HEURISTIC EVALUATION OF USER INTERFACES

Jakob Nielsen and *Rolf Molich*

Technical University of Denmark
Department of Computer Science
DK-2800 Lyngby Copenhagen
Denmark
datJN@NEUVM1.bitnet

Baltica A/S
Mail Code B22
Klausdalsbrovej 601
DK-2750 Ballerup
Denmark

ABSTRACT

Heuristic evaluation is an informal method of usability analysis where a number of evaluators are presented with an interface design and asked to comment on it. Four experiments showed that individual evaluators were mostly quite bad at doing such heuristic evaluations and that they only found between 20 and 51% of the usability problems in the interfaces they evaluated. On the other hand, we could aggregate the evaluations from several evaluators to a single evaluation and such aggregates do rather well, even when they consist of only three to five people.
KEYWORDS: *Usability evaluation, early evaluation, usability engineering, practical methods.*

INTRODUCTION

There are basically four ways to evaluate a user interface: *Formally* by some analysis technique, *automatically* by a computerized procedure, *empirically* by experiments with test users, and *heuristically* by simply looking at the interface and passing judgement according to ones own opinion. Formal analysis models are currently the object of extensive research but they have not reached the stage where they can be generally applied in real software development projects. Automatic evaluation is completely infeasible except for a few very primitive checks. Therefore current practice is to do empirical evaluations if one wants a good and thorough evaluation of a user interface. Unfortunately, in most practical situations, people actually *do not* conduct empirical evaluations because they lack the time, expertise, inclination, or simply the tradition to do so. For example, Milsted et al. [1989] found that only 6% of Danish companies doing software development projects used the thinking aloud method and that nobody used *any* other other empirical or formal evaluation methods.

In real life, most user interface evaluations are heuristic evaluations but almost nothing is known about this kind of evaluation since it has been seen as inferior by most researchers. We believe, however, that a good strategy for improving usability in most industrial situations is to study those usability methods which are likely to see practical use [Nielsen 1989]. Therefore we have conducted the series of experiments on heuristic evaluation reported in this paper.

HEURISTIC EVALUATION

As mentioned in the introduction, heuristic evaluation is done by looking at an interface and trying to come up with an opinion about what is good and bad about the interface. Ideally people would conduct such evaluations according to certain rules, such as those listed in typical guidelines documents. Current collections of usability guidelines [Smith and Mosier 1986] have on the order of one thousand rules to follow, however, and are therefore seen as intimidating by developers. Most people probably perform heuristic evaluation on the basis of their own intuition and common sense instead.

We have tried cutting the complexity of the rule base by two orders of magnitudes by relying on a small set of heuristics such as the nine basic usability principles from [Molich and Nielsen 1990] listed in Table 1. Such smaller sets of principles seem more suited as the basis for practical heuristic evaluation. Actually the use of very

Simple and natural dialogue
Speak the user's language
Minimize user memory load
Be consistent
Provide feedback
Provide clearly marked exits
Provide shortcuts
Good error messages
Prevent errors

Table 1. *Nine usability heuristics (discussed further in [Molich and Nielsen 1990]).*

complete and detailed guidelines as checklists for evaluations might be considered a formalism, especially when they take the form of interface standards.

We have developed this specific list of heuristics during several years of experience with teaching and consulting about usability engineering [Nielsen and Molich 1989]. The nine heuristics can be presented in a single lecture and explain a very large proportion of the problems one observes in user interface designs. These nine principles correspond more or less to principles which are generally recognized in the user interface community, and most people might think that they were "obvious" if it was not because the results in the following sections of this paper show that they are difficult to apply in practice. The reader is referred to [Molich and Nielsen 1990] for a more detailed explanation of each of the nine heuristics.

EMPIRICAL TEST OF HEURISTIC EVALUATION

To test the practical applicability of heuristic evaluation, we conducted four experiments where people who were not usability experts analyzed a user interface heuristically. The basic method was the same in all four experiments: The evaluators ("subjects") were given a user interface design and asked to write a report pointing out the usability problems in the interface as precisely as possible. Each report was then scored for the usability problems that were mentioned in it. The scoring was done by matching with a list of usability problems developed by the authors. Actually, our lists of usability problems had to be modified after we had made an initial pass through the reports, since our evaluators in each experiment discovered some problems which we had not originally identified ourselves. This shows that even usability experts are not perfect in doing heuristic evaluations.

Scoring was liberal to the extent that credit was given for the mentioning of a usability problem even if it was not described completely.

Table 2 gives a short summary of the four experiments which are described further in the following.

Experiment (short name)	*No. Evaluators*	*Total Known Usability Problems*	*Average Problems Found*
Teledata	37	52	51%
Mantel	77	30	38%
Savings	34	48	26%
Transport	34	34	20%

Table 2. Summary of the four experiments.

Experiment 1: Teledata

Experiment 1 tested the user interface to the Danish videotex system, Teledata. The evaluators were given a set of ten screen dumps from the general search system and from the Scandinavian Airlines (SAS) subsystem. This means that the evaluators did not have access to a "live" system, but in many situations it is realistic to want to conduct a usability evaluation in the specification stage of a software development process where no running system is yet available.

The evaluators were 37 computer science students who were taking a class in user interface design and had had a lecture on our evaluation heuristics before the experiment. The interface contained a total of 52 known usability problems.

Experiment 2: Mantel

For experiment 2 we used a design which was constructed for the purpose of the test. Again the evaluators had access only to a written specification and not to a running system. The system was a design for a small information system which a telephone company would make available to its customers to dial in via their modems to find the name and address of the subscriber having a given telephone number. This system was called "Mantel" as an abbreviation of our hypothetical telephone company, Manhattan Telephone (neither the company nor the system has any relation to any existing company or system). The entire system design consisted of a single screen and a few system messages so that the specification could be contained on a single page.

The design document used for this experiment is reprinted as an appendix to [Molich and Nielsen 1990] which also gives a complete list and in-depth explanation of the 30 known usability problems in the Mantel design.

The evaluators were readers of the Danish *Computerworld* magazine where our design was printed as an exercise in a contest. 77 solutions were mailed in, mostly written by industrial computer professionals. Our main reason for conducting this experiment was to ensure that we had data from real computer professionals and not just from students. We should note that these evaluators did not have the (potential) benefit of having attended our lecture on the usability heuristics.

Experiments 3 and 4: Two Voice Response Systems: "Savings" and "Transport"

Experiments 3 and 4 were conducted to get data from heuristic evaluations of "live" systems (as opposed to the specification-only designs in experiments 1 and 2). Both experiments were done with the same group of 34 computer science students as evaluators. Again, the students were taking a course in user interface design and were given a lecture on our usability heuristics, but there was no overlap between the group of evaluators in these experiments and the group from experiment 1.

Both interfaces were "voice response" systems where users would dial up an information system from a touch tone telephone and interact with the system by pushing buttons on the 12-key keypad. The first system was run by a large Savings Union to give their customers information about their account balance, current foreign currency exchange rates, etc. This interface is referred to as the "Savings" de-

sign in this article and it contained a total of 48 known usability problems. The second system was used by the municipal public transportation company in Copenhagen to provide commuters with information about bus routes. This interface is referred to as the "Transport" design and had a total of 34 known usability problems.

There were four usability problems which were related to inconsistency across the two voice response systems. Since the two systems are aimed at the same user population in the form of the average citizen and since they are accessed through the same terminal equipment, it would improve their collective usability if they both used the same conventions. Unfortunately there are differences, such as the use of the square[1] key. In the Savings system, it is an end-of-command control character, while it is a command key for the "return to the main menu" command in the Transport system which does not use an end-of-command key at all. The four shared inconsistency problems have been included in the count of usability problems for both systems.

Since the same evaluators were used for both voice response experiments, we can compare the performance of the individual evaluators. In this comparison, we have excluded the four consistency problems discussed above which are shared among the two systems. A regression analysis of the two sets of evaluations is shown in Figure 1 and indicates a very weak correlation between the performance of the evaluators in the two experiments (R^2=0.33, p<0.01). So while some people *are* better than others at doing heuristic evaluation of user interfaces, this tendency is not very strong. We do not have enough evidence to form a firm conclusion but it seems that it might be the case that there is very little consistency in the ability of evaluators to find usability problems. The two evaluations compared in Figure 1 concerned quite similar interfaces (both were voice response systems), and it would be a plausible hypothesis that evaluators would perform even less consistently in evaluations of more varied systems.

We should note that the evaluators in these two experiments all had the same level of usability expertise. Even though we do not have formal evidence to show this, we do believe that usability experts will be better at heuristic evaluation than average computer professionals. It is likely that experience in usability and empirical user tests provides a good background for recognizing and conceptualizing usability problems. With regard to the latter, expertise in *running* user tests would probably not be as much help as the observations of actual user behavior made by the experienced tester over the years.

[1] This key is also sometimes called the "pound key". In fact one of the inconsistency problems was that this single key had two different names in the two systems (*firkant* and *rude*, respectively, in Danish).

THE USABILITY PROBLEMS

We have already mentioned a usability problem related to the "consistency" rule in the description of experiments 3 and 4. A few other examples of usability problems are:

- The Mantel system overwrites the telephone number entered by the user so that it is no longer visible when the name and address of the corresponding subscriber is displayed (found by 95%).
- The Transport system shifts from reading submenus to reading the main menu without any pause or indication that the user is moved to another level of menu (found by 62%).
- The error message "UNKNOWN IP" in Teledata (where IP stands for information provider) can be made much more readable (found by 54%).
- Users who do not have the printed user's guide will never learn that the Savings system has an online help facility (found by 35%).
- The key to accessing certain information in the Transport system is the transport company's internal departmental organization instead of the bus numbers known by the public (found by 12%).

The validity of these usability problems is an important question: Will they in fact present problems to real users, and to what degree do they constitute the complete set of usability problems? We have not conducted traditional empirical usability tests to measure this. On the other hand, we do have two arguments in support for the validity of the problems as usability problems. The first argument is simply that most of these design issues are "obviously" problems according to established knowledge in the usability field. The second, and perhaps more convincing argument is that the very method of our experiments actually forms a kind of empirical support for the usability problems. For

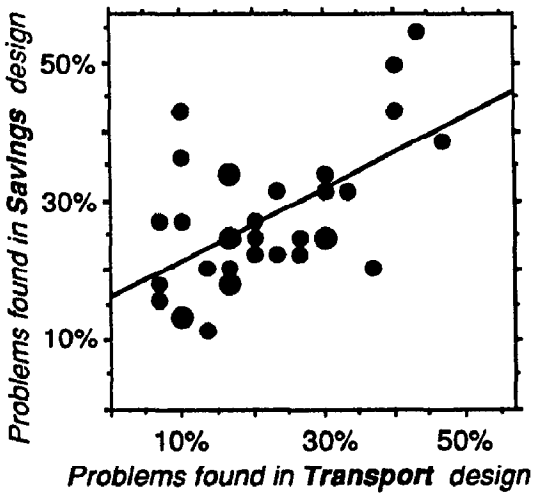

Figure 1. *Scatterplot of the proportion of usability problems found by the same evaluators in two different interfaces. The regression line has R^2=0.33 and shows that there is only a very weak correlation between the evaluators' performance in the two experiments.*

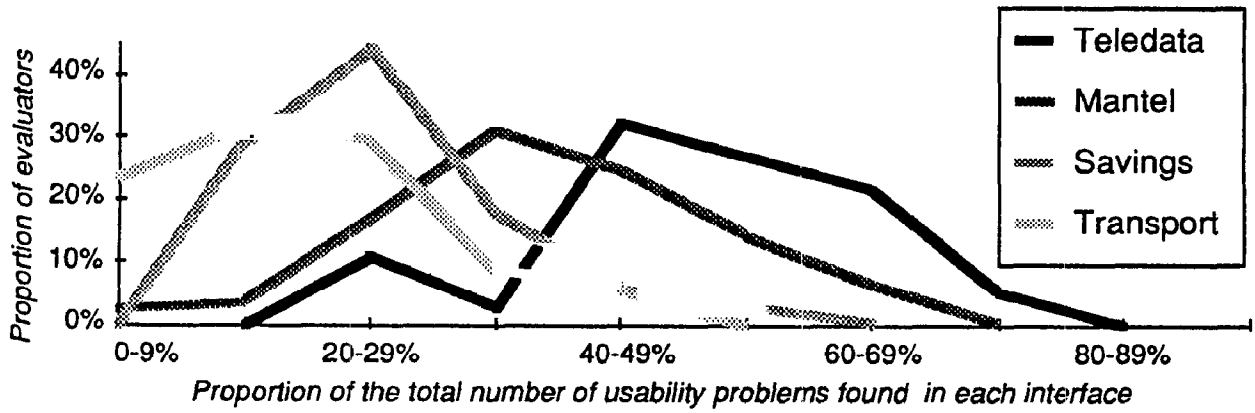

Figure 2. *Distribution for each of the four experiments of the number of usability problems found by the evaluators (expressed as percent of the total number of problems in each interface to enable comparisons).*

each system we have had at least 34 people work their way through the interface. If we view these people as experimental subjects rather than as evaluators, we realize that it is very unlikely that any of the systems would have had any major usability problem which did not bother some of these subjects enough to complain about it in their report.

In spite of these arguments, it is always impossible to know for sure whether one has found every single usability problem in an interface. It might be that the next subject would stumble over something new. Therefore we have stated for each experiment the "known" number of usability problems, and the statistics in the following sections are based on this number of known problems.

Furthermore, the usability problems of an interface do not form a fixed set in real life. For any actual use of a system by real users in a real context, only some of its potential weaknesses will surface as problems. Some aspects of a design might never bother a particular user and could therefore not be said to be "problems" as far as that user is concerned. Even so, we will still consider a design item as a usability problem if it could be expected to bother some users during some reasonable use of the system. The decision whether or not to remove the problem in a redesign should then be based on a judgement of the number of users it impacts and a trade-off analysis of whether removing it would reduce the efficiency of use or other desirable usability parameters for other users. One can only get the option to make this judgement and trade-off analysis, however, if one has identified the usability problem in the first place.

A weakness of our approach is that we only looked at individual usability "problems" in the phase of a development process where one has completed the overall design and needs to polish it. It would also be interesting to consider more holistic evaluations of entire interfaces such as those that would be required to select which of two competing products to purchase or which of two completely different design approaches to pursue. It is likely, however, that a different set of techniques will be needed for that kind of evaluation.

EVALUATION RESULTS

The most basic result from the four experiments is that heuristic evaluation is difficult. The average proportion of usability problems found was 51%, 38%, 26%, and 20% in the four experiments respectively. So even in the best case only half of the problems were found, and the general case was rather poor. Actually, even these numbers are not all that bad. Even finding *some* problems is of course much better than finding *no* problems, and one could supplement the heuristic method with other usability engineering methods to increase the total number of problems found.

Figure 2 shows the distribution of the number of problems found in each of the four experiments. We can see that the distributions as expected mostly have a shape like the normal distribution, even though the curve for the Transport experiment is somewhat skewed. In other words, most evaluators do about average, a few do very well, and a few do rather badly.

Table 3 presents information related to individual differences in the performance of evaluators. First, the number of usability problems found is expressed in percent of the

	N	Min %	Max %	D_1 %	D_9 %	Q_1 %	Q_3 %	Max/Min	D_9/D_1	Q_3/Q_1
Teledata	37	22.6	74.5	26.6	67.9	43.2	58.5	3.3	2.6	1.4
Mantel	77	0 [6.7]	63.3	23.3	53.3	30.0	46.7	∞ [9.4]	2.3	1.6
Savings	34	10.4	52.1	14.4	39.8	18.8	31.3	5.0	2.8	1.7
Transport	34	6.7	46.7	8.8	35.6	11.8	26.5	7.0	4.0	2.2
Average		**13.2**	**59.3**	**18.3**	**49.2**	**26.0**	**40.8**	**5.1**	**2.9**	**1.7**

Table 3. *Individual differences in evaluators' ability to find usability problems.*

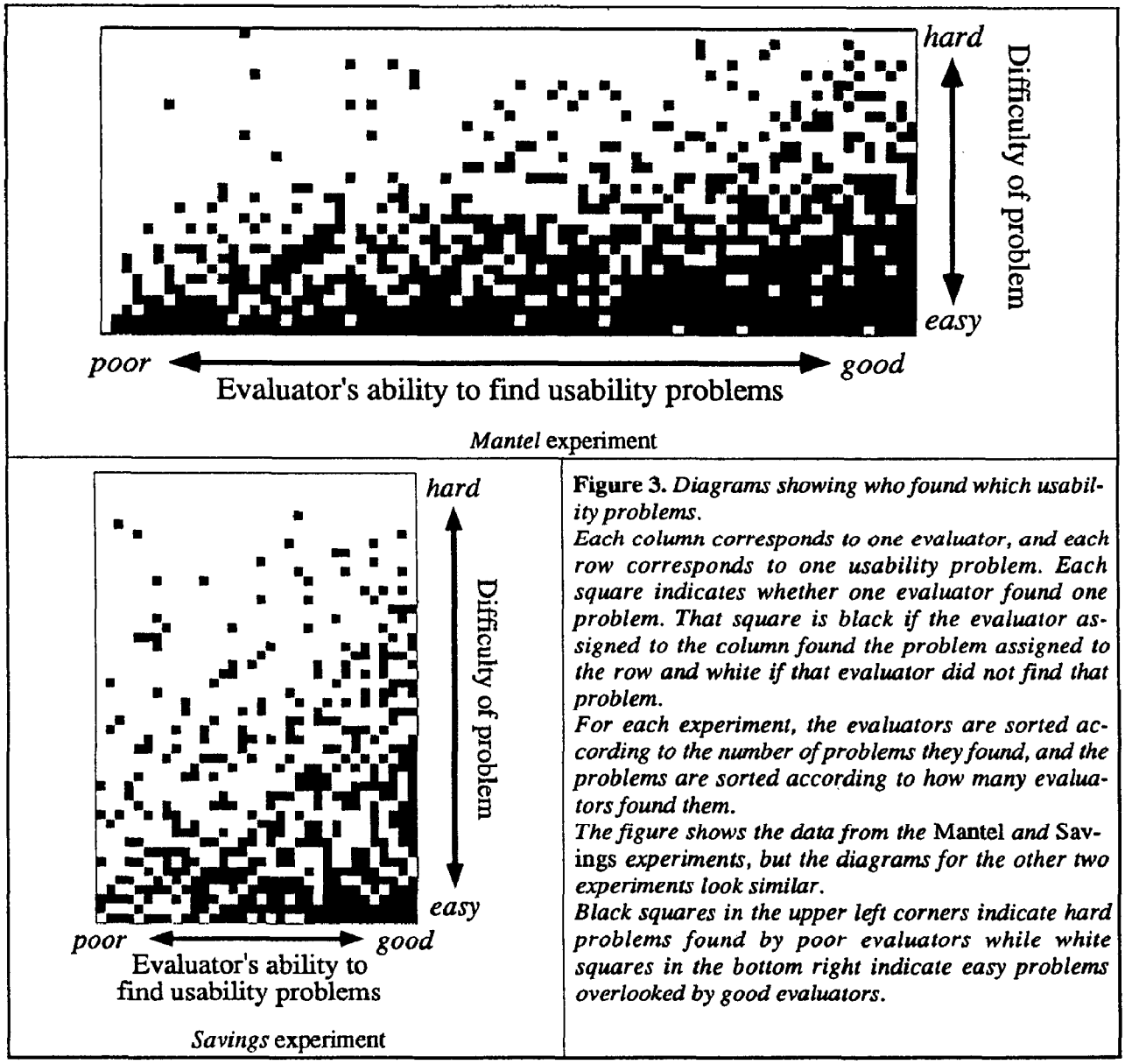

Figure 3. *Diagrams showing who found which usability problems.*
Each column corresponds to one evaluator, and each row corresponds to one usability problem. Each square indicates whether one evaluator found one problem. That square is black if the evaluator assigned to the column found the problem assigned to the row and white if that evaluator did not find that problem.
For each experiment, the evaluators are sorted according to the number of problems they found, and the problems are sorted according to how many evaluators found them.
The figure shows the data from the Mantel *and* Savings *experiments, but the diagrams for the other two experiments look similar.*
Black squares in the upper left corners indicate hard problems found by poor evaluators while white squares in the bottom right indicate easy problems overlooked by good evaluators.

total number of usability problems in each interface. For each of the five experiments, the table then lists the proportion of problems found by the worst and best evaluator, the first and ninth decile, and first and third quartile, as well as the ratios between these values. In the Mantel experiment, one of the evaluators did not find any problems at all, so the table also lists the problems found by the second worst evaluator. The Mantel experiment has been excluded from the calculation of the averages of the minimums and of the max/min ratios.

We see that the individual differences correspond to the Q_3/Q_1 ratios of about 2 listed by Egan [1988] for text editing but are lower than the ratios of 2 to 4 listed for information search and for programming. They correspond closely to the Q_3/Q_1 ratio of 1.8 for time needed to learn HyperCard programming [Nielsen 1990] by the same category of computer science students as those used in three of the four experiments.

We see from Tables 2 and 3 that some systems are easier to evaluate heuristically than others. One interesting trend from Table 3 is that the individual differences between evaluators are larger the more difficult the interface is to evaluate. Table 2 further shows that the voice response systems were especially hard to evaluate. The problem with heuristic evaluation of voice interfaces is that they have an extremely low persistence [Nielsen 1987] because all system messages are gone as soon as they are uttered. This again means that evaluators get no chance to ponder details of the interface design at their leisure.

In general, there were rather few false positives in the form of evaluators stating that something was a usability problem when we would not classify it as such. Therefore we have not conducted a formal analysis of false positives. For a practical application of heuristic evaluation, false positives might present a problem to the extent that one evaluator's finding of a false positive could sidetrack the discus-

sion in a development group. Our experience is that a given false positive normally is not found by more than a single evaluator, so the other members of the development group should be able to convince the finder of the false positive that it is not a real usability problem. If not, then an empirical test could serve as the ultimate arbiter. We would in general recommend that one does not rely exclusively on heuristic evaluation during the usability engineering process. Such methods as thinking aloud should be used to supplement the heuristic evaluation results in any case.

We should note that we have only tested heuristic evaluation of fairly small-scale interfaces. We do not know what happens during the heuristic evaluation of much larger interface designs. Furthermore, we studied evaluations of complete designs in the form of paper prototypes or running systems. It is also of interest what happens during the "inner loop" of design [Newell and Card 1985] where a designer rapidly evaluates various alternative subdesigns before they are finalized in a complete design. It is likely that such evaluations are often heuristic in nature, so some of the same results may apply.

AGGREGATED EVALUATIONS

Figure 2 and Table 3 show that some evaluators do better than others. One might have supposed that the difference in performance between evaluators was due to an inherent rank ordering of the difficulty of finding the usability problems, such that a "good" evaluator would be able to find all the easy problems found by a "poor" evaluator as well as some additional, harder problems. Figure 3 shows, however, that this is not the case. Even poor evaluators can sometimes find hard problems as indicated by the black squares in the upper left part of the diagrams. And good evaluators may sometimes overlook easy problems as indicated by the white squares in the lower right part of the diagrams. In other words, the finding of usability problems does not form a perfect cumulative scale (a Guttman[2] scale [Guttman 1944]).

Because of this phenomenon, we have the potential for dramatically improving the overall result by forming *aggregates* of evaluators since the "collected wisdom" of several evaluators is not just equal to that of the best evaluator in the group. Aggregates of evaluators are formed by having several evaluators conduct a heuristic evaluation and then collecting the usability problems found by each of them to form a larger set.

For this aggregation process to work, we have to assume that there is some authority that is able to read through the reports from the individual evaluators and recognize the usability problems from each report. This authority could be a usability expert or or it could be the group itself during a meeting of the evaluators. We have not tested this assumption empirically but it seems reasonable for the kind of usability problems discussed in this paper since they are of a nature where they are "obvious" as soon as somebody has pointed them out.

Our experience from conducting the four experiments and discussing them with the evaluators indicates that people are usually willing to concede that something is a usability problem when it is pointed out to them by others. At least for the kind of usability problems considered in this paper, the main difficulty lies in finding them in the first place, not in agreeing on the aggregated list.

On the basis of our data showing which evaluators found which usability problems, we have constructed hypothetical aggregates of varying sizes to test how many problems such aggregates would theoretically find. The aggregates were not formed in a real project but given our assumption of a perfect authority to form the conclusions, that should make no difference. For each of our four experiments, aggregates were formed by choosing the number of people in the aggregate randomly from the total set of evaluators in that experiment. For each experiment, it would of course have been possible to select an optimal aggregate of the better evaluators but in a real company one would not have that luxury. Normally one would have to use whatever staff was available, and that staff would have been hired on the basis of many other qualifications than their score in heuristic evaluation experiments. And in any case, Figure 1 indicates that people who are good evaluators in one experiment may not be all that good in the next experiment.

Figure 4 shows the results from selecting random aggregates of evaluators. The figure shows the average number of usability problems found by each size of aggregate. These averages were calculated by a Monte Carlo technique where we selected between five and nine thousand

[2] The evaluations do approximate a Guttman scale with an average Guttman reproducibility coefficient R = 0.85 (coefficients ranging from 0.82 to 0.87). The average minimal marginal reproducibility, MMR is 0.80 (ranging from 0.79 to 0.82), however, indicating that the scale is not truly unidimensional and cumulative since the coefficient of scalability is only 0.06. The Guttman coefficient indicates the degree to which the data follows a unidimensional cumulative scale, with a value of 1 indicating a perfect scale. The Guttman coefficient of 0.85 shows that only 15% of the data deviates from that expected of such a perfect scale. But the minimal marginal reproducibility indicates the degree to which the individual values could be predicted from the average values even disregarding potential scaling properties. From knowing e.g. that a certain usability problem was only found by 20% of the evaluators, we would be able to correctly predict 80% of the data for that problem without taking that evaluators general problem-finding abilities into account by just predicting for each evaluator that he or she would not find the problem. So the assumption of strict ordering only gains us an improvement from 80% to 85%, indicating that it has poor explanatory powers. In any case, it is the deviation of 15% from the Guttman scale which allows us to form the aggregates we discuss here.

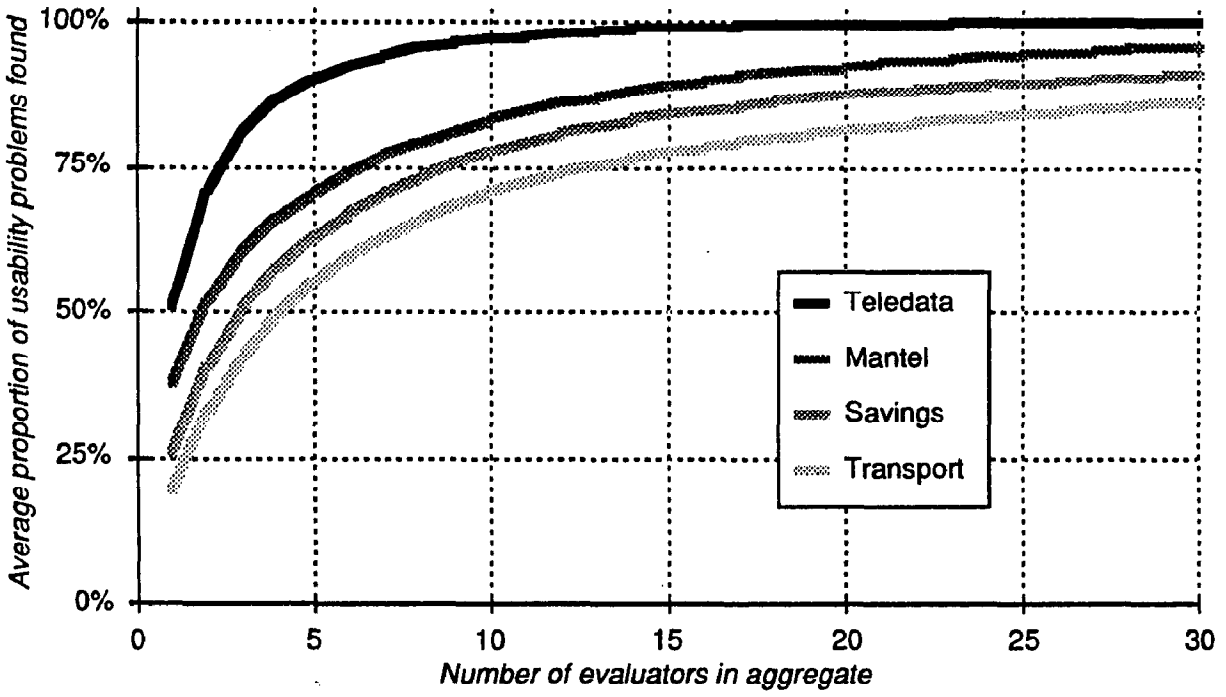

Figure 4. *Proportion of usability problems found by aggregates of size 1 to 30.*

random aggregates for each aggregate size and experiment. Table 4 gives the exact numbers for selected sizes of aggregates.

It is apparent from Figure 4 that the curves for the four experiments have remarkably similar shapes. Each curve rises drastically in the interval from one evaluator to about five evaluators, it then flattens out somewhat around the interval from five to ten evaluators, and the point of diminishing returns seems to have been reached at aggregates of about ten evaluators. It is interesting to see that even for the Transport interface which was the hardest to analyze, aggregates of five evaluators are still able to find more than half of the usability problems. In general, we would expect aggregates of five evaluators to find about two thirds of the usability problems which is really quite good for an informal and inexpensive technique like heuristic evaluation.

For the aggregated evaluation to produce better results than the individual evaluations, it is likely that the evaluators should do their initial evaluations independently of each other and only compare results *after* each of them has looked at the design and written his/her evaluation report. The reason we believe this is that evaluators working together in the initial evaluation phase might tend to bias each other towards a certain way of approaching the analysis and therefore only discover certain usability problems. It is likely that the variety in discovered usability problems apparent in our experiments would have been smaller if the evaluators had worked in groups. And it is of course the variety which is the reason for the improvement one gets from using aggregates of evaluators.

Aggregate:	*1*	*2*	*3*	*5*	*10*
Teledata	51%	71%	81%	90%	97%
Mantel	38%	52%	60%	70%	83%
Savings	26%	41%	50%	63%	78%
Transport	20%	33%	42%	55%	71%

Table 4. *Average proportion of usability problems found in each of the four interfaces for various sized aggregates of evaluators.*

CONCLUSIONS

This study shows that heuristic evaluation is difficult and that one should not rely on the results of having a single person look at an interface. The results of a heuristic evaluation will be much better if you have several people conduct the evaluation, and they should probably do so independently of each other. The number of usability results found by aggregates of evaluators grows rapidly in the interval from one to five evaluators but reaches the point of diminishing returns around the point of ten evaluators. We recommend that heuristic evaluation is done with between three and five evaluators and that any additional resources are spent on alternative methods of evaluation.

Major advantages of heuristic evaluation are:
- It is cheap.
- It is intuitive and it is easy to motivate people to do it.
- It does not require advance planning.
- It can be used early in the development process.

A disadvantage of the method is that it sometimes identifies usability problems without providing direct suggestions for how to solve them. The method is biased by the current mindset of the evaluators and normally does not generate breakthroughs in the evaluated design.

ACKNOWLEDGEMENTS

The authors would like to thank Jan C. Clausen, John Schnizlein, and the anonymous *CHI'90* referees for helpful comments.

REFERENCES

1. Egan, D.E. Individual differences in human-computer interaction. In: M. Helander (Ed.): *Handbook of Human-Computer Interaction*. Elsevier Science Publishers, Amsterdam, 1988, pp. 543-568.
2. Guttman, L. A basis for scaling qualitative data. *American Sociological Review* **9** (1944), 139-150.
3. Milsted, U., Varnild, A., and Jørgensen, A.H. Hvordan sikres kvaliteten af brugergrænsefladen i systemudviklingen ("Assuring the quality of user interfaces in system development," in Danish). *Proc. NordDATA'89 Joint Scandinavian Computer Conference* (Copenhagen, Denmark, 19-22 June 1989), 479-484.
4. Molich, R. and Nielsen, J. Improving a human-computer dialogue: What designers know about traditional interface design. *Communications of the ACM* **33**, 3 (March 1990).
5. Newell, A. and Card, S.K. The prospects for psychological science in human-computer interaction. *Human-Computer Interaction* **1**, 3 (1985), 209-242.
6. Nielsen, J. Classification of dialog techniques: A CHI+GI'87 workshop, Toronto, April 6, 1987. *ACM SIGCHI Bulletin* **19**, 2 (October 1987), 30-35.
7. 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.
8. Nielsen, J. Assessing the learnability of HyperCard as a programming language. Manuscript submitted for publication 1990.
9. Nielsen, J. and Molich, R. Teaching user interface design based on usability engineering. *ACM SIGCHI Bulletin* **21**, 1 (July 1989), 45-48.
10. Smith, S.L. and Mosier, J.N. *Guidelines for Designing User Interface Software*. Report **MTR-10090**, The MITRE Corp, Bedford, MA, August 1986.