 1991), 58–59.
5. Engelbeck, G., and Roberts, T.L. The effect of several voice-menu characteristics on menu selection performance. *Behaviour & Information Technology* **in press**.
6. Gould, J.D., and Boies, S.J. Speech filing—An office system for principals. *IBM Systems Journal* **23**, 1 (1984), 65–81.
7. Halstead-Nussloch, R. The design of phone-based interfaces for consumers. *Proc. ACM CHI'89* (Austin, TX, 30 April–4 May 1989), 347–352.
8. Jeffries, R., Miller, J.R., Wharton, C., and Uyeda, K.M. User interface evaluation in the real world: A comparison of four techniques. *Proc. ACM CHI'91* (New Orleans, LA, 27 April–2 May 1991), 119–124.
9. Karat, C.-M., Campbell, R., Fiegel, T. Comparisons of empirical testing and walkthrough methods in user interface evaluation. *Proc. ACM CHI'92* (Monterey, CA, 3–7 May 1992).
10. Kellogg, W.A. Qualitative artifact analysis. *Proc. INTERACT'90 3rd IFIP Conf. Human–Computer Interaction* (Cambridge, U.K., 27–31 August 1990), 193–198.
11. Lewis, C., Polson, P., Wharton, C., and Rieman, J. Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. *Proc. ACM CHI'90* (Seattle, WA, 1–5 April 1990), 235–241.
12. Molich, R., and Nielsen, J. Improving a human-computer dialogue. *Communications of the ACM* **33**, 3 (March 1990), 338–348.
13. Nielsen, J. Usability engineering at a discount. In Salvendy, G., and Smith, M.J. (Eds.), *Designing and Using Human–Computer Interfaces and Knowledge Based Systems*, Elsevier Science Publishers, Amsterdam, 1989. 394–401.
14. Nielsen, J. Paper versus computer implementations as mockup scenarios for heuristic evaluation. *Proc. INTERACT'90 3rd IFIP Conf. Human–Computer Interaction* (Cambridge, U.K., 27–31 August 1990), 315–320.
15. Nielsen, J. Applying heuristic evaluation to a highly domain-specific interface. *Manuscript submitted for publication.*
16. Nielsen, J. *Usability Engineering*. Academic Press, San Diego, CA, 1992.
17. Nielsen, J., and Molich, R. Heuristic evaluation of user interfaces. *Proc. ACM CHI'90* (Seattle, WA, 1–5 April 1990), 249–256.
18. Teitelbaum, R.C., and Granda, R.E. The effects of positional constancy on searching menus for information. *Proc. ACM CHI'83* (Boston, MA, 12–15 December 1983), 150–153.
19. Wharton, C., Bradford, J., Jeffries, R., and Franzke, M. Applying cognitive walkthroughs to more complex interfaces: Experiences, issues, and recommendations. *Proc. ACM CHI'92* (Monterey, CA, 3–7 May 1992).