Advances in Human Factors/Ergonomics, 19B

Human-Computer Interaction:

Software and Hardware Interfaces

Proceedings of the Fifth International Conference on Human-Computer Interaction, (HCI International '93), Orlando, Florida, August 8–13, 1993
Volume 2

Edited by

Gavriel Salvendy
Purdue University, West Lafayette, IN 47907, USA

and

Michael J. Smith
University of Wisconsin, Madison, WI 53706, USA

ELSEVIER

Amsterdam – London – New York – Tokyo 1993

An Adaptive Intelligent Help System

Chi-Tien Chiu, Chaochang Chiu, and A. F. Norcio

Department of Information Systems, University of Maryland Baltimore County, Baltimore, MD 21228-5398

Abstract

This paper introduces the architecture of an adaptive intelligent help system. The paper also discusses how this system can operate with different users and applications adaptively and intelligently. Novice users certainly need help to learn and to use the target application. Even experienced users need help to make the most effective use of the application. This system also emphasizes the motivation of learnability through help strategies.

1. INTRODUCTION

A help system should assist the user in specific situations. There are two aspects on which an intelligent help system should focus in order to provide effective support to the user. These two aspects are *who the user is* and *when the help should be activated.* The first requirement for the system which is its ability to adapt and give advice intelligently, is that the system must know the expertise level of the user. The system adopts the methodology directly from the model which infers an individual user's task-specific expertise dynamically[1]. The second requirement is that the system should be able to monitor a user's activity in order to detect any error or inefficiency. To meet these two criteria, our system includes a Help Strategy Map (HSM) and a Task-domain Knowledge Checker (TKC). Both are discussed below.

An adaptive human-computer interface can potentially fulfill three key roles in support of the user: (1) it helps users accomplish intended goals on the target application; (2) it enables users to understand how the application operates; and (3) it can increase the abilities of the user [2]. In this paper, we suggest that an intelligent help system should dynamically trigger an adequate help strategy for each individual user as the situation requires. By doing so, our system aims to achieve the following goals: 1) problem solving -- helping user out of the current interaction mistake or dilemma; and 2) task learning -- helping the user to gain familiarity with the task. By monitoring the previous five commands, this system generates a help message if the user commits an error or has obvious difficulties. Figure 1 illustrates the architecture and the components of the system.

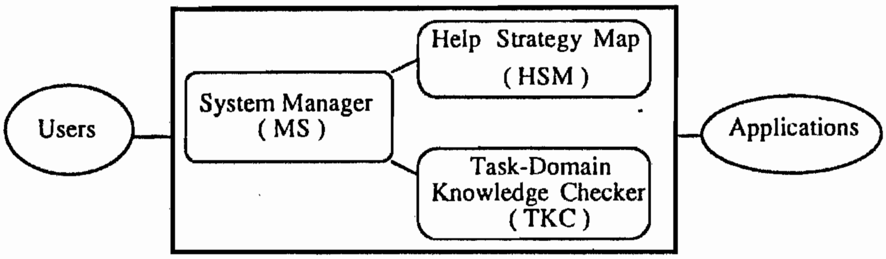

Figure 1. System Architecture

2. THE HELP SYSTEM

There are three basic concepts underlying this help system:

1. Task-orientation: The system emphasizes the representation of more specific, domain-oriented knowledge [3]. The purpose is to acquire sufficient knowledge about a particular problem area. The system is task-dependent so that its knowledge base is a state transition diagram.

2. Explicit user's goals: In order to address the system's generic functionality, the goal of each task is pre-determined for the user. The object is to modify the behavior of the interface to meet the inferred goals of the user. It appears that the mechanism of our system could have the ability to detect not only the error, but also the user's intentions. However, this is not in the scope of our focus.

3. Problem-solving and learnability: This is the main task which our help system is designed to accomplish. Our design focuses on the specific context of help message. This help system does not act like a tutor that provides the so-called best learning strategy. The user has to solve the specific problem by using the target application (in our example, a word processor). The system tries to help the user achieve the appropriate goals as well as gain familiarity with the task.

3. SYSTEM MANAGER (SM):

The System Manager handles the dialogue between the user and the system. It issues an appropriate help message to the user according to the help strategy from the Help Strategy Map (HSM) and the type of help content from the Task-Domain Knowledge Checker (TKC).

4. HELP STRATEGY MAP (HSM):

expertise of the user and monitor the user's performance. Two parameters are directly derived from the previous information using a fuzzy inference engine and neural network processes, namely the User Knowledge Level (UKL) and the Observer Performance Pattern (OPP) [1].

The fuzzy inference engine identifies user's level of expertise for a specific task domain dynamically, and the neural network determines the user's continuous performance pattern. The HSM is a two dimensional table with five help strategies. The strategy in each cell is determined by the user knowledge level (UKL) and observed performance pattern (OPP).

OPP / UKL	VS	LS	US	LU	VM
1	A	A	B	B	C
2	A	B	B	C	D
3	B	B	C	D	D
4	B	C	D	D	E
5	C	D	D	E	E

TABLE 1. Help Strategy Map

where:

OPP: **VS**: Very-likely stable **LS**: Likely stable **US**: Unstable
LU: Likely misunderstood **VM**: Very-likely misunderstood

UKL: **1**: Expert **2**: Near expert
3: Experienced **4**: Near experienced **5**: Novice

According to Fisher, Lemke and Schwab, a good help system must have a passive and an active component [5]. Most on-line help systems are passive. A user must activate the help system by either typing an help command or clicking the help icon. Some help systems track the user's navigation and the possible error the user committed. But those error messages focus only on general types of errors. This means that the user frequently does not know the system tried to help. The five help strategies in Table 1 provide a user with the adequate help messages (passive or active). They include:

Strategy A: Message only. This category occupies the leftmost and highest three positions in HSM. These positions stand for the user's high expertise and stable performance of the task. Hence, the system should let this level of user be active and give the error message only. The system does not provide any help or hint. The user must either activate the on-line help or issue another command. By doing so, the system not only can obtain feedback, but also achieve its goals. This feedback assists the system in deciding whether this user should be reclassified. In another case, if this user was misclassified, we can assume that this user is likely to commit another error and the system must reevaluate the current classification. Thus, the system can offer an alternative help strategy based on the classification.

Strategy B: Message and ask if need help. This category occupies those upper and left positions in HSM. We can see that users in this category might need some assistance. The system provides an error message and optional help. The latter is to ask whether the user needs the system to be active; that is, provides the right step. The user can bypass the help.

This explains how the system can stimulate the user's learnability to enhance understanding of the task.

Strategy C: Message and suggestion for the next right step. This category fills up the diagonal positions in HSM. These users definitely need help in order to maintain their current classification. If a user takes the suggestion, the user remains in the current category. Otherwise, most likely the user is downgraded. At this point, the system acts semi-actively, that is the user still can ignore the suggestion. Therefore, the system helps the user to comprehend the task.

Strategy D: Mandatory next step. This category occupies the lower and right positions in HSM table. A user in this category should do the next step in order to continue the task. From this point on, a user is passive until the system reclassifies the user to a higher classification.

Strategy E: Give information about the task and tell next step. This category consists of the lowest and rightmost three positions in HSM. A user who is classified in this category is unaware of the task. The system provides more information about the task until the user has learned enough to move to another category.

5. TASK-DOMAIN KNOWLEDGE CHECKER (TKC)

The TKC is used to track the user's actions during the task-performing process and to evaluate its correctness and efficiency. This procedure not only can detect if any error occurred and but also can determine proper help messages. Based upon the state transition diagram of the specific task (for example, the text editor in Figure 2), a rule-based system is constructed to store the necessary navigation to accomplish the task. By monitoring every step the user attempts, the TKC is able to determine if the user is correct, wrong, or lost.

5.1 Technique of TKC

As we mentioned before, our example in Figure 2 is only one small portion of a text editor. Even in a complex system, text editing consists of a number of tasks based on their functionality. As long as a task state transition diagram can be formed for each one, there is a way to monitor how the user navigates.

It is generally agreed that an intelligent interface should be able to infer and evaluate a user's plans and intentions. It should also adapt its behavior to the individual user and to the user's current task [6]. Our methodology can also be used in a task-independent environment. By monitoring the user, the system should be able to detect the user's goals. In our system, we assume that user's goals are already known. The system monitors the user's movements and makes comparison to the state transition diagram. Therefore, the system knows where the user is and whether any error has occurred.

5.2 Example

For instance, there is a file called EXAMPLE, the task is to delete the whole second paragraph and then save it. This example is trivial. The TKC forms the most optimal route for navigation based upon the state transition diagram. That is:

If the user didn't type or select "OPEN", the TKC constructs the different message for him according to the user's category in HSM.

Strategy A: On the screen, maybe it says "ERROR! Press RETURN to continue.". After the user enters RETURN, the system returns the user to the previous state.

Strategy B: On the screen, it says "ERROR! "XXXXX" is not correct here. Press RETURN to continue or "ESC" for help.".

Strategy C: On the screen, it says "ERROR! "XXXXX" is not correct at this point. "OPEN" is the right command. Press RETURN to continue.".

Strategy D: On the screen, it says "ERROR! "OPEN is the right command. Please Enter "OPEN" to continue.".

Strategy E: On the screen, it says " To delete one paragraph in a file, you have to "OPEN" the file first, and "BLOCK" the paragraph, "CUT" it, and "SAVE" the file.

The above error message for each strategy is only a possible suggestion.

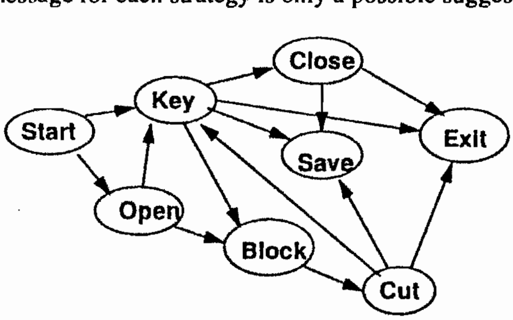

Figure 2. State Transition Diagram Example

6. DISCUSSION

Since the novice-expert continuum exists, an intelligent help system should provide the adaptive mechanism for the individual user. Novice users certainly need help both in learning and use of the system. Even users proficient in specific tasks may have more to learn [7]. That is why our proposed help system issues a help strategy based on not only the classification, but the current performance of the users. The only way to use a computer is by issuing a command. This can be achieved either by typing a command, clicking an icon, selecting from a menu, or even talking to the computer system. According to Norman [8], the interface should be designed to allow the individual user to: (1) execute an action by more directly expressing high-level intentions; and (2) receive direct help in planning how to carry out the task. The help messages in our system are designed either to recommend a correct command or to assist using a correct command.

7. CONC

This
strength
errors an
strategies
observed
supposed
Also, the
1. H
system is
individua
system sh
individual
2. Hc
additional
on error-c

REFERE

1. C. Chiu
Users i
System
2. Jeffery
Tutorin
Erlbaur
3. Shortlif
Elsevie
4. Fowler,
the ada
Winder
Press.
5. G. Fishe
Ergono
P. Gorn
6. Greenbe
viability
7. Tyler, S
context
327, 19
8. Noman,
Eds. Me

7. CONCLUSION and REMARKS

This proposed help system is designed to provide help to different classes of users. One strength of the system is the generic task-specific knowledge checker which can detect errors and inefficiencies. Another strength is the HSM table which generates different help strategies determined not only by the level of a user's knowledge, but also by the user's observed performance. Regardless of the multimode which an intelligent interface is supposed to have, we believe our help system can function adaptively and intelligently. Also, there are several challenging directions for future research:

1. How to integrate the help system to an intelligent user-computer interface. A help system is only one of the functions of the entire interface. The multimode of interaction for individual group of users is necessary for an intelligent interface. By the same token, a help system should be able to determine what type of interaction mode is more suitable for the individual user.

2. How to implement this mechanism in a task-independent environment. There are additional concerns, such as detecting the user's goal, necessary before the TKC can work on error-checking.

REFERENCES

1. C. Chiu, and A. F. Norcio (1992), "A Fuzzy Neural Network Architecture for Modeling Users in Adaptive Human-Computer Interfaces.", Submitted to the IEEE Transactions on Systems, Man, and Cybernetics.
2. Jeffery G. Bonar, "Interface Architectures for Intelligent Tutoring Systems", In Intelligent Tutoring Systems, Ed. Hugh Burns, James W. Parlett, and Carol L. Redfield, Lawrence Erlbaum Associates, Publishers (1991), Hillsdale, NJ
3. Shortliffe, E. H. (1976). Computer-based medical consultations: MYGIN. New York: Elsevier.
4. Fowler,C., Macaulay, L., & Siripoksup, S. (1987). An evaluation of the effectiveness of the adaptive interface module (AIM) in matching dialogues to users. In D. Diaper & R. Winder (Eds.), People and computers III (pp.345-359). New York:Cambridge University Press.
5. G. Fisher, A. Lemke, and T. Schwab, "Active help Systems", In Readings on Cognitive Ergonomics - Mind and Computers, Ed. G. C. van Veer, M. J. Tauber, T.R.G. Green and P. Gorny, Proceedings of the 2nd European Conference Gmunden, Austria, Sep. 1984
6. Greenberg, S., and Witten, I. (1983). "Adaptive personalized interfaces - A question of viability.", Behavior and Information Technology, vol. 4, pp. 31-35.
7. Tyler, S. W. and S. Treu, "An interface architecture to provide adaptive task-specific context for the user", International Journal of Man-Machines Studies, vol.30, pp.303-327, 1989
8. Noman, D. (1983). Some observations on Mental Models. In D. Gentner & A. Stevens, Eds. Mental Models. Hillside, NJ:Lawrence Erlbaum Associates.