Improving a human-computer dialogue

Authors: Rolf Molich and Jakob Nielsen
Date: Mar. 1990
From: Communications of the ACM(Vol. 33, Issue 3)
Publisher: Association for Computing Machinery, Inc.
Document Type: Article
Length: 2,665 words

Abstract:
A survey of professional computer software designers uses an informal contest to analyze the common problems in user interfaces. For an application program to be user-friendly it should satisfy nine conditions. These include simple language familiar to users, repetition of necessary information so users do not have to remember it, consistency of instructions and keystrokes, frequent feedback and clear error messages, shortcuts for more experienced users and clear methods of backing out from every point. Many of the contestants emphasize the problems expected by the researchers but on average, the participants found only about 37 percent of the problems in the exercise and only 44 percent of the 'serious' problems. This survey indicates the need for clear, accessible guidelines for development of user interfaces.

Full Text:
Improving a Human-Computer Dialogue Any system designed for people to use should be easy to learn and remember, effective, and pleasant to use. Over the years there has been a considerable increase in designing interfaces that score highly on these issues. This experience has been documented in a number of guidelines for constructing good human-computer interfaces [5, 10]. Following these guidelines is commonly considered a necessary but insufficient condition for constructing good human-computer interfaces.

Most often, following such guidelines during the design phase imposes little extra effort on a development project. Guideline reports, however, are often lengthy. Documents of more than 400 pages are not uncommon. The mere size of a guideline report often means that it is not consulted during design or design review simply because the work of locating relevant guidelines is not considered worth the effort.

This article describes a survey that we undertook to investigate whether industrial data processing professionals would be able to recognize serious interface problems in simple but realistic dialogues. Seventy-seven designers and programmers from industry and academia participated. Fifty-one were from industry, 10 were teachers or students from universities or high schools, and 16 had occupations that were not specified. Many of them were designers and programmers of administrative systems -- the people who design, write, and maintain our daily programs.

This article consists of four parts. We first present the survey and a number of conclusions from it. The second part of the article presents the exercise used in the survey -- a dialogue that we asked the participants to evaluate as expressed in Appendix 1. The third part contains our annotated solution as shown in Appendix 2 and a suggestion for an improved design as characterized in Appendix 3.

FACTS ABOUT THE SURVEY

The Exercise

We constructed an exercise in evaluating a simple human-computer dialogue. In order to test the reader's understanding of basic features of good interface design, we designed the dialogue for simple display terminals which are still common in many administrative data processing systems: a display of 24 lines of 80 characters each and a keyboard; no color, no mouse, and no graphics.

The Danish edition of Computerworld magazine published the exercise as an informal contest under the heading "The Unofficial Danish Championship in Dialogue Evaluation [6]." To stimulate interest in the contest, a sponsor offered $700 in U.S. currency worth of software for the best entry. The text of the exercise appears in an English translation in Appendix 1.

The functional specification has been constructed solely for the purpose of the Computerworld contest and does not reflect any specific existing system. On the other hand, each of the usability problems in the design can be observed in many systems in the real world.

The Participants

Seventy-seven entries were submitted with suggestions for improving the human-computer interface of the exercise. Based on the professional appearance of many of the submitted entries, we estimate that most of the participants used between two and five hours to complete their entries. Several participants noted that they had found the exercise worthwhile and rewarding in itself. These two facts lead us to conclude that the participants were highly motivated, and therefore the results should be better than those produced by standard designers and programmers.

PROBLEM CLASSIFICATION

We classified the usability problems in the dialogue in accordance with a short checklist of usability considerations in a good dialogue. This checklist reflects our personal experience. The principles correspond to similar principles described by others [1]. Almost all usability problems fit well into one of the categories.

Simple and Natural Dialogue

Dialogues should not contain irrelevant or rarely needed information. Every extraneous unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. All information should appear in a natural and logical order.

Speak the User's Language

The dialogue should be expressed clearly in words, phrases, and concepts familiar to the user rather than in system-oriented terms.

Minimize the User's Memory Load

The user's short-term memory is limited. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. Complicated instructions should be simplified.

Be Consistent

Users should not have to wonder whether different words, situations, or actions mean the same thing. A particular system action -- when appropriate -- should always be achievable by one particular user action. Consistency also means coordination between subsystems and between major independent systems with common user populations [7].

Provide Feedback

The system should always keep the user informed about what is going on by providing him or her with appropriate feedback within reasonable time.

Provide Clearly Marked Exits

A system should never capture users in situations that have no visible escape. Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue.

Provide Shortcuts

The features that make a system easy to learn -- such as verbous dialogues and few entry fields on each display -- are often cumbersome to the experienced user. Clever shortcuts -- unseen by the novice user -- may often be included in a system such that the system caters to both inexperienced and experienced users.

Provide Good Error Messages

Good error messages are defensive, precise, and constructive [9]. Defensive error messages blame the problem on system deficiencies and never criticize the user. Precise error messages provide the user with exact information about the cause of the problem. Constructive error messages provide meaningful suggestions to the user about what to do next.

Error Prevention

Even better than good error messages is a careful design that prevents a problem from occurring in the first place.

EVALUATION PROCEDURE

All entries were intially evaluated by one person. The 13 best entries were subsequently reevaluated by two other judges. All three judges then jointly selected the winner. There were only minor differences between the results of the initial evaluation and the reevaluations.

Grading was very liberal. We gave credit for even the simplest item that related to one of our problems. In many cases, a point was awarded for a correct reformulation of a message even if the general principle (for instance, keep the user informed by providing appropriate feedback within reasonable time) did not appear. An example: Problem 18 concerns the lack of feedback during 30-second database searches (problem numbers refer to the detailed solution in Appendix 2). Here, we awarded a full point for the suggestion, Inform the user that it may take as long as 30 seconds before the reply appears, while no point was awarded for the statement A response time of 30 seconds is simply unacceptable, because the statement does not indicate why the response time is unacceptable or what could be done to alleviate the problem.

COMMENTS ON OUR SOLUTION

Our solution was constructed by carefully applying the nine principles in the usability checklist presented earlier in this article. The submitted entries caused us to revise our original solution. We had overlooked two problems: problem 14 ("Questions must be expressed from the user's point of view") and problem 17 ("Coordinate placement of error messages with the rest of the system"). Problem 27 (" 'Try again' is meaningless") was expressed more precisely by a number of participants.

It is possible that our solution includes some bad points or that we have overlooked some problems. The MANTEL system has not been subjected to empirical tests to indicate how real users would use it.

Problem 20 ("There may be no emergency exit from the initial prompt") and problem 22 ("It may not be possible to edit input in the initial prompt") have a somewhat special character since many of the possible tools for implementing the Telephone Index system would automatically offer the user these facilities. Since some tools do not provide such facilities, however, we need to have this requirement stated explicitly in the system specification.

Comments from the Participants

After our solution to the exercise was published, we spoke to several people who wondered if we had overlooked their solutions. These people had compared their solution with our published solution and felt that they had discovered more than 18 problems (the number of problems that the winner detected). In each case, we were able to convince the participant that our assessment of their solution was reasonable.

An example: One of the solutions stated: ILLEGAL NUMBER -- Nonsense, of course, and also unfriendly. It should say "The number cannot be correct," but it would be better to indicate what is wrong. Even more important: the input field can be constructed in such a way that the error will almost never occur. For this observation we gave credit for problem 23 ("The word ILLEGAL may intimidate the user") and problem 24 ("The error messages are to vague"), but the author also expected credit for problem 29 ("Accept other common forms of telephone number as input") and problem 31 ("Show an example of a telephone number in the initial prompt").

We think that this indicates that the problems appear insultingly simple when you read our solution but that many of them are hard to express precisely. We have little doubt that before the survey several of the participants overestimated their abilities in the human factors area. There is a marked difference between actual and alleged knowledge of the elements of user friendly dialogues. The strength of our survey is that it demonstrates actual knowledge.

WHAT SYSTEM DESIGNERS AND

PROGRAMMERS ACTUALLY KNOW

The results of the survey are summarized in Table I and Figure 1. The average number of problems mentioned was 11.2 out of 30 problems (37 percent). The winner mentioned 18 of 30 problems (60 percent). Our expectations were somewhat higher, when one considers the nature of this study. Presumably, the only solutions submitted were those which the authors felt were good enough to stand some chance of winning.

Some of the problems may prevent some users from using the system in a meaningful way. These problems are marked "Serious" in both the table and in the solution in Appendix 2. The average number of serious problems mentioned was 3.5 out of 8 serious problems (44 percent).

Three problems score notably higher than the rest (problems 9, 10, and 15). Over the last 15 years, there have been several campaigns to make Danish designers and programmers aware of the importance of using Danish instead of English terms in computer output [8]. The high score for problems 9 ("Use Danish terms") and 10 ("Use Danish characters") indicates that the campaigns may have been successful. It also indicates that such campaigns may actually influence people.

Many participants did not understand the meaning of PORT073 and MANTEL INFO RELEASE 4.2. The prevailing attitude of many respondents was that since they did not understand it, they suggested that this information should be removed. Only a few stated that they had to know the exact meaning of this information before deciding on whether it should be reformulated or removed.

Many entries indicated that the respondent considered a questionable feature in the original design good design practice. Several entries contained rephrasings of the message "ILLEGAL NUMBER. TRY AGAIN!" that were more precise in pointing out the problem but which still contained the questionable phrases "Illegal" and "Try again!"

In our opinion, several of the suggestions for improving the interface hardly improved the interface. A few entries correctly noted the spelling error in "subscriper" but suggested that it be changed to another misspelling, such as "supscriber." Other entries suggested barring the user from entering incorrect telephone numbers by rejecting characters that were not digits using a beep as an error message. A beep, however, is not a good error message. It is vague; it does not tell the user what to do next, and it is not expressed in the language of the user.

CONCLUSIONS

Gould and Lewis [4] have succeeded in expressing the basic requirements for the design process in three short principles: early focus on users and tasks; empirical measurement; and iterative design. Gould et al. have demonstrated the applicability and usefulness of these principles in their design and development of the 1984 Olympic Message System [3]. As indicated by some of the questionable suggestions for improvements that resulted from our survey, some designers may have difficulties in applying Gould and Lewis' principle of iterative design appropriately unless they also have similarly simple basic requirements for the design product. Our survey demonstrates the need for expressing and propagating simple and intellectually manageable requirements for the design product. These requirements could be similar to the nine principles we used to construct our solution.

A good dialogue is error-tolerant and provides carefully phrased informative messages in situations where the user may need help. Most specific interface problems can be either avoided or their consequences can be minimized by suitable design of a system. The problem categories covering this aspect of interface design are "Provide good error messages" and "Prevent errors." Except for problem 24 ("The error messages are too vague"), none of the problems in these categories were mentioned in more than 42 percent of the entries. Fifty-five percent did not mention any of the problems in the category "Prevent errors." Only 8 percent suggested that the system should accept other common forms of telephone numbers as input (problem 29); in our opinion, this is the most important problem in the category "Prevent errors." Regrettably, it is our conclusion that many designers and programmers are not sufficiently aware of the importance of designing dialogues in a way that would either prevent or tolerate errors.

A recent study of intelligent help systems [2] concluded that ". . . [The authors] are less confident that the state of the art in user interfaces is clean enough to provide the kind of testbed we wanted." Our study seems to support this point. We have demonstrated that industrial designers and programmers have considerable difficulty in recognizing potential problems in the review of a simple human-computer dialogue.

What can we do to to solve this problem? The first and most difficult step is to realize that we are indeed facing a serious problem. Human-computer dialogue construction appears deceptively simple, yet it is full of subtle pitfalls as we have demonstrated. Second, some intellectually manageable set of dialogue principles should be proposed and its usability demonstrated, in a similar way to Gould and Lewis' three principles for the design process. Third, designers should be made aware of the necessity for extensive review of human-computer interfaces. As our own experience with the MANTEL system shows, the more people that look at the interface, the more problems are detected.

Computer systems are hard for most people to learn and use today. We believe that if human-computer dialogues were designed by people who understand and apply basic dialogue principles, they would achieve much higher usability marks. The results of our survey indicate that many of these principles are neither common knowledge nor intuitive.

Acknowledgments. The authors would like to thank Peter Carstensen, Jan Clausen, Anker Helms Jorgensen, and Bodil Schroder for valuable comments on earlier versions of this article.

Source Citation (MLA 9th Edition)
Molich, Rolf, and Jakob Nielsen. "Improving a human-computer dialogue." *Communications of the ACM*, vol. 33, no. 3, Mar. 1990, pp. 338+. *Gale Academic OneFile*, link.gale.com/apps/doc/A8490466/AONE?u=ull_ttda&sid=summon&xid=76f54b82. Accessed 7 July 2022.
Gale Document Number: GALE|A8490466