Thinking-aloud in user interface design: A method promoting cognitive ergonomics

Article *in* Ergonomics · April 1990

DOI: 10.1080/00140139008927157 · Source: OAI

1 author:

Anker Helms Jørgensen
IT University of Copenhagen
36 PUBLICATIONS **838** CITATIONS

SEE PROFILE

ERGONOMICS, 1990, VOL. 33, NO. 4, 501–507

Thinking-aloud in user interface design: a method promoting cognitive ergonomics

ANKER HELMS JØRGENSEN

Department of Psychology, Copenhagen University,
Njalsgade 88, DK-2300 Copenhagen S, Denmark

Keywords: Usability engineering; User interfaces; Evaluation method; Thinking-aloud.

The thinking-aloud method has been used successfully in user interface research. The present study investigates the application of the method for user interface design. The study is based on interviews with nine systems designers who had used the method in design practice. Their application of the method was in general very successful. This paper focuses on a unique property of this method: its inherent promotion of cognitive ergonomics due to the timely, genuine and applicable feedback to the designers in the design context.

1. Introduction

One of the most frequent statements in scientific papers in the field of human–computer interaction is 'User interface design is more of an art than a science'. This is not to say that science in the field has not provided applicable products for user interface designers—it certainly has. But applications in design practice have been sparse and the results disappointing.

The products of research fall largely into two categories: formal approaches and informal guidelines. The formal approaches comprise formal languages such as the Command Language Grammar developed by Moran (1981), and cognitive models such as Cognitive Complexity Theory by Kieras and Polson (1985). These approaches tend to be restricted in scope, fail to incorporate the underlying psychological issues sufficiently, and lead to unwieldy specifications (see for example Sharratt 1987).

The informal guidelines are brief statements covering all aspects of the user interface, such as dialogue design, screen layout and phrasing of on-line help. Several collections of such guidelines exist; the report compiled by Smith and Mosier (1986) is the most comprehensive. Although these guidelines are directly addressed to system designers, the designers experience considerable difficulty in applying them as documented by Mosier and Smith (1986). One of the main problems is that the guidelines are either too specific or too general for the design problem at hand. These findings are in line with previous studies of the application of human factors guidance, see for example (Meister and Farr 1967). Given this state of affairs, on what grounds are user interfaces in fact designed? According to the handful of empirical studies of interface design practice, experience and intuition play a major role. This is illustrated in the following quote from the study by Hammond *et al.* (1983), where a designer of a word processing system states:

> If I'm copying a box on the screen for deleting or copying, is it better to have that box video reversed, or is it better to have the first and last character blinking, or is it better to have it underscored or higher intensity? We just marked the first and last with a blink character, the most unobtrusive technique, OK? (p. 43).

This dependency on individual designers leads to the 'subjective' approach to usability, i.e., the role of personal characteristics such as skill, experience, intuition, and attitude—in contrast to the formal 'objective' approaches.

I have elaborated on this contrast between these approaches to user interface design in earlier work (see Jørgensen 1986 a). There I reported on a classical experiment on the structure of the users' task in an interactive system. The experiment provided a neat result on task structure—the 'objective' evidence (Jørgensen 1986 b). However, the experiment turned out to have a most useful side-effect. As the experimenter I ran the subject sessions in which most of the subjects commented spontaneously on the system, i.e., unknowingly applied the 'thinking-aloud' method. To put it mildly, I was often taken completely by surprise by the subjects' reactions to the system—which I myself had designed. Their views differed so profoundly from my own conceptions. However, a little analysis always revealed the logic underlying the users' views. Running the subjects provided timely, genuine and relevant feedback. In fact it changed my view substantially on user interface design and thus inherently promoted cognitive ergonomics.

This observation raises a number of questions: Is this experience valid for other designers? Will the thinking-aloud method have a similarly profound effect on other designers? This study sets out to answer these questions. The study is based on interviews with nine systems designers who had used the method in the design of computer systems.

Four types of issues were addressed in the interviews: the organizational issues (e.g., how is the method introduced in the organization?), the methodological issues (e.g., is a pilot test required?), the usability issues (e.g., how much is the usability enhanced when the errors revealed are eliminated?), and the psychological issues (e.g., how do designers cope with being confronted with the users' difficulties?). The emphasis in this paper is on the psychological issues; the results addressing the other issues are reported elsewhere (Jørgensen 1989).

2. The thinking-aloud method

The thinking-aloud method consists in having a user working with a computer system (prototype, paper mock-up or documentation) while 'thinking-aloud', i.e., spontaneously (or prompted) verbalizing ideas, facts, plans, beliefs, expectations, doubt, anxiety, etc. that comes to mind during the work. Typically a scenario is developed for the tests, i.e., an artificial work context with specific tasks that can be accomplished by means of the system.

The thinking-aloud method has its roots in cognitive psychology where it has been the subject of a heated debate between its critics, e.g., Nisbett and Wilson (1984) and its proponents, e.g., Ericsson and Simon (1984). Suffice it to say that the purpose of cognitive psychology is to study human cognitive processes, while in user interface design the purpose is to identify errors in the user interface of a system. How the users' cognitive processes take place is of no interest to the designers (but has of course implicit implications for them).

The thinking-aloud method was introduced in research into user interfaces by Lewis (1982) and colleagues, and it has since been applied successfully here. The method has also been applied by Human Factors practitioners, see for example Clark (1981).

3. The designers, the projects and the systems

Nine system designers who had used the thinking-aloud method were interviewed in their workplace. Five of the designers worked in industry, two in a R&D organization and two were computer science students using the method in their MSc. thesis project. They were all experienced in systems design. They qualified for the investigation by having used the thinking-aloud method. The interviews lasted about 1·5 h and were recorded on tape. These were later fully transcribed and analysed in detail. In the following the context of the application of the method is outlined for the nine designs (organizational setting, type of system, number of users tested, etc.).

Designers D1 and D2 were M.Sc. students who applied the method in their thesis work. D1 ran two testing rounds with a total of nine users. He modified the system—a small book-keeping system—after the first round, based on the results of the test. The first round revealed 38 errors, the second only ten. D2 used the method on an electronic mail system. He also ran two rounds with seven users in all.

Designers D3 and D4 worked in a R&D organization on a knowledge-based retrieval system running on a graphical workstation. D3 carried out a think-aloud test on a paper mock-up as soon as the functionality and user interface had been designed. He ran four users in two rounds. When a working prototype had been developed, D4 ran one round with five users.

Designer D5 undertook a fairly thorough thinking-aloud test of a bulletin board system in a bank: two rounds with a total of 12 users. The first round revealed 20 errors while the second only revealed five.

Designer D6 served as an in-house consultant in a large software house to a project developing a journal system. One round with seven users revealed 44 errors.

Designer D7 used the method for testing a 20-page tutorial for a word processor. She used the method in an *ad hoc* fashion and spent only a small amount of time on the test.

Designer D8 had introduced the method in a large bank. After initial trials on selected systems the designers managed to 'sell' the method to upper management as well as to many individual designers. In this effort a video tape showing a user having severe trouble with a 'simple' system proved extremely useful.

Designer D9 had heard about the method at an in-house seminar and applied it when he had the chance to do so. As a project manager for the development of a system for setting up bank loans in foreign currencies, he had a working prototype developed for the sole purpose of the thinking-aloud tests. It was tested in one round with three users. This revealed 32 errors, most of which were corrected in the development of the real system.

4. Results

In this paper I will focus on the results regarding the unique property of the method: the inherent promotion of cognitive ergonomics—i.e., the psychological issues. The results on the organizational, the methodological, and the usability issues are reported elsewhere (Jørgensen 1989).

A major reason for the success of the method is that all the designers except one (D9) ran the subject sessions themselves. They therefore received direct feedback—which can be quite demanding on the part of the designer. Andersen (personal communication) reported on a case where a thinking-aloud test had to be interrupted because the designer got very cross due to the 'stupid' user. None of the designers here reported problems of this kind—although several of them at times felt the users 'were a little slow' or 'made the same error repeatedly'.

Designer D9 who did not run the sessions himself expected that he would not be able to 'keep his hands off the keyboard'—and consequently asked Quality Assurance colleagues to run the sessions. He received feedback by talking to the users after the sessions and via written and verbal reports from the QA people.

It is appropriate here to mention the motivation of the designers. They were undoubtedly more motivated towards usability than average designers in that they themselves had taken the initiative to run the tests. The extent of their motivation is illustrated by designer D4 who developed the knowledge-based retrieval system. Due to delays in the project there was hardly time to do the tests. But designer D4 was so keen to carry them out after having programmed for six months that he managed to get time from project management. But the response times—up to 15 min—rendered realistic tests impossible! However, he did not give up. He quickly coded a small 'cheat' version of the system, having the same user interface but limited functionality and data structures. In addition he coded the logging facilities. He then conducted five user sessions.

Designer D6, who served as a consultant to a development project, initially experienced some resistance from the project manager because she only reluctantly had accepted the tests due to tight time schedules. However, as the tests went on, she gradually changed her mind. She received a taste of the direct feedback by participating in the social events between the tests (e.g., having lunch with the users). She ended up being so interested that she modified the system in the evening between the two days of testing.

In many cases the tests provided clearcut answers to design problems that had been discussed at length at the design stage. An example was given by designer D9 who developed the banking system for setting up loans in foreign currencies. The system ran on new hardware that only allowed entry of digits into the left-hand side of a field. This feature coincided with another feature and this had catastrophic consequences. The bank only accepted certain rounded figures for loans, e.g., in units of hundreds for British pounds and thousands for French francs. These constraints were mediated to the users by displaying the required minimum number of trailing zeroes in the right hand side of the field. Thus, if a user was establishing a loan of 2000 pounds (GBP) the following would appear after the user had typed the digits '20':

Amount: 20 00 GBP

The reason for the space between the '20' and the '00' is that the system must allow ample space for a large number of trailing zeroes for certain currencies, e.g., Italian lire. This display is a clearcut violation of the gestalt law of proximity and it caused severe trouble to all the users. This was indeed a substantial error in a banking system. Designer D9 stated that this feature itself would have caused the customers to reject the system. The design feature had in fact been discussed several times during the design and the designers had even committed the same error when they had 'put themselves in the users' shoes'. However, they had not managed to come up with an appropriate solution. The think-aloud test made them see clearly how to get around the problem: to underscore the field.

Designer D4 experienced an interesting case that reveals the strength of the thinking-aloud method. A user was creating and deleting windows in the retrieval system in a completely unexpected manner. Thus in one case, an empty window appeared on the screen that should have been half full of text. D4 thought it was a system error and started to take remedial action. However, the system seemed to run

correctly. D4 just couldn't figure out what was 'wrong'—until the user came up with the right explanation, although he knew 'much less' than the designer. Designer D4 was so constrained by a certain way of thinking about the system that he could not transcend his own understanding. The point is that the user's knowledge perspective is completely different from that of the designer. These window-managing design features had also been discussed at length during the design without any clear solutions.

The strongest surprise was experienced by designer D1. He had designed a very small book-keeping system—in fact only two screens. He conducted two test rounds. In the first he had called in six users. While preparing the test sessions he thought that he had called in the users in vain:

> There can't be any problems in such a simple system. This is chickenfeed. There is nothing to misunderstand.

He was completely taken by surprise: 38 errors were revealed. In fact he only counted errors that were made by two or more users and errors that had catastrophic consequences. He stated:

> Even today I'm kind of shocked by the 38 errors when I initially thought there wouldn't be any trouble. As a designer it has been incredibly instructive to go through this process!

These are examples of one of the strengths of the thinking-aloud method: the surprises the designers get by seeing users' behaviour differ so greatly from what they had expected. One of the main problems in designing user interfaces is that it is almost impossible to imagine the users' potential conceptions and misunderstandings of the system—not least for the designers whose views in many cases are strongly biased by implementation considerations.

It is noteworthy that none of the designers had formal training in Human Factors or psychology. Some of them had attended 3-day usability engineering courses, and some had only heard about the method. However, many of them consulted a popular Danish book on usability engineering (Beyer *et al.* 1986) which describes the method in some detail.

In order to consolidate the designers' views on the thinking-aloud method I asked them for their views on a number of issues. First, the designers found the tests worthwhile in spite of the trouble they had experienced, both in setting up the tests and in observing the users having difficulties with their 'easy-to-use' system. The answers to this question were given very promptly.

All the designers were taken by surprise during the tests, as I was myself while running the sessions with the task structure experimental system. In fact, six of the nine designers felt that this feedback had changed their attitudes towards system development so that they would approach future designs in a more user-oriented and iterative manner.

In addition, all the designers stated that they would run thinking-aloud tests again. Again, their responses were very prompt. They also answered positively to the question whether they would recommend the method to colleagues. Most of them had in fact already done so—and two of them had even written articles in the house journal. The final question was whether they would plan for the use of the method if they were project managers. Again, all the answers were positive and given without hesitation. However, a few qualified their answers by stating 'Yes, if the system was intended for end-users!'

5. Discussion

One of the most striking results of the application of the thinking-aloud method is the value of the feedback the designers get in context, thereby overcoming one of the main problems with the formal methods (limitation of scope) and informal guidelines (mismatch between the normative guidelines and the design context at hand). Whiteside and Wixon (1987) put it thus:

> Usability, ultimately, lives in user experience. Therefore usability engineering must be grounded in experience. Usability engineering provides tools for uncovering user experience (p. 17).

The thinking-aloud method is an excellent tool for uncovering user experience in that it generates highly specific and applicable information on the spot. The translation to more general usability engineering principles is however, left to the designers. In contrast, the formal models and informal guidelines require translation of general principles to the specific design context. These approaches do not exclude each other; on the contrary, the models and guidelines may help individual designers organize their fragmented experience and knowledge from thinking-aloud sessions into a more coherent body of knowledge. And vice versa, the experience of individual designers can trigger a more profound understanding of the principles behind the models and guidelines.

Finally, the method is in no way a substitute for other activities in system development such as task analysis or field testing, let alone systematic studies of specific design features aiming at finding optimal design features. The method is very flexible and can be applied at any time during the development phase and with any number of users. Thus, the method facilitates the 'quick and dirty' approach to user interface design that pervades current design practice, which—according to Bellotti (1988)—is the only feasible way due to the practical circumstances of user interface design.

6. Conclusion

This study has shown that the thinking-aloud method is a successful instrument in user interface in systems development as applied by motivated system designers. It is fairly straightforward to use and can even be applied with little or no human factors training. The method has the property of inherently promoting cognitive ergonomics by providing timely, genuine and applicable feedback to the system designers.

Acknowledgements

Thanks are due to the nine designers and also to Rolf Molich and Jakob Nielsen for comments on an earlier version of this paper.

References

ANDERSEN, S. 1984, personal communication.

BELLOTTI, V. 1988, Implications of current design practice for the use of HCI techniques, in D. M. Jones and R. Winder (eds) *People and Computers IV* (Cambridge University Press, Cambridge), 13–34.

BEYER, P., CARSTENSEN, P., JØRGENSEN, A. H., MOLICH, R. and PEDERSEN, F. H. 1986, *Brugervenlige edb-systemer* (User-friendly computer systems) (Teknisk Forlag, Copenhagen).

CLARK, I. A. 1981, Software simulation as a tool for usable product design. *IBM Systems Journal*, **20**, 272–293.

ERICSSON, K. A. and SIMON, H. A. 1984, *Protocol analysis: Verbal reports as data* (MIT Press, Cambridge, MA).

HAMMOND, N., JØRGENSEN, A. H., MACLEAN, A., BARNARD, P. and LONG, J. 1983, Design practice and interface usability: evidence from interviews with designers, *Proceedings of CHI '83*, Boston, December 1983, (ACM, New York), 40–44.

JØRGENSEN, A. H. 1986 a, Two approaches to usability: the objective and the subjective, *Proceedings of Work with Display Units*, Stockholm, May 1986, 940–943.

JØRGENSEN, A. H. 1986 b, An experiment on the task structure in interactive computer systems, *Proceedings of Work with Display Units*, Stockholm, May 1986, 777–780.

JØRGENSEN, A. H. 1989, Using the thinking-aloud method in systems development, in G. Salvendy and M. J. Smith (eds) *Designing and Using Human Computer Interfaces and Knowledge Based Systems* (Elsevier, Amsterdam), 743–750. (Proceedings of HCI International '89: 3rd International Conference on Human–Computer Interaction, Boston, 18–22 September 1989).

KIERAS, D. and POLSON, P. G. 1985, An approach to the formal analysis of user complexity, *International Journal of Man–Machine Studies*, **22**, 365–394.

LEWIS, C. 1982, Using the 'thinking-aloud' method in cognitive interface design, IBM Research Report RC 9265, 2/17/82 (IBM T. J. Watson Research Center, Yorktown Heights, NY).

MEISTER, D. and FARR, D. E. 1967, The utilization of human factors information by designers, *Human Factors*, **9**, 71–87.

MORAN, T. 1981, The Command Language Grammar: a representation of the user interface of interactive computer systems, *International Journal of Man–Machine Studies*, **15**, 2–50.

MOSIER, J. N. and SMITH, S. L. 1986, Application of guidelines for designing user interface software, *Behaviour and Information Technology*, **5**, 39–46.

NISBETT, R. E. and WILSON, T. D. 1984, Telling more than we can know: verbal reports on mental processes, *Psychological Review*, **84**, 231–259.

SHARRATT, B. D. 1987, Top-down interactive systems design: some lessons learnt from using Command Language Grammar, in H.-J. Bullinger and B. Shackel (eds) *Proceedings of Human–Computer Interaction—INTERACT '87*, (Elsevier, North-Holland), 395–399.

SMITH, S. L. and MOSIER, J. N. 1986, Guidelines for designing user interface software, Report ESD-TR-86-278, (The MITRE Corporation, Bedford, MA).

WHITESIDE, J. and WIXON, D. 1987, The dialectic of usability engineering, in H.-J. Bullinger and B. Shackel (eds) *Proceedings of Human–Computer Interaction—INTERACT '87*, (Elsevier, North-Holland), 17–20.