A Study on an Intelligent Help System that synchronizes with Operation of Application Software

Ichiro KOBAYASHI[†,‡], Tomoko TSUNAWAKI[¶]

[†]Dept. of Information Sciences, Faculty of Science, Ochanomizu University

[‡]Brain Science Institute, RIKEN

[¶]Hitachi System and Services, Co. Ltd.

E-mail: koba@is.ocha.ac.jp

Abstract-In this paper, as a part of development of a system which supports people who are unfamiliar to use a computer, we have developed an intelligent help system that synchronizes with operation of application software. We rebuilt a word-processing software called JAVA Word Processor as example application software of the system. The implemented system can provide suitable instructions for each user at proper timing, synchronizing with user's operation of the software. The system has the user model obtained from a subject experiment, which decides suitable instruction contents and proper timing to each user.

I. Introduction

The computer has spread into ordinary homes in recent years, and the opportunity for many people to touch a computer has been increasing. Today there is a variety of application software, however, since each of the software has its own operation methods and functions, a user needs big labor to master the operation methods. Therefore, in this paper, as a part of development of a system which supports people who are unfamiliar to use a computer, we have developed an intelligent help system that synchronizes with operation of application software. There are several studies on development of user support system to use application software [1, 2]. Those studies have proposed mechanisms to predict a user's next operation and to provide the user with instructions related to the operation at proper timing by using probabilistic reasoning. On the other hand, our system calculates proper timing and proper instruction contents based on the user model obtained from subject experiments. — We have obtained user's behavior in operating application software and constructed the user model for each user based on the analysis of the user's operation history. We rebuilt word processing software called JAVA Word Processor as an example application software of the system and have developed a word processor which provides proper instructions to a user at proper timing, synchronizing with user's operation of the software.

II. Subject Experiment For User Model

In order to provide proper instructions for each user at proper timing, we have observed user's tendency to operate a word processor by a subject experiment. Based on the obtained data, we have constructed the user model and then personalized it for each user.

A. Experiment settings

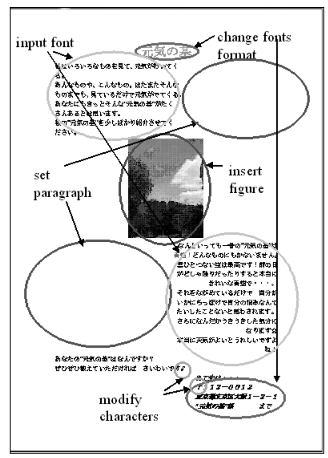

Figure 1: Objective document

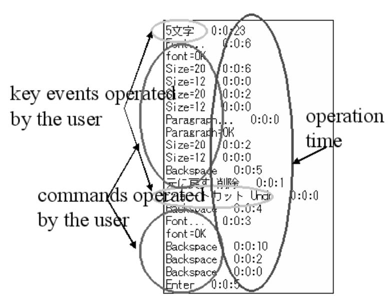

Figure 2: User's operation history

• Characteristics of subjects

We took an experiment using seventeen subjects who are undergraduate and graduate students. Some of them are familiar and others are unfamiliar to using a computer.

• Task

Subjects are requested to make a document by using word processor software which is unfamiliar to the subjects. The software is called JAVA Word-processor which is a sample program of Swing, the graphic component of

JAVA programming language [3], and has basic functions to edit a documenet, i.e., 'editing', 'paragraphing', 'inserting files', etc.
The document shown in Fig. 1 is given to the subjects and asked to make the same document with no reference and then we observed where the subjects need instructions to make the document smoothly.
To make the objective document, the subjects have to use at least the following functions, 'inputting fonts', 'changing font format', 'setting paragraph', 'inserting files', and 'modifying fonts'.

• Obtained data

In order to obtain the user's operation history through the subject experiment, we have rebuilt the JAVA Wordprocessor so as to be able to observe user's operation history from 'key event name', 'command name' user operated, and 'operation time'(see, Fig. 2).

B. Result of analysis

Fig. 3 shows the total history of user's document making process. From this graph, we can see that the user inputs texts at the first stage ((a) in Fig. 3) and s/he modifies each part of the input texts ((b) in Fig. 3), and then s/he repeated the same operation to set paragraph ((c) in Fig. 3). From the history of user's operation, we see the point where users tend to have problems in operation.

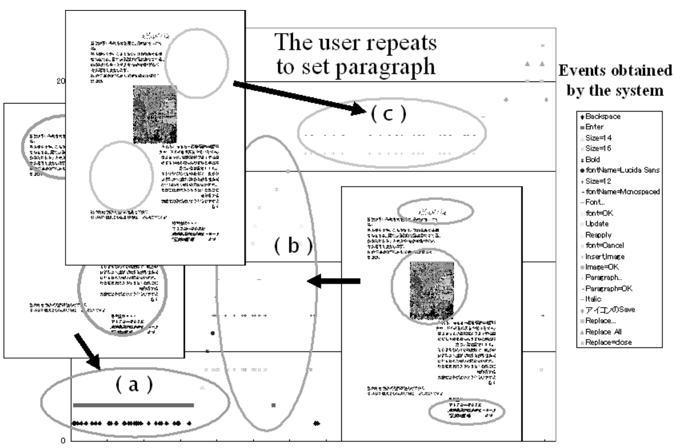

Figure 3: User's document making process

As the result of the analysis of user's operation behavior, we have recognized that the following five factors have to be considered to construct a user model.

1. In case that the user repeats the same operation several times, it is understood that the user is not familiar to that operation. (This is observed by (c) in Fig. 3.)
2. In case that the interval time between operations by the user is long, it is recognized that the user does not understand that operation.
3. In case that the user opens and closes menu disorderly, it is recognized that the user is finding an objective operation item.
4. The more the user is familiar to operation, the more variety of functions of the software the user can use. (This reduces the number of operations to perform a task. The correlation between total operation times and total time to make the document is shown in Fig. 4. The correlation coefficient is 0.49.)
5. Typing speed is related to the familiarity of user's operation of a computer. (The correlation between typing speed and total time to make the document is shown in Fig. 5. The correlation coefficient is -0.53.)

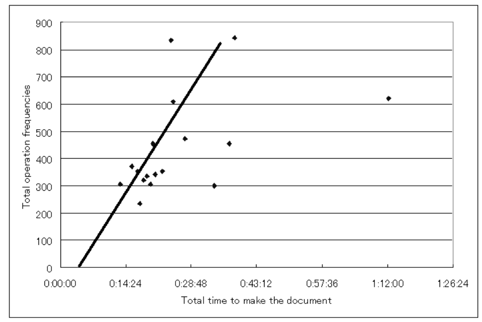

Figure 4: Correlation between total operation times and total time to make the document

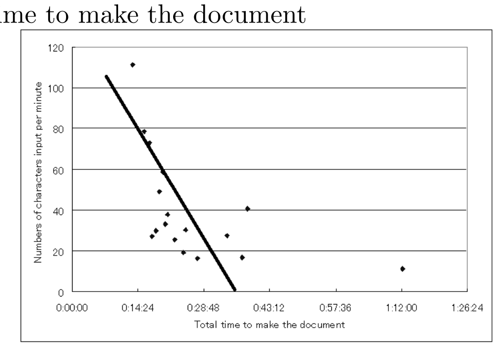

Figure 5: Correlation between typing speed and total time to make the document

C. User model

The user model is constructed as the one which provides proper instructions at proper timing, considering user's operation process of the application software. Since both the timing to provide instructions and the contents of instructions are differed by each operation item, each instruction is set by each operation item. Moreover, how a user can contact to the item depends on the constitution of menu and tool bars, and also since how serious the user intends to operate is observed by the behavior of user's operation of tool bar and menu bar, therefore, considering this, we have set the instruction contents for each operation item.

The user model is constructed based on the user's operation categorized with the five types reflecting the constitution of tool bar and menu bar as shown in Fig. 6.

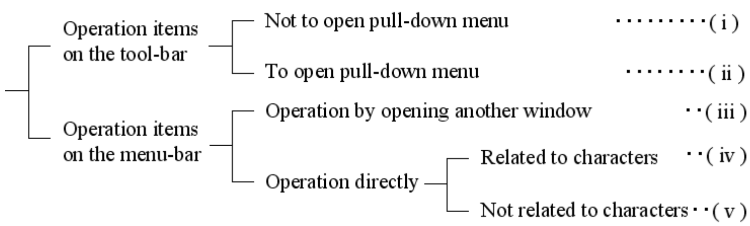

Figure 6: Five types of user's operation

As for the operation on tool-bar, there are two types of operation; (i) one is not to open the pull-down menu (see, Fig. 7) and (ii) the other is to open the pull-down menu (see, Fig. 8). Compared (i) with (ii), we can see that the user is more serious to operate in case (ii) rather that in case (i) because s/he opens the menu purposely. By this fact, different instruction contents are provided to those cases.

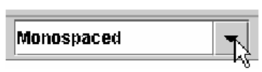

Figure 7: Not to open pull down menu

Figure 8: To open pull-down menu

As for the operation on menu-bar, there are three types of operations; the first one (iii) is to operate through another window (see Fig. 9), the second one (iv) is operation directly related to characters (see Fig. 10) and the third one (v) is operation not directly related to characters — The reason why the direct operation is divided into two categories, i.e., (iv) and (v) is that the way of providing instructions is differed by the states of operation on characters. (The details will be explained later.)

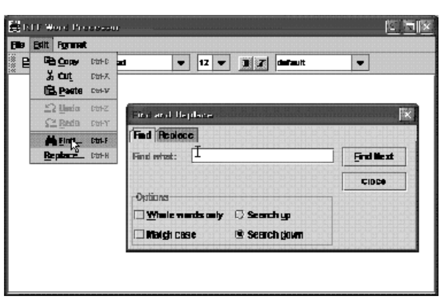

Figure 9: Operation by opening another window

Figure 10: Operation directly related to characters

As an example of these five types of the user model based on the constitution of menu and tool bars, we show the relation between instruction contents and timing in Table 1, which shows the case of the operation (ii) in Fig. 6.

Table 1: Relation between instruction contents and timing

previous operation / operation times	no previous operation	has input characters	has selected characters
Few	OO BO		I-D-RO
	short		immediately
Many	I-ID-RO	OO I-D-RO	OO I-D-RO
	short	short	immediately

'Previous operation' is the operation preformed beforehand the current operation. Instructions given to the user will be differed by whether or not there is the previous operation related to the current operation. In case of the operation items, 'Font' and 'Size', the states of the previous operation are categorized as the three cases (a) 'no previous operation', (b) 'has input characters', and (c) 'has selected characters'. Operation times is the number the user has ever operated the operation item. If this number is few, it can be recognized that the user is not familiar to this operation, and vice versa.

The contents of the instructions are divided into four categories: 'operation overview'($\boxed{\text{OO}}$ in Table 1), 'basic operation'($\boxed{\text{BO}}$ in Table 1), 'information directly related to the operation' ($\boxed{\text{I-D-RO}}$ in Table 1), and 'information indirectly related to the operation' ($\boxed{\text{I-ID-RO}}$ in Table 1). 'operation overview' is the most elemental instruction normally provided to beginners. 'information indirectly related to the operation' is the most advanced instruction normally provided to experts.

III. Personalization Of A User Model

We have established five types of user model in terms of operation items. However, the concrete values of the models have not yet been determined. They have to be determined by considering user's skill level. We introduce an equation to decide the concrete value of operation times, 'few', 'many' and grace time, 'short', 'long' for each user. Since we see that there is relation between user's operation skill level and typing speed from the correlation shown in Fig. 5, the equation is defined with the variable of typing speed.

In order to decide the concrete values for 'few' and 'many' of operation times for each user, we used the data obtained from the experiment. At first, we obtained the average of typing speed from the top five data.—The average is 74.04 characters input per minute. And then from the worst five data, we obtained 17.85 characters input per minute as the average. Here, we defined that the concrete value of 'few' of operation times for experts is one times, and five times for beginner. By using these values, the following equation is obtained.

$$f(x) = -0.07 \times x + 6.27$$

$f(x)$	:	The value to distinguish between 'few' and 'many'
x	:	The number of characters a user can type per minute

From this equation, the concrete values of 'few' and 'many' are able to be decided based on user's operation skill level, i.e., typing speed in this case.

As well as operation times, the grace time is also decided by the same method. We obtained the average of necessary time to perform one operation from the top five data. The average is 6.23 seconds. And then from the worst five data, we obtained 12.71 seconds as the average of the data. By using these values and the values of the typing speed, that is, 74.04 characters per min. for experts and 17.85 characters per min. for beginners. The following equation is obtained.

$$g(x) = -0.115 \times x + 14.78$$

$g(x)$	:	The average of necessary operation time to finish the task
x	:	The number of characters the user can type per minute

By this equation, we have set the concrete value of grace time before the system provides instructions, 'immediately', 'short', and 'long' as follows:

immediately	:	0 seconds at any user's level
short	:	$-0.115 \times x + 14.78$ seconds
long	:	$-0.115 \times x + 14.78 \times 2$ seconds

Here, we regard 'short' as the average time to finish a task. Therefore, 'short' for beginner is longer than 'short' for experts. In other words, in case that 'short' for experts is obtained as 6 seconds from the equation, experts are expected to finish the task within 6 seconds, and only when s/he needs more time than 6 seconds, instructions for the task will be provided by being judged that s/he has problems in operation. On the other hand , in case of beginner, 6 seconds is considered too short to finish the task. Therefore, 'short' for beginner is set as longer than that for experts. Fig. 11 shows this mechanism. (In Fig. 11, 'short' for beginner is regarded as 12 seconds.)

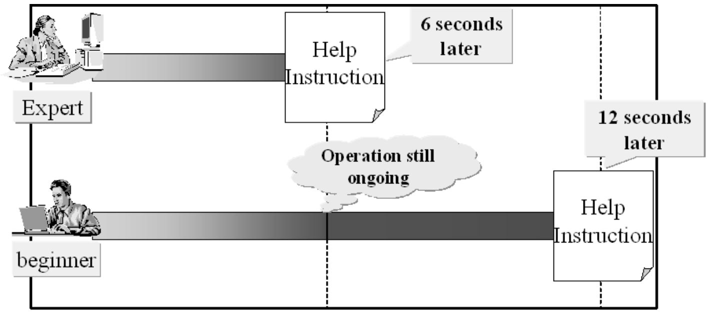

Figure 11: Grace time to provide instructions for each user

'long' is set twice as long as 'short' so that 'long' is regarded by the user as enough time to finish the task. Therefore, if the user cannot finish the task within 'long' period, it is recognized that the user suffers from the operation. And if the user finishes the task within that period and moves onto the next task, instruction of the task will not be given to the user so as to avoid redundancy in providing unnecessary instructions.

IV. Operation Examples

We show concrete examples when a beginner and an expert use the developed system.

Case A: the user who can usually type 28 characters per min.

Operation times is calculated as $f(28) = -0.07 \times 28 + 6.27 = 4.31$ times, therefore, by 4 operation times, it is regarded as 'few' operation times and if it exceeds 5 operation times, it is regarded as 'many' operation times. As well as operation times, operation average time, 'short', is calculated as $g(28) = -0.115 \times 28 + 14.78 = 11.56$ seconds, 'long' is 23.12 seconds, therefore, in case that no previous operation have ever done before and user's operation times are less than 4 times, 'operation overview' is 'immediately (i.e., 0 seconds later)' provided to the user ((a) shown by the dotted line in Fig.12). By this, the user will know what operation s/he can do by using this software.

In case that there is a previous operation have done before and user's operation times are less than 4 times, operation overview is provided to the user 12 seconds later since the user has started the operation ((b) shown by the dotted line in Fig. 12). Actual operation example of this case is shown in Fig. 13

previous operation / operation times	(a) no previous operation	(b) has input characters	has selected characters
Few (less than 4 times)	OO	BO	OO I-D-RO
	0 seconds	12 seconds	0 seconds
Many (more than 5 times)	I-ID-RO	OO I-D-RO	OO I-D-RO
	12 seconds	12 seconds	0 seconds

Figure 12: Instruction contents and timing for Case A

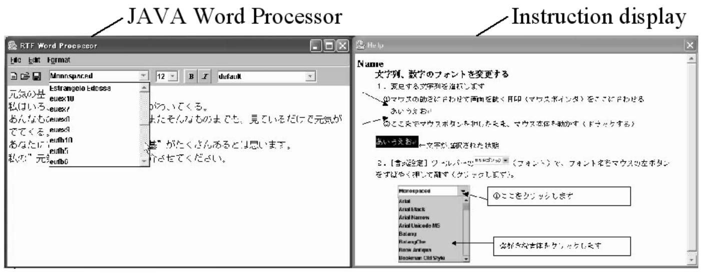

Figure 13: Instruction about basic operation is given 12 seconds later

Case B: the user who can usually type 80 characters per min.

Operation times is calculated as $f(80) = -0.07 \times 80 + 6.27 = 0.67$, therefore, by 0 operation times, it is regarded as 'few' operation times and if it exceeds 1 operation times, it is regarded as 'many' operation times. As well as operation times, operation average time, 'short', is calculated as $g(80) = -0.115 \times 80 + 14.78 = 5.58$ seconds. Therefore, 'long' will be 11.16 seconds. In this case, the user model is obtained as shown in Fig. 14. Actual operation example of (c) shown by the dotted line in Fig. 14 is shown in Fig. 15.

previous operation / operation times	no previous operation	(c) has input characters	has selected characters
Few (less than 0 times)	OO	BO	I-D-RO
	6 seconds	6 seconds	0 seconds
Many (more than 1 times)	I-ID-RO	OO I-D-RO	OO I-D-RO
	6 seconds	6 seconds	0 seconds

Figure 14: Instruction contents and timing for Case B

Instruction contents

As we see the instruction contents shown in Instruction display in Fig. 13 and Fig. 15 respectively, depending on the user's level, i.e., user's typing speed, the instruction contents are differed even if the same instruction (e.g., 'basic operation ($\boxed{\text{BO}}$) in this case) is provided to the user. By this mechanism, we realize to provide users with suitable instructions at proper timing.

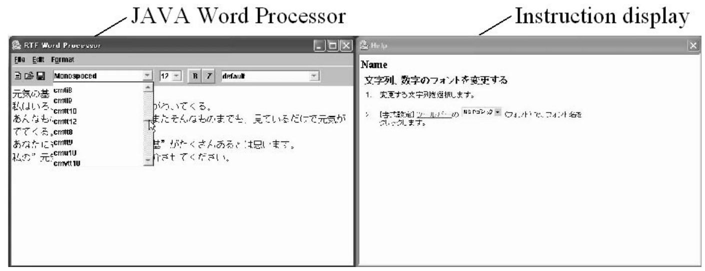

Figure 15: Instruction about basic operation is given 6 seconds later

V. Conclusions

In this paper, as a part of development of a system which supports people who are unfamiliar to use a computer, we have developed an intelligent help system that synchronizes with operation of application software. In the system, we have proposed a mechanism that provides proper instructions to a user at proper timing. This mechanism works based on the user model which is obtained through a subject experiment. At this point we have proposed another approach to construct a user model from the approach to construct it based on probabilistic reasoning, and have verified our approach is also useful to construct a user model. As future works, we attempt to have more subject experiments to obtain more precise data for personalizing the user model. Moreover, in order to provide more tailor-made instructions to users, we attempt to compile instruction contents in the form of XML so that instruction contents are more flexibly changed to user's operation skill level.

References

[1] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K.: The Lu-miere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users, Proc. of the 14th Conference on Uncertainty in Artificial Intelligence (1998) 256-265.

[2] M.Virvou and B.D.Boulay: Human Plausible Reasoning for Intelligent Help ,User Modeling and User-Adapted Interaction,Vol.9,(1999) 321-375 .

[3] SWING, http://www.javafaq.nu/java/free-swing-book/

[4] Yoji Kiyota,Sadao Kurohashi and Fuyuko Kido: "Dialog Navigator ":A Question Answering System based on Large Text Knowledge Base, In Proceedings of 19th COLING (COLING02), (2002) 460–466.

[5] Hideto Kazawa,Hideki Isozaki and Eisaku Maeda: NTT Question Answering System in TREC 2001

[6] Kristina Hook, "Steps to take before Intelligent User Interfaces become real" ,Journal of Interaction with Computers Vol.12,No.4,(2000),409-426.