Human Error, Reliability, Resilience, and Performance, Vol. 33, 2022, 115–119
https://doi.org/10.54941/ahfe1001575

Design of Conversation Interaction Between Users and Systems to Prevent Human Misunderstanding

Toru Nakata

National Institute of Advanced Industrial Science and Technology, Tokyo, 135-0064, Japan

ABSTRACT

Many human errors and inefficiencies in operation are evoked by bad designs of conversational interaction between users and systems. There are typical problematic conversation styles that commonly exist in many user interfaces and manual documents. This paper discusses about three major types of them with proposing improvement ideas for them. 1) The first one is use of denial logical statement. Humans are not good at comprehending meaning with denial logics. 2) When a system guides a human worker to check something, it should make the worker focus on evidence and not on his memories. 3) Entangled logic structure should be eliminated in conversation. Each question toward users should be independent to other questions, so that the system can align a series of questions without inefficiencies.

Keywords: User interface, Human computer interaction, Human error, Cognitive science

INTRODUCTION

Some designers create human interfaces and manual documents under their assumption that human users basically think in logical way. They think that people succeed to understand messages from the systems when the interactions are conducted completely 'logical'. Nevertheless, people misunderstand even if message from the system is logical. There are many examples of human errors that users failed to understand during conversation with machines. Conversation design is one of important aspect for human error prevention.

Cognitive science researchers consider human ability of thinking as quite imperfect. Our logical ability is intuitive but not logically strict (Kahneman, 2011). Also, capacity of our memory is limited, so that we cannot think about many things together. Aristotle also said humans tend to focus only on imperfect parts of the problem. We thus fail to make decisions correctly in many cases.

Systems interacting with human users should take care of guide the users to avoid their logical weakness. We should not limit such "systems" only in user interfaces of computer systems. Even paper manual documents should be designed likewise electronical interactive systems from the viewpoint of interaction design. When a system accepts commands from a user, it asks some questions to the user several times to clarify user's demand. The style

of the conversation is quite important, so it should be designed with considerations on questioning style, order of questions, concreteness of question, and so forth.

This paper proposes three principles on questioning design as follows:

- Principle of positive wording and avoiding negative expression.
- Principle of independency from user's memory
- Principle of elimination of logical entanglement

PRINCIPLE OF POSITIVE CONDITIONING

We often see questions and guidance with logical negative form: for example, *"If you aren't a student, go to the office."* The negative description may be liked by some designers when the converted statement into positive style is much longer than denial form.

However, we should avoid the negative conditioning as much as we can. Humans are not good at handling of negative logic. Users cannot clearly imagine the meaning of question when they are asked *"If not"* first. On the other hand, positive and explicit description of *"If you are a professor, go to the office"* is very straightforward and produces correct understandings.

Moreover, double negative and triple negative sometimes appears in the conversations. In many case, double denials (such as *"If you aren't a student, don't go to the classroom"*) can be converted a simple positive logic like *"If you are a student, go to the classroom."*

We, however, must be careful about 'the law of excluded middle'. *"Not disliking"* is not inevitably equivalent to *"liking."* In some case, true meaning of a double denial statement is too ambiguous to assure explicit and positive meaning. The meaning of the example above might be intended as *"If you are a student, there is no instruction for the time being."*

Most people cannot convince that the meaning of *"Nobody does not eat nothing"* is equivalent to *"Anybody eats nothing."* Surprisingly, even modern machine translation programs fail to comprehend the meaning of triple denial sentences. Off course it exceeds normal human ability.

Multiple denial seems ridiculous and hard to handle, but there are many of them in the user interfaces. Some computers ask us to press the 'cancel' button to cancel the cancel command that we are ready to execute.

Also, multiple negative forms appear when the system refer to other systems or document. As shown in Table 1, some instructions force the users to wait: it refers conditions that are described in ahead, so the readers cannot find correct answer then immediately.

PRINCIPLE OF INDEPENDENCY FROM USER'S MEMORY

The question should be answered by a user without remembering things in the past. One of typical bad questions is to inquire the experience memory of user's actions. Consider an example of 1) in Table 2. This instruction triggers only when the operator is remembering that own action is completed. Memory is unstable, and actions are not physical objects.

Table 1. Bad design pattern of use of negative logic.

Bad Design		Improvement Idea	
Negative statement	*"If it isn't A, do X."*	Explicit and positive instruction	*"If it is B or C, do Y."*
Multiple Denial	*"If it isn't A, don't X."*	Simplification	*"If it is A, do X."*
Withholding of judgment	*"If it matches with the condition below, do Y. - It is B."*	Elimination of referring with delay	*"If it is B, do Y."*
Combination of bad features	*"If it doesn't match with the condition below, don't Y. - It isn't B."*	Simplification	*"If it is B, do Y."*

Table 2. Bad design pattern of requiring users' memory too much.

Bad Design		Improvement Idea	
1) Check relying on memory of action	*"If you have put salt and sugar in, your operation is completed."*	Check on objective status	*"Put salt and sugar in. Then, taste the soup."*
2) Parallel conditions	*"If it satisfies A, B and C, do X."*	Step-by-step judgement with fixed order.	*"1) If it satisfies A, read the next line. 2) If it satisfies B, read the next line. 3) If it satisfies C, do X."*

Remembering action experiences are more difficult for users than examining objects in front of them. The user's answer might be incorrect.

Problematic checks depending on action memory are widely observed in industries. In some companies, operation manuals require workers to make 'double check' of a unskillful way: the manuals guide by saying a statement like *"Press the button. Write the check mark in the box if you have pressed the button."* In this case, the verification is carried out immediately after the operation. The worker who checks will be the same person of the operation, so that there are few objectivities for verification.

To increase reliability of check, we must design the procedure as follows:

- The checker should be different from the worker who made the operation.
- Timing of the operation and the check should be separated in order to refresh worker's recognition.
- Check must focus on evidence and not on worker's memory Checking on present objects requires no memory about the past, so there are few possibilities that some misunderstandings get in.

Table 3. An example of entangled option design.

	Card Brand A	*Card Brand B*
With milage program	Possible	Not Available
Without milage program	Possible	Possible

Also, a question with many parallel conditionings requires user's memory effort. Users must memorize long preconditions until they conclude a judgement.

Omission is one of the most frequent types of human errors. In the study of workflow management, it is considered that the cause of omission is parallel preconditioning. When the manual guides the user with parallel preconditions connected with 'and,' for example *"If you have done A and B, then do X,"* the user may forget *A or B*. Connection of preconditions with 'and' (conjunction) is called 'and-join' and it is regarded as troublesome logical structure for workflow control (Van der Aalst et al., 2004).

Such long preconditions should be decomposed into a series of short conditionings as shown in Table 2.

PRINCIPLE OF ELIMINATION OF LOGICAL ENTANGLEMENT

A conversation should go forward smoothly without stepping back, but sometimes it happens due to logical entanglement among the questions.

Suppose a situation as follows. The first question is *"Which do you select as a credit brand A or B?"* Then, only for users who chose brand *A*, the following question is given: *"Brand A has an option of courteous milage program of X airline. Do you want it?"* If brand *B* is chosen at the first question, the user never meets the question talking about the possibility of the *milage* thing. Even if the *milage* option is very attractive, the user loses opportunity to select it. Otherwise, after being informed about the existence of the *milage* option, the user will cancel his choice of brand *B* to step back to the first question. It wastes user's effort and time.

A brute force solution for the entanglement problem is to flatten all questions into a list of choices, which is called 'disjunctive normal form' (DNF). The system can paraphrase the offer as *"There are three choices: 1) brand A with milage program, 2) brand A without milage program, or 3) brand B without milage program."* DNF, however, often becomes too long for practical use because it shows all possibilities at once.

In our society, we use another solution that reorder the questions in respect to importance. If the option of *milage program* is more important than choice of card brand, it should be confirmed earlier than them. We, however, cannot know preference of all users beforehand, so this kind of reformation may not fit for a certain part of users.

The fundamental improvement is elimination of logical dependency. If brand *B* can have the option of *milage program* too, we can ask the two questions independently in any order.

Although the logical dependency makes the conversation design complex and inefficient, companies employ it very often under an attempt to control customers' behavior. When a company wants to sell brand *A* rather than *B*, they attach an attractive option only on brand *A*. For this reason, may conversations of user interface in our society become complex, and some users face difficulty to make their orders.

CONCULSION

The set of consumers spread with very high diversity. Even though some parts of consumers are good at logical comprehension, general people may fail to catch the meaning of machines messages. Human logic ability has characteristics, and the design of user interfaces should be fit for it. This paper pointed out the three major points of them.

As future work, one can consider about automation of those improvement. The improvements presented in this paper can be realized only by designers' hands now. However, those reformations have fixed patterns, so they will be able to be processed automatically. The risks contained in user interfaces and manual documents can be pointed out by language processing ability of recent artificial intelligence (AI).

REFERENCES

Aristotle. (2011), Problems, Harvard University Press.

Kahneman, Daniel. (2011). Thinking, Fast and Slow, Farrar, Straus and Giroux.

Van der Aalst, W., Van Hee, K. (2004). Workflow Management: Models, Methods, and Systems, MIT Press.