RC 9265 (#40713) 2/17/82
Computer Science 6 pages

Research Report

Using the "Thinking-aloud" Method in Cognitive Interface Design

Clayton Lewis

IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598

IBM Research Division
San Jose · Yorktown · Zurich

RC 9265 (#40713) 2/17/82
Computer Science 6 pages

Using the "Thinking-aloud" Method in Cognitive Interface Design

Clayton Lewis

IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598

Abstract: "Thinking-aloud" is a method for studying mental processes in which participants are asked to make spoken comments as they work on a task. The method is appropriate for studying the cognitive problems that people have in learning to use a computer system. This note discusses the strengths and weaknesses of the method, and gives some suggestions about its use based on laboratory experience at Yorktown.

In designing the physical interface of a computing system the problem is to match the characteristics of a keyboard or display to the characteristics of fingers and eyes. In designing the *cognitive* interface the problem is to fit the informational aspects of the system, including menus, messages, training manuals, and much more, to the *mental* characteristics of users. Memory demands, problem-solving or inference that may be required of the user, possible interference from prior experience, and other cognitive costs become central.

In this cognitive domain of system design we are hampered by very limited understanding of the mental machinery to which systems must be fit. For the foreseeable future we must rely on an essentially empirical approach in which an interface is designed, some form of mock-up is devised and tried out with real users, the design is modified to reduce the difficulties users are observed to encounter, and the process is repeated.

In this iterative approach to design it is necessary to have effective means of identifying the weak and strong points of a design, and to learn enough about *why* a given design feature is good or bad to guide modifications. The *thinking-aloud* method offers some advantages for this purpose.

This technique was developed and popularized within cognitive psychology by Newell and Simon (1972) for the study of problem-solving. It has come into wider and wider use in investigating a wide range of mental processes, from memory retrieval to reading comprehension. IBM Human Factors groups are using the method to study problems from customer setup to application program operation.

At Yorktown, Bob Mack, Jack Carroll, Karen Greer, and I have been using the method to study how people learn to use text-processing systems. In this note I want to share with people who are not familiar with the method some of what we have learned in using the method in our laboratory. Ericsson and Simon (1980) can be consulted for an in-depth discussion of the method and associated theoretical issues.

COMPARISON WITH OTHER METHODS

As we apply it, the thinking-aloud method is a form of detailed observation of what the user is doing with the system and documentation. As in other observational studies, we use video recording to capture what is happening on the screen of the system. The special feature of the thinking-aloud method is that while working or reading the participant is asked to keep up a running commentary of his or her thoughts: what s/he is trying to do, what questions or confusions s/he is concerned about, what s/he expects will happen next, and the like. An observer is present to prompt the participant to keep up the flow of comments.

Let's compare this approach with more traditional observational methods, and with methods relying on actual performance measures to evaluate a design.

Thinking-aloud has some disadvantages and limitations:

Realism: The presence of the observer, and the demand to talk while working may influence the participant. Of course, any laboratory work arrangement is unlike a real job for the participant, but the thinking-aloud situation may be particularly unusual. Our experience is that participants soon get used to the presence of the observer and the demand to talk. But their behavior is altered. It seems that performance may be unusually *good* because of participants being more conscientious, persistent, and methodical than they might be in an unobserved work situation, but we do not have direct evidence of this.

Labor: Since an observer must work with each participant, it is hard to collect data from large groups of participants. Also, the analysis of observations is laborious because of the amount of detailed information that must be scanned for points of importance.

Accuracy: Some proportion of comments given by participants will be inaccurate, since people are not aware of all aspects of their mental processes and sometimes seem to make up reports that fit what they find themselves doing. However, the thinking-aloud method seems to suffer less from this than some other methods, such as post-mortem interviewing, since participants are free to report only what they find easy to talk about.

Timing: Thinking-aloud will throw off measurements of reaction time or task completion time.

Summarization: The method does not produce one or a small number of numerical dependent measures that can easily be summarized with standard statistical techniques. Instead, one obtains a detailed record of very complicated behavior, which must be analyzed for evidence about the effects of a wide variety of system features.

Given these limitations, the technique does not lend itself to a "benchmark" kind of assessment in which a pass-fail rating must be assigned to a design on the basis of a quantitative criterion. Instead, it is useful in a "find and fix" investigation, where specific information about design features and how they work or fail to work is needed.

Against these limitations, thinking-aloud offers some unique advantages.

Pinpointing problems: Often users have trouble because of an unfamiliar or misleading word, like "default" or "paginate". These difficulties show up clearly in their comments, so that one learns not just that a certain paragraph or menu is causing delays or errors but specifically what must be fixed. More generally, key concepts such as that of a cursor or a command with deferred execution may cause trouble that would be difficult to trace without participants' comments.

Finding WHY a problem occurs: It is often easy to see that a user is having trouble doing something, but hard to see what has caused the trouble. The participant's comments often make it obvious what is wrong, so that a remedy can be found. Here are a few examples.

One learner finished an exercise in which a lease was produced as an example. The lease was a real mess, for the learner had made many errors. But the learner went right on to the next training unit. Why? Had she given up? Did she think she understood her mistakes? Her comment showed clearly what was needed in the manual: "A lease is a legal document, so I suppose it is supposed to look like this." The manual needed a picture of the finished product for learners to check their work.

Another learner stopped in the middle of an apparently clear sequence of instructions. The next step was a key press that she certainly knew how to perform. Why would she not go on and do that next step? Her comments showed that her understanding of the *goal* of the exercise made her think the next step was incorrect. So the steps, where the problem seemingly occurred, were in fact OK, but the description of the purpose of the exercise needed work.

Another learner was trying to insert a few lines of text at the end of a document. He paged back and forth through the manual for some time, and never found the appropriate method. Why? He was looking for a way to "move the end of the file down", not a way to insert lines.

Catching problems when they occur: Often performance or observational testing is followed by in-depth interviews with the test participants. While much can be learned this way, it is *NOT* a substitute for the thinking-aloud method. People just can't remember enough about the problems they had to give the kind of detailed information needed to pinpoint them. It is especially hard for someone to tell why something was confusing if s/he later comes to understand it.

Detecting "minor" problems: Many problems may not have measurable impact on aggregate learning time or task completion, given the amount of noise in quantitative performance data. In the thinking-aloud procedure, participants will often remark on the annoyance or confusion of such minor problems, which can then be isolated and repaired. Why bother with minor problems? First, they are irritating and give users a poor impression of the system. Second, seemingly minor problems sometimes combine with others to form major problems for some users. For example, confusion about whether to use upper or lower case in typing commands may rarely cause trouble in itself, if the system accepts either case. But if a command fails to work users will often assume that it was the case used that caused the problem. This distracts them from finding the real problem.

Learning the facts of cognitive life: besides providing useful specific information about an interface, thinking-aloud data can give a general sense of how users approach tasks. We have learned a good deal about our users, who are experienced office workers but new to computers, from their comments. For example, they consistently try to make sense of what they are doing, and are not content simply to follow instructions "by the numbers." Because they know so little about computers, they are often frustrated and baffled in their attempts to understand what is happening. It is important that designers learn these more general facts about their audience.

Studying attitude: Participants' comments will reveal a lot about how they *feel* about the system they are using. This information is very important for products whose use is discretionary. For example, an office system may offer a new way keeping track of appointments. Since people already have ways of doing this, they will not use the new method if they don't *like* it.

Using small samples: The thinking-aloud approach can be used with a very small number of participants and still give valuable results. Because problems can often be pinpointed, data from even one person may give useful feedback to the designer. By contrast, performance data from one person are unlikely to provide any specific feedback at all. Of course, where one is attempting to generalize about a population, as in benchmarking, an appropriate sample size is needed. But many defects can be spotted by studying even a handful of participants.

Using mockups: Since the thinking-aloud method doesn't focus on speed or task success, useful information can be gathered by observing how participants work with pieces of a proposed design. For example, the interpretation of menus and screens can be studied with paper representations or screens mocked-up without a running system. Of course, some important aspects of the system won't be represented in the mockup, but the timeliness of the results makes this kind of work worthwhile.

METHOD NOTES

Here are some notes about the method as we have used it in our laboratory.

Participants. As in any behavioral testing participants must be representative of the intended users in knowledge and experience. The method will work only with real users, not as a walk-through by experienced people trying to spot the problems real users will have. It is impossible for an informed person to pretend to be uninformed in evaluating an interface.

Instructions. The basic instructions can be very simple: "Tell me what you are thinking about as you work." People can respond easily to this, especially if the experimenter gives a few categories of thoughts as examples: things they find confusing, decisions they are making, and the like.

We make a few further points in the instructions we give participants. We tell them we are not interested in their secret thoughts, but only what they are thinking about the task under study. We make clear that we have no stake in the system they are using, so they need not be afraid of hurting our feelings if they make a criticism. We stress that it is the system that we are evaluating, and not they themselves. This is extremely important, since we have found participants tend to blame themselves for any problems they encounter. This creates bad morale, and can interfere with the study as well as being unpleasant for the participant.

The observer. It is very easy to influence by questions or comments the data participants will give you. The observer must try to remain an observer, and not an interviewer. That is, don't ask questions about what *you* want to know about. Let the participants tell you what *they* are thinking. It is necessary to prod most participants to keep talking, especially at first. We try to restrict ourselves to *prompts* like "What are you thinking?" or "What is that telling you?", if the participant is reading. We try to avoid *probes* like "Why did you do that?" or "What do you think 'parameter' means?" If we have specific questions we save them for the end of the session so that they cannot affect the participant's ideas during the study.

The observer gives help when it is really needed. The control of assistance is crucial in any evaluation, because a tiny amount of human insight can clear up very serious problems. In our work we have been interested in the situation in which the customer has no on-site assistance at all, and so we have provided help only when we felt that otherwise the participant would quit. In other studies that rule would be too strict; the point is just that it is necessary to have a clear policy about when help will be given, and what form it will take.

A consequence of our limited help policy is that we must explain to participants that although we very much want them to ask questions, we won't answer. This seems odd at first but becomes natural after a bit.

We watch the participant's morale carefully, and remind him or her not to take the blame for problems that may occur. We have found that this is essential to avoid placing unreasonable stress on the participant.

Recording. We use a simple video-recording system pointed at the screen that captures what is happening there. We use a lapel mike close to the participant's throat, but participants usually talk loudly enough that an ordinary table mike would be adequate. We also use a separate audio recorder, since it is easier to transcribe comments from an audio cassette, using a standard transcription player, than from a video cassette.

We do not have a way of capturing keystrokes on the systems we have studied so far, so the observer must sometimes call out what keys the participant has used so that the information goes on to the soundtrack. In the same way, the observer notes what section of the manual the participant is using, if it is not clear from the participant's comments.

We make careful notes during the session of anything of interest. This is very hard to do early in a study, but after one knows what to watch for it gets easy to keep up. These notes are a great time-saver in analysis, since getting information off the video tapes is slow. We record the time of each note, and use a computer program to keep track of the starting and stopping of the video tape. We can then easily work out where in the tape an episode referred to in a note should be found.

Analysis. Here is a sketch of the procedure we have used in analysing the data from our text-processing studies. The approach may be useful in other domains as well, though the particular categories of problems that are found will be different.

We go through our notes and collect episodes in which users are having trouble or registering complaints. We make a listing of each such episode, coded by participant, and keyed to the original notes for checking. We then try to relate each episode to any aspect of the system or manual to which it may refer. Of course, some difficulties may not be traceable from the comments, but most are. By grouping the episodes, we now can collect all those that seem to involve a given aspect of the design. For example, we collect problems of terminology, cursor control, menu flow, restarting training exercises, and many others. Episodes in these groups are then examined to determine what separate problems may be occurring in each area, and, if desired, what proportion of participants encountered a given problem.

We also attempt to determine *why* a given problem is occurring. Sometimes this is obvious, as with vocabulary problems. Other times it is more subtle, as when typewriter experience interferes with newly-learned text operations. Sometimes no reason can be identified, or no single reason, but very often the participants' comments, as well as the form of errors, will suggest something.

Reporting results. In passing results back to designers, if you are not yourself the designer, the watchword is *specific.* While a summary of types of errors may be interesting what the designers need is specific suggestions for changes. What words weren't understood? Which message was misleading? Which screen led people astray? We have found that if we can provide this information the designers may then want to hear about more general concerns we may have about the design.

Another note: designers often cannot believe that users have the troubles that they do, or they feel that these troubles are in some way the fault of the user. We remind them that they have a great deal of specialized knowledge that helps them to understand systems, and that many users do not. We also remind them that they must adapt to the limitations and preferences of the user, not the other way around: the customer is always right!

WHAT DO YOU NEED TO GET STARTED?

These notes describe the approach we've taken, but the method is flexible. The common-sense goal to keep in mind is to create a situation in which realistic users can tell you about their difficulties as they work. Be careful about instructions, and about your role as observer. Beware of becoming an interviewer, who directs the participant's comments. But don't worry about equipment: useful data can be collected with no other equipment than a note pad, and the system can exist only as a paper-and-pencil mockup. You'll still get a new feeling for your design, and for your audience.

ACKNOWLEDGEMENTS

This material was developed in collaboration with Robert Mack, Jack Carroll, and Karen Greer. John Gould and Jack Carroll made helpful comments on the manuscript.

REFERENCES

Ericsson, K.A. and Simon, H.A. Verbal reports as data. *Psychological Review,* 1980, *3,* 215-251.

Newell, A. and Simon, H.A. *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall, 1972.