A mirror-effect-based mutual tutorial - a tutorial system for different interface of a single information service -

Keiko Katsuragawa†‡, Takeshi Oono†, Minoru Tomikashi†, Satoru Kogure*, Toshihiko Itoh**, Tatsuhiro Konishi*, and Yukihiro Itoh‡

†Nissan Motor Co., Ltd.
‡Graduate School of Science and Technology, Shizuoka University, Japan
* Faculty of Informatics, Shizuoka University, Japan
** Graduate School of Information Science and Technology, Hokkaido University, Japan
kei-k@mail.nissan.co.jp

Abstract - The purpose of this paper is to propose a mirror-effect-based mutual tutorial on different kind of interfaces for a single information service. When a user operates a system via a certain interface, the mirror-effect-based tutorial helps him/her learn the operating procedures of other interfaces of the same system. When conducting a certain task, the system uses the same program and input data, irrespective of which interface is used by a user. Taking advantage of this fact, our tutorial system generates operation procedures for implementing the same function on different interfaces when a user carries out a certain task. Accordingly, our tutorial involves the following two features. (1) Presenting the procedure to perform the same task from a different interface when a user operates a system via a certain interface. (2) Providing the efficient procedure depending on each interface. The task completion time is reduced by 24% and the input acceptance rate is increased by 17%.

Keywords: mirror-effect-based, mutual tutorial

1 INTRODUCTION

There are many different information services readily available in daily life today. The types of information terminals used and the environments in which users access these services are continually diversifying. In line with these trends, the interfaces for accessing information services are also becoming truly diverse.

For example, the Shinkansen ticket reservation system [1] provides two kinds of interfaces for cell phones and PCs. Many car navigation systems also have a built-in microphone for inputting voice commands while driving, in addition to ordinary control switches and a remote controller. Users can change input devices in accordance with match their circumstances.

In addition, since car navigation systems are now connectable to the Internet, Web sites, which were previously accessed mainly from PCs or mobile phones, can now also be accessed from equipment built in vehicles [2]. As a result, such information systems must now be designed to be accessible from a wide variety of terminals.

Although there have been a lot of studies on the usability of interfaces, most of those studies examine an interface accessed from a single type of device [3][4], and there have been only a few studies[5] which examine an interface accessed from multiple types of device.

Figure 1 shows a typical example of a system accessible from multiple interfaces. In this paper, we use the term "device" to refer to such input equipment as a mouse or a microphone that is used with a PC, a car navigation system, etc. We further user the term "interaction style" in the sense of Shneiderman [6]. He cites menu selection, form fill-in, command language, natural language and direct manipulation as five typical interaction styles in interactive software. Finally, we use the term "interface" to refer to an input/output mechanism that combines the devices and the interaction styles provided by terminals.

For example, one interface combines a PC keyboard and command language, and another interface combines switches on a car navigation system and menu selection. There is also an interface that combines speech recognition through a microphone and menu selection.

Although many systems provide multiple interfaces for the purpose of selective use of multiple interfaces according to the user's circumstances, there are virtually no examples of an interface design policy that is aimed at easy shift from one interface to another. In many cases, each interface is designed separately. Consequently, even when users operate a familiar system, they must newly learn the operating procedure if they use unfamiliar interfaces. In such a case, they must once again go through a process of repeated trial and error, just as they did the first time they operated the system. This means that we cannot improve the system's usability by simply increasing the variety of terminals or interfaces capable of accessing a system and merely improving the individual operating ease of each device/interface.

Selective use of multiple interfaces impose a heavy burden on users, and it cannot be neglected especially when systems take advantage of a ubiquitous environment, in which people selectively use various types of information terminals according to their circumstances. For this reason, there are growing demands for interfaces such that users can smoothly shift from one to another depending on their circumstances.

For in-vehicle equipment in particular, users have substantially different circumstances depending on whether they are driving or not. Unless a separate interface tailored to each of these circumstances is provided, the usability may decline

ISSN1883-4566/09

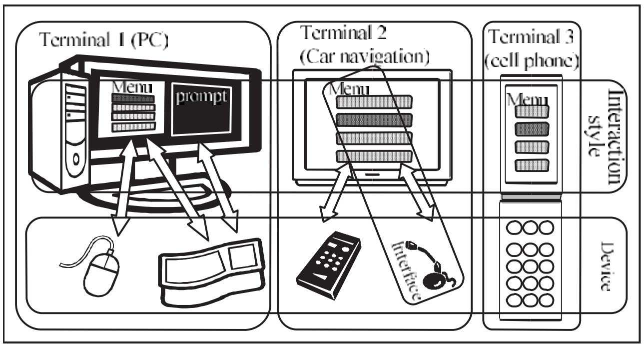


Figure 1: Example of multiple interfaces.

markedly. In addition, for systems that are intended to be accessed both from in-vehicle equipment and from outside a vehicle (e.g. from a home PC via the Internet), it is necessary to provide an interface that is easy to use from each type of device although the circumstances are completely different from each other. Consequently, systems accessible from in-vehicle equipment in particular inevitably require some mechanism that allows users' selective use and smooth shift of their interfaces.

In this regard, this paper proposes a tutorial that helps users learn the operating procedures of unfamiliar interfaces during conducting a task through a familiar interface. When users are carrying out a certain task through a familiar interface, our system demonstrates the procedures for the same task through another interface. We call this learning support system a mirror-effect-based mutual tutorial system. By adopting this tutorial system, we can construct an operating environment for information system that fully takes advantages of multiple interfaces.

This paper provides an evaluation of the mutual tutorial function. The function is implemented onto the drive planning (DP) system constructed in our previous studies [7]. The DP system is accessible from two kinds of interfaces. One interface is designed for the natural language interaction style with a PC keyboard, and the other is designed for the menu selection interaction style with a car navigation system's switches on its instrumental panel. The usability of our system incorporating the tutorial method to be proposed is evaluated on the basis of the acceptance rate of the user input and task accomplishment time.

2 MIRROR-EFFECT-BASED MUTUAL TUTORIAL SYSTEM

We propose a mirror-effect-based mutual tutorial system as a method of improving the usability of a system with multiple interfaces. This section explains the method in detail.

2.1 Definition and Key Principles

We define the term "mirror-effect-based mutual tutorial" as follows; for every task a user actually executes via one interface, the system demonstrates how to perform the same task efficiently on another interface. In the mirror-effect-based tutorial, when a user executes a task using a certain interface, the efficient counterpart operation on another interface is automatically presented to the user as if it were a mirror image of the original operation. In this way, users can unconsciously learn how to use unfamiliar interfaces without active learning of the unfamiliar interfaces.

Figure 2 shows the basic design of the mirror-effect-based mutual tutorial system, which generates an efficient procedure on an interface which corresponds to the task executed on another interface.

The role of an interface at the time of a user input is to interpret the input, identify the necessary program and generate input data into the program. To accomplish it, each interface is equipped with an interpreter that interprets the user's input. In Figure 2, the interpreter A and the interpreter B interpret the inputs from the device A and the device B, respectively.

Generally, identification of the program set to be executed and generation of the input data based on the user's input operation are conducted uniquely. In cases where the input from each interface is aimed at executing the same task, the respective inputs are converted into the same input data and trigger the same program set inside the system. Consequently, if the system has an inverse interpreter for each interface that generates the targeted operation on the interface based on the program set to be executed and the input data for each program, the input to a certain interface feeds the system so that the system generates the corresponding operations for other interfaces. In Figure 2, the interpreter A^{-1} is the inverse interpreter that generates input operations for the interface A, and the interpreter B^{-1} is the inverse interpreter that generates input operations for the interface B.

When implementing an inverse interpreter, the following three possibilities must be considered:
(1) There is no input operation corresponding to the input data and the program to be executed.
(2) There is only one input operation corresponding to the input data and the program to be executed.
(3) There is more than one input operation corresponding to the input data and the program to be executed.

In the case (1), the task to be executed on the basis of the identified program and input data cannot be performed from the interface associated with the inverse interpreter. In this case, it is sufficient to indicate to the user that the task cannot be executed via that interface.

In the case (2), the input operation is the only output from the inverse interpreter.

In the case (3), the procedure which is expected to require the shortest operating time is selected among the operating procedures that can be generated. This enables the user to learn the most efficient operating procedure. If there is no difference in the expected operating time of the generated operating procedures, a predetermined typical procedure is selected. (From the viewpoint of efficiency in this particular sense, inputting a sentence via a natural language interface is not different from inputting another sentence in which a synonymous expression is used.)

How to design inverse interpreters depends on the deference of input flexibility among interfaces. However, the number of programs to be executed is finite, and the finiteness

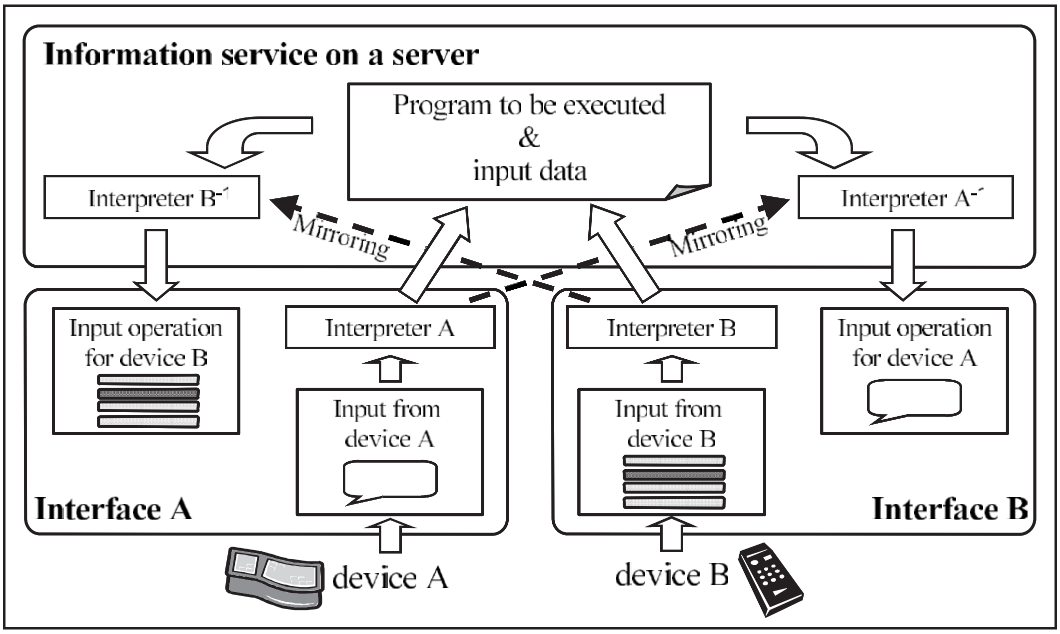


Figure 2: Basic design of mirror-effect-based mutual tutorial system.

makes it possible to define appropriate inputs for each interface to run a given program (e.g. sentence styles for natural language interface, the kind of input information and its order for menu selection interface, etc.). In addition, if the number of parameters for a given program is finite and if the specification of the parameters is definable (e.g. substituting a suitable word into an input sentence with a designated sentence style, selecting an item from a menu list, etc.), we can design an inverse interpreter for any interpreter.

2.2 Features and Expected Effects

Based on the key principles explained in the previous subsection, we can realize a mutual tutorial with the following two features.

a. When a user executes a task via a certain interface, the system teaches the procedure for the same task to be performed on a different interface.

When a user carries out a certain task, the user employs a particular strategy. In facility search, for example, there are at least two strategies: (1) specifying the genre and the landmark near the target facility and (2) specifying the genre and the address of the target facility. A user who frequently uses a particular type of strategy from one interface is expected to employ the same strategy from another interface. The mirror-effect-based mutual tutorial system demonstrates the operating procedure via another interface which corresponds to the user's strategy, only when the user actually carries out a task from one interface. This allows a user to learn operating procedures efficiently by limiting them to the functions that the person actually needs. In addition, since the tutorial is given when the user executes a task on a familiar interface, we can expect that the user learns how to use another interface while using the interface. In other words, we can expect that the user does not have to spend time to especially learn operations on unfamiliar interfaces.

b. The system teaches the most effective procedure according to each interface.

Note that operating procedures executable from different interfaces may differ in their efficiency. Some operating procedures are efficient if they are performed on a certain interface, but they are not if executed via another interface; i.e., other procedures are more efficient in the latter case. Further, there are cases where an efficient operating procedure executable from one interface is simply unavailable on another interface. These facts make it more complicated for users to learn how to operate each interface efficiently according to the task to be performed.

Taking the above considerations into account, the system teaches the most efficient operating procedure on another interface for executing the same task as the one the user actually performs. In other words, the system does not teach the operating procedure involving the same operating steps as the user's original operation if the procedure is not efficient. This feature can help users notice that the most efficient operating procedure on one interface is not necessarily the most efficient procedure on another interface and vice versa. In Figure 2, the selection of the most efficient procedure is conducted when the inverse interpreters generate operating procedures. When an inverse interpreter can generate more than one procedure, the most efficient procedure is selected and thus the system can teach the interface-specific efficient procedure.

3 MUTUAL TUTORIAL IN THE DRIVE PLANNING SYSTEM

This section describes design examples of inverse interpreters for a menu selection interface and for a natural language interface. We also explain examples of the mirror-effect-based mutual tutorial employing these inverse interpreters.

3.1 Drive Planning System

In this study, we have selected our drive planning (DP) system [7] as an example of a system in which the same functions can be used from multiple interfaces. The mutual tutorial function has been implemented in the DP system.

The DP system allows drive plan data created on a PC to be uploaded to an online server; the system supports a variety of helpful functions including route guidance, facility search and plan editing via a car navigation system or a mobile phone. We have selected the facility search function as a target of implementing the mirror-effect-based mutual tutorial function.

When users access to the DP system from a PC, the system offers a natural language interface with keyboard input in order to take full advantage of a standard input device of a PC (i.e., keyboard) and the user- friendliness of natural language input. The natural language interface is aimed at enabling users to input their requests in whatever style they like. For example, a user can simultaneously input a complex combination of search conditions as in "Please look for a hotel near the national university in Hamamatsu".

The car navigation system interface for the DP system adopts a menu selection style manipulated by control switches on the instrumental panel. The interface also allows users to search for a facility by specifying relatively complex conditions (e.g., by designating the type of target facility and its nearby facility). Nearly all of the various facility search functions that can be used from a PC are also available on the car navigation system interface.

3.2 Generation of an Efficient Operating Procedure for Another Interface in the DP System

The facility search module of the DP system accepts search conditions like the target facility's category, address, and nearby facility. The specified search conditions are expressed in a tree structure and transferred to the search module. Figure 3 shows an example of the tree structure for a "search for a fast food restaurant near a train station in Hamamatsu City, Shizuoka Prefecture." The nodes of the tree are connected by links (solid lines in the figure) expressing upper/lower hierarchies or entire/partial hierarchies between concepts and by links (dashed lines) expressing the modifying/modified relationships between concepts. The tree nodes are expressed as frames that store pairs of an attribute and its value and/or pairs of an attribute and a pointer to its attribute value. For example, the address frame stores a pair of the attribute "prefecture" and its value "Shizuoka", and a pair of the attribute "municipality" and a pointer to its value (the address frame containing

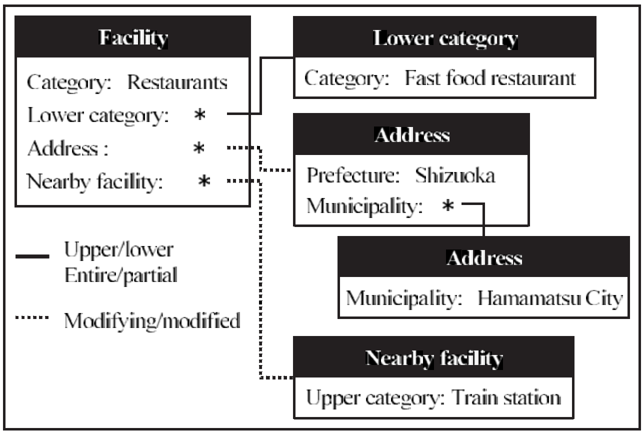


Figure 3: Example tree structure for restaurant search: "search for a fast food restaurant near a train station in Hamamatsu City, Shizuoka Prefecture".

"Hamamatsu City").

Both interfaces use the same search module of the DP system. They transfer the same tree structure to the module if the search conditions are the same. Accordingly, the search module and the tree structure can be regarded as the source for the inverse generation of operating procedure on an interface. This is because the search module determines the program to be executed and the tree structure gives input data to the program irrespective of which interface is used.

We do not discuss the details of the conversion method from natural language input and menu selection input to the tree structure in the DP system since the discussion falls outside the scope of this paper. Conversion from natural language input to the tree structure is explained in [7]. The conversion from menu selection input to the tree structure involves defining the correspondence between each menu item and each attribute of the node in the tree, and transmitting the values selected from the menu to the corresponding attribute values in the tree structure. The following subsections explain the procedures for the inverse conversion from the tree structure to the operating procedure for each interface.

3.2.1 Inverse Generation of Natural Language Input from the Tree Structure

In generating an input natural language sentence from the corresponding tree structure, the system first extracts surface expressions corresponding to the attribute values in each frame. Facility categories and addresses are hierarchized according to their upper-lower or whole-part relation. An expression corresponding to an upper/whole concept is connected with an expression corresponding to its lower/part concept through "no" (the Japanese counterpart to English "of"). If the meaning of a lower/part expression is unambiguous without the restriction imposed by its upper/whole concept, the upper/whole expression is omitted and only the lower/part expression is used so as to generate a non-redundant natural language expression.

Expressions generated from modifying concepts are then connected with the modified expressions through connecting words like "no (of)", "de (in)", "no-chikaku-no (near)", etc. These processes generate a natural language expression from the corresponding tree structure. In order to deal with the cases in which a certain concept is associated with multiple connecting words and is restricted by multiple modifying concepts, their linear order is predetermined according to the attributes in a given frame. The system can use different connecting words depending on whether a modifying expression immediately preceding the modified nominal, or a modifying expression is separated from the modified nominal by another intervening modifying expression.

Suppose that the tree structure in Figure 3 is generated as a result of menu selection operation. In this case, "restaurants" and "fast food restaurants" have an upper-lower relation and "Shizuoka" and "Hamamatsu City" have a whole-part relation. Since "fast food restaurants" and "Hamamatsu City" are unambiguous without their upper/whole concepts, the lower/part concepts are used in generation of a natural language expression. The tree structure has two modifying links to "address" and "nearby facility". Their relative order is predetermined so that the address precedes the nearby facility. The connecting words used in this configuration are predetermined so that "de" is used for the address and "no-chikaku-no" is used for the nearby facility. As a result, "Hamamatsu-shi-de eki-no-chikaku-no famiresu (a fast food restaurant in Hamamatsu City near a station)" is generated as an efficient input for a natural language interface.

3.2.2 Inverse Generation of Menu Manipulation Procedure from the Tree Structure

The menu selection interface is designed to put search conditions into the tree structure for facility search; that is, each choice on the menu is uniquely associated with a particular position in the tree structure. Consequently, by taking advantage of this unique relation between the menu selection and the tree structure, the system can determine which item should be selected on which menu. The system therefore can generate menu selection procedure from the corresponding tree by checking whether each attribute has its value and by determining which choice on the menu is corresponding to the attributes that have their values.

Specification of the values of a lower/partial level of hierarchized attributes can be done only if the upper/whole concepts have been specified. For that reason, the upper/whole concepts are specified first. The order for specifying other attribute values is basically arbitrary, but it is more efficient to specify them in the order in which they are displayed on the screen. In cases where the same value can be specified for multiple selection operations, the procedure with the fewest operations is selected.

For example, the system generates menu selection procedure from the tree structure in Figure 3 as follows. Since "Shizuoka" and "Hamamatsu City" have a whole-part relation, "Shizuoka" is selected on the prefecture menu before the selection of "Hamamatsu City" on the city menu. Since "restaurants" and "fast food restaurants" have an upper-lower relation, "restaurants" is first selected on the genre menu. Then the restaurant category menu becomes available and "fast food restaurants" is selected on the restaurant category menu. The facility category, the address, the nearby facility (station) can be specified in any order. However, since they are displayed on the screen in the order of the facility category, the address, the nearby facility, the system generates the menu selection procedure with this particular order.

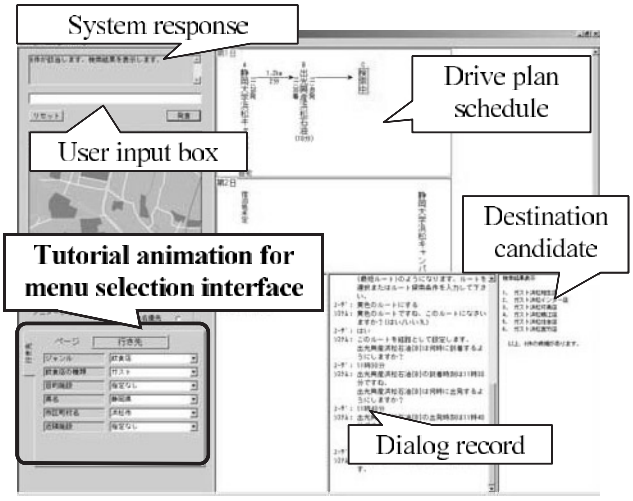


Figure 4: Example tutorial of the menu selection interface during operation on the natural language interface.

3.3 Mutual Tutorial in the DP System

As the tutorial of the menu selection interface while operating on the natural language interface of the PC, we have adopted an explicit method whereby the actual input procedure is shown in an animation. The PC system has a menu display for tutorial that is identical to that of the car navigation system. Figure 4 shows an example of a tutorial of the menu selection interface during operation on the natural language interface. User requests in Japanese are input into "User input box". The response of the system is displayed in "System response". "Dialog record" shows the dialog history. In this example, the user specified a gas station as the first destination, route, departure/arrival time and travel period. Now, the user going to specify the second destination so he inputs "Hamamatsu-shi-no famiresu-ni iki-tai (I want to go to a fast food restaurant in Hamamatsu City)". The menu selection interface on the PC then shows an animation that provides how to conduct facility search with the same search conditions on the menu selection interface. The animation enables the user to directly watch which choices should be selected on which menu in what order.

The tutorial of the natural language interface during operation on the menu selection interface adopts the following method. When a user conducts facility search on the menu selection interface, the DP system gives a response sentence to confirm the search conditions. In generating the response sentence, the system use natural language expressions suitable for the most efficient search on the natural language interface and those expressions are highlighted in the response

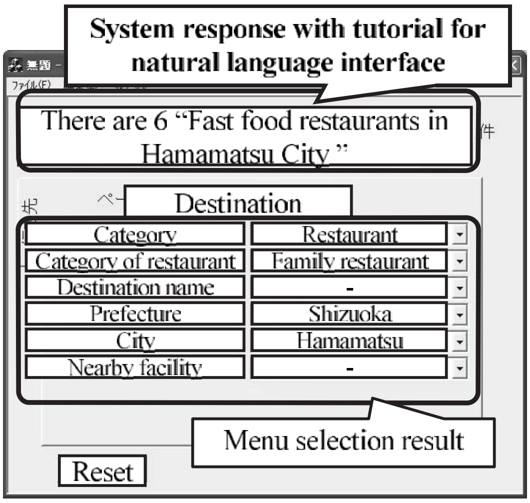


Figure 5: Example tutorial of the natural language interface during operation on the menu selection interface.

sentence. Suppose that a user selects " restaurants", " fast food restaurants", " Shizuoka" and " Hamamtsu City" on the genre menu, the restaurant category menu, the prefecture menu and the city menu, respectively. The system then replies " Hama-matsu-shi desu-ne. **Hamamatsu-shi-no famiresu-**wa 24-ken desu. (You have selected Hamamatsu City. There are 24 **fast food restaurants in Hamamatsu City**.)". In this reply, " Hamamatsu-shi-no famiresu (fast food restaurants in Hamamatsu City)" is highlighted. In repeating such tutorials, users unconsciously acquire natural language expressions suitable for the most efficient search on the natural language interface. Figure 5 shows an example of a tutorial of the natural language interface while operating on the menu selection interface.

When a user uses a function that is unavailable on another interface, the system must show the user that the function is unavailable on another interface. In our DP system, detailed editing of a drive plan such as setting the departure/arrival time can only be done from the PC system, and these operations are not available on the car navigation system's menu selection interface. When a user uses such functions on the PC system, the tutorial system shows their unavailability by not showing an animation of the operations on the menu selection interface.

4 EVALUATION OF THE MUTUAL TUTORIAL SYSTEM IN THE DP SYSTEM

This section describes the evaluation of the effectiveness of the mutual tutorial system.

4.1 Effect of the Tutorial of the Natural Language Interface During the Operation of the Menu Selection Interface

First, a car navigation system without a response-based tutorial function was selected as the system for comparison. The mutual tutorial function was incorporated in another car navigation system, and the effect of the tutorial on the operation of the PC system interface was investigated. We have evaluated the effect of the tutorial of the natural language interface during the operation of the menu selection interface. The operation results of the two user groups have been examined. One group operates the natural language interface after using the car navigation system with the tutorial function, and the other group operates it after using the car navigation system without the tutorial function.

4.1.1 Experimental Procedure

The subjects were twenty engineering students and they were divided into two groups. Group NT first used the car navigation system without tutorial function and then used the DP system on a PC. Group T first used the car navigation system with the tutorial function and then operated the DP system on a PC. The two car navigation systems have no difference except the tutorial function. In order to familiarize the subjects with their respective car navigation system, we made them practice operating the system for 30 minutes every day over a three-day period. The subjects then subjectively evaluated the car navigation system that they used. After that, a prepared conversation of a certain family talking about their desires regarding a family trip was presented to the subjects. All of the subjects then used the same PC system to create a drive plan that satisfied the desires of the family members.

The conditions under which searches could be conducted, and the operations particular to the DP system (e.g., halt a search midway by entering the command "stop search") were explained to the subjects beforehand using the same manual sheet alone.

4.1.2 Experimental Results

After creating the drive plan, the subjects evaluated the PC system subjectively by responding to a questionnaire using a 7-point evaluation scale. An objective evaluation was also made on the basis of the input acceptance rate, the number of input times, the number of key words per sentence and the task accomplishment time.

As the first step, a questionnaire was conducted among the members of both groups concerning their respective car navigation systems. The purpose of the questionnaire was to confirm that there were no differences between the two navigation systems in terms of their performance.

The questionnaire results of the two groups and the results of a t-test between the two groups are shown in Table 1. The mean scores for Question 1 concerning the ease of use (easy to use = 7) are 4.9 for Group NT and 5.3 for Group T. The mean scores for Question 2 concerning the kindness of the responses (kind = 7) are 4.9 for Group NT and 5.2 for Group T. Those for Question 4 about the ease of understanding the screen display (easy to understand = 7) are 5.0 for Group NT and 5.6 for Group T. Although the scores for Group T are slightly higher, the t-test indicates that there is virtually no significant difference between the two groups. The evaluation scores for Question 3 concerning the uncertainty about

Table 1: Questionnaire results for the two car navigation systems.

Questions	NT	T	p-value
1.Ease of use of system	4.9	5.3	0.419
2.Kindness of responses	4.9	5.2	0.624
3.Uncertainty about what to input	3.6	3.4	0.633
4.Ease of understanding of screen display	5.0	5.6	0.382
5.Discomfort from the response sentences	—	4.5	—

Table 2: Questionnaire results for the PC system.

Questions	NT	T	p-value
1.Ease of use of system	3.9	5.4	0.044
2.Kindness of responses	4.1	5.3	0.046
3.Uncertainty about what to input	2.3	3.8	0.002
4.Ease of understanding of screen display	5.0	5.5	0.486
5.Helpfulness of the car navigation system response	—	6.2	—

what to input (no uncertainty = 7) are nearly equal to each other. Thus the questionnaire results indicates that there is no difference between the car navigation systems in terms of the impression the subjects received when using their respective systems.

Table 2 shows the questionnaire results after the subjects made the drive plan on the PC system following the use of their respective car navigation systems. The scores for the questions 1, 2 and 3 show significant difference between the two groups. The mean scores of Group NT and Group T are 3.9 and 5.4 for Question 1, and 4.1 and 5.3 for question 2, respectively, indicating that higher evaluation scores have been given by Group T that used the car navigation system with the tutorial function. Presumably, the effect of the tutorial from the car navigation system makes it clear what sentences should be entered via the natural language interface when searching for a certain facility. If this is the case, it surely improves the overall ease of use of the system, thereby accounting for Group T's higher evaluation scores.

The mean score of Group T for Question 3 concerning the uncertainty about what to input (no uncertainty = 7) is also higher at 3.8 than 2.3 for group NT. The fact that Group T gave a higher evaluation to the PC system also indicates the effect of the tutorial on this group. Group T also gave a high mean score of 6.1 in response to Question 5 about whether the navigation system responses were helpful (helpful = 7). This question was only given to Group T. The high score indicates that the subjects actually felt the effect of the tutorial.

Table 3 shows the results of the objective evaluation. Significant difference is observed between the two groups for all of the evaluation items. Item 1 in Table 3 shows the total time required to create the drive plan. The other evaluation results only pertain to the input process for conducting a facility search and the other functions available on the PC system alone (e.g., a route search function, a function for setting the desired arrival time, etc.) are excluded from the evaluation. This is because the facility search function is the only function that is supported by the tutorial function.

Table 3: Objective evaluation results for the PC system.

Evaluation items	NT	T	p-value
1.Plan creation time(min:sec)	13:23	10:09	0.016
2.Number of input times	24.3	11.2	0.005
3.Number of key words per sentence	1.8	2.6	0.002
4.Acceptance rate of input sentences(%)	72.4	89.7	0.005

The mean time spent by Group T to create the drive plan is 10 minutes 9 seconds, which is more than 30% shorter than that of Group NT (13 minutes 23 seconds). Item 2 is the mean number of data inputs for conducting a facility search. The result for Group T is 11.2 times, which is less than half the number for Group NT (24.3). The results indicate the efficiency of the tutorial. The efficiency is explained in terms of Item 3, where Group T put more key words into an input sentence than Group NT whereby Group T needed fewer sentences than Group NT in making the facility search. Key words here are those words that express search conditions like facility categories, prefectures/cities of the destinations.

Item 4 represents the rate at which the system correctly understood the intention of the user inputs. Group T had a better result at 89.7% compared with 72.4% for Group NT. A good reason for this difference is that Group NT used many facility search conditions that the current system does not accept, whereas Group T almost never used such inappropriate conditions. The system does not accept search conditions that are not stored in the data base (e.g., yasui hoteru (cheap hotel)), conditions that involve the purposes of search alone (e.g., shokuji-suru (have a meal)), and so on. The difference between the two groups can be attributed to the effect of the tutorial from the car navigation system; that is, the tutorial makes it clear what search conditions can be used.

The results indicate that the tutorial greatly helps users unsure about what to input on the natural language interface, by putting efficient natural language inputs for the PC system into the responses from the car navigation system.

4.2 Effect of the Tutorial of the Menu Selection Interface During the Operation of the Natural Language Interface

We also investigated the effect of showing an animation on the PC system screen as a tutorial for the operation of the car navigation system.

4.2.1 Experimental Procedure

The test subjects were twenty engineering students who were divided into two groups, as was done in the experiment described in the preceding section. Group NT first used a PC

Table 4: Questionnaire results for two PC systems.

Questions	NT	T	p-value
1.Ease of use of system	5.4	4.9	0.358
2.Kindness of responses	4.5	5.1	0.399
3.Uncertainty about what to input	4.6	4.1	0.221
4.Ease of understanding of screen display	4.8	5.3	0.540
5.Discomfort from the displays animation	—	3.9	—

Table 5: Questionnaire results for the car navigation system.

Questions	NT	T	p-value
1.Ease of use of system	4.7	5.1	0.587
2.Kindness of responses	5.0	4.6	0.529
3.Uncertainty about what to input	3.3	2.9	0.540
4.Ease of understanding of screen display	5.1	5.1	—
5.Helpfulness of the displayed animation	—	5.9	—

system without any tutorial function before using the car navigation system. Group T first used a PC system incorporating the tutorial function before using the car navigation system. Before the experiment, each group used its respective PC system for 30 minutes every day over a three-day period and then answered a questionnaire. After that, all of the subjects were asked to read the manual of the car navigation system. They were then asked to execute six types of facility search tasks using the same car navigation system. One task, for example, was to " search for a hospital near the present location." The subjects then evaluated the car navigation system subjectively by responding to a questionnaire, and an objective evaluation was made based on their facility search time.

The test subjects were twenty engineering students, and they were divided into two groups as in the experiment in the preceding section. Group NT first used the PC system without the tutorial function before using the car navigation system. Group T first used the PC system with the tutorial function before using the car navigation system. Before the experiment, each group used its respective PC system for 30 minutes every day over a three-day period and then answered a questionnaire. After that, all of the subjects were asked to read the manual of the car navigation system. They were then asked to execute six types of facility search tasks using the same car navigation system. One task, for example, was to search for a hospital near the present location. The subjects then evaluated their respective PC systems subjectively by responding to a questionnaire, and an objective evaluation was made based on their facility search time.

4.2.2 Experimental Results

The subjective evaluation was done by a questionnaire using a seven-point evaluation scale after the subjects had completed the 6 facility search tasks. The objective evaluation was based on the task accomplishment time.

First, a questionnaire was conducted to confirm that there was no difference between the two PC systems in terms of their performance.

The questionnaire results and the results of a t-test between the two groups are shown in Table 4. Group T was asked whether they felt any discomfort from the displayed animation (no discomfort = 7). The mean evaluation score of 3.9 was slightly worse than the median. However, the scores for the other questions were virtually identical. This indicates that the tutorial animation did not influence the subjective evaluation of the PC systems.

Table 5 shows the subjective evaluation results for the car navigation system. No large difference was seen in the questionnaire results between the two groups. However, the mean score of Question 5 for Group T is 5.9. The question asked if the animation displayed on the PC system screen was helpful in operating the car navigation system. This indicates that Group T was actually aware of the effect of the tutorial.

The time needed to complete each facility search task is shown in Table 6. Since Task 1 is very simple to execute, no large difference was seen between the two groups. Task 2 is slightly more complicated because it requires the specification of a reference location. The difference between the two groups tends to increase as the complexity of the task becomes large. Task 3 requires approximately twice as many operations as the tasks 1 and 2 because multiple operations are needed to specify the reference location. A significant difference is seen between the two groups with respect to this task. Task 4 involves nearly the same operating procedure as Task 2. Since this task was performed soon after the subjects experienced Task 2, it is presumed that the effect of being familiar with the procedure resulted in there being no difference between the two groups. Task 5 is even more difficult to execute as it requires specifying the travel time (or distance). A significant difference is also seen between the two groups with respect to this task. Task 6 involves more complicated specification of the reference location than Task 5. A difference is seen in the mean task accomplishment time, although it is not statistically significant.

These results show that the tutorial has the effect of reducing the difficulty of learning the menu selection operating procedure, especially for complicated operations.

5 CONCLUSION

This paper has proposed a mirror-effect-based mutual tutorial system as a solution to the problem of having to separately learn the specific procedure for using a different interface every time a user changes an input device. This problem emerges when users use a system that can be operated from multiple information terminals with different interfaces. Even though a user has learned how to operate such a system from a certain interface, the user may well be unfamiliar to the procedure for operating the system from a different interface, and have to learn the operating procedure anew. Experiments were conducted to evaluate the mutual tutorial function incor-

Table 6: Time required for facility search using the car navigation system.

Destination	NT	T	p-value
1.Gas station in Hamamatsu City	0:47	0:37	0.529
2.Hospital near the present location	0:40	0:30	0.142
3.Hotel near Hamanako Paruparu amusement park	2:08	1:03	0.028
4.Park near the train station	1:10	1:12	0.905
5.Lawson convenience store within 30min. from the present location	1:14	0:47	0.026
6.Police station within 10 km from Hamamatsu Castle	2:13	1:35	0.256

porated in a DP system. The results obtained have shown that the mutual tutorial system is effective in shortening the task accomplishment time and in improving the input acceptance rate, among others.

The effectiveness of the mirror-effect-based mutual tutorial depends on how easily each interface is understood. If an interface is very easy to understand, the users would easily learn how to use it without the mutual tutorial. On the other hand, if an interface is very complicated and hard to understand, the users might not acquire how to use it through the mutual tutorial alone. Although we have confirmed the effectiveness of the mirror-effect-based mutual tutorial in the DP system, we may need further experiments using interfaces with various degrees of complexity (difficulty) in order to clarify the effective range of the proposed mirror-effect-based mutual tutorial. To specify the mirror-effect, we also need a comparison experiment on tutorial during operation of other tutorial systems such as a Microsoft office assistant [8].

The proposed mutual tutorial system adopts a framework that takes advantage of operating the same system; a user may well use the same set of functions irrespective of the interfaces he/she actually uses. By enabling users to efficiently learn operations of preferred functions on multiple interfaces, they are expected to be aware of the merits of using the same system from different interfaces. As a result, the same system is expected to be chosen by users irrespective of terminal types.

As a ubiquitous environment is steadily put in place, it is predicted that there will be increasing opportunities for using the same system from a variety of devices depending on the user's circumstances. These devices will include mobile terminals such as cellular phones and personal digital assistants (PDAs) as well as PCs. In such an environment, there will be an even greater necessity to be able to use multiple interfaces selectively and smoothly. The mirror-effect-based mutual tutorial system proposed in this paper should be effective toward that end.

REFERENCES

[1] Express E-reservations, http://expy.jp/top.php

[2] Nissan CARWINGS, Car Navigation Guide, http://drive.nissan-carwings.com/WEB/

[3] J. Nielsen, Usability Engineering, Academic Press (1994).

[4] D. A. Norman, The Psychology of Everyday Things, Basic Books (1988).

[5] S. Aneja, Y. Lim, Designing for Totality of Mobile and Non-Mobile Interaction: A Case Study, Extended Abstracts CHI '07, pp. 1685–1690 (2007).

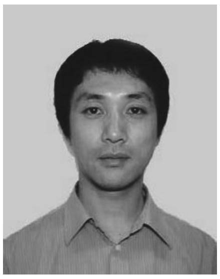
[6] B. Shneiderman, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison-Wesley (1987).

[7] K. Katsuragawa, et al., Construction and Evaluation of a Sub System DPS-PC Which Supports Users in Making and Editing a Drive Plan, Transactions of the Information Processing Society of Japan, Vol. 44, No. 12, pp. 2990–3001 (2003) (in Japanese).

[8] Microsoft Office Online, http://office.microsoft.com/ja-jp/default.aspx

(Received October 15, 2008)
(Revised July 21, 2009)

Keiko Katsuragawa received ph.D from Shizuoka University in 2009. She has been working at Nissan Motor Co., Ltd. Since 2003. Her interests are in man-machine interaction in vehicles. She is a member of Information Processing Society of Japan(IPSJ) and Japanese Society of Automotive Engineers(JSAE).



Takeshi Oono (Member, IPSJ) received the Master's degree from Tsukuba University. He has been working at Nissan Motor Co., Ltd. since 1991. His current research interests is human machine interaction system.

Minoru Tomikashi received his M. in Information Engineering from Tokyo Institute of Technology in 1980. He has been working at Nissan Motor Co., Ltd. since 1980. His current research interests are spoken dialogue system in vehicles and application of drivers' emotion recognition. He is a member of IPSJ, JSAI, IEICE, and the Society of Automotive Engineers of Japan.

Satoru Kogure received Ph.D from Toyohashi University of Technology and became an research associate with Aichi University of Education in 2002. Currently, he is an assistant professor with the Shizuoka University. His interests are in Spoken Dialogue System and so on. He is a member of the Institute of Electronics, Information and Communication Engineers in Japan (IEICE), Acoustical Society of Japan(ASJ), Information Processing Society of Japan(IPSJ), and the Japanese Society for Artificial Intelligence(JSAI).

Toshihiko Itoh received Ph.D from Toyohashi University of Technology and became an research associate with Shizuoka University in 1999. Currently, he is an associate professor with the Hokkaido University. His interests are in Spoken Dialogue System and so on. He is a member of the Institute of Electronics, Information and Communication Engineers in Japan (IEICE), Acoustical Society of Japan(ASJ), Information Processing Society of Japan(IPSJ), and the Japanese Society for Artificial Intelligence(JSAI).

Tatsuhiro Konishi received Ph.D from Waseda University and became an research associate with Shizuoka University in 1992. Currently, he is an associate professor with the Shizuoka University. His interests are in Intelligent Educational Systems and so on. He is a member of the Institute of Electronics, Information and Communication Engineers(IEICE), Japanese Society for Information and Systems in Education(JSiSE), Japanese Society for Artificial Intelligence, the Japanese Cognitive Science Society, and Japan Society for Educational Technology.

Yukihiro Itoh received Ph.D from Waseda University in 1986. He was a associate professor with the Shizuoka University from 1990 to 1999. Currently he is a professor with the Faculty of Informatics, Shizuoka University. His current interests are in natural language processing, Intelligent Educationl Systems and so on. He is a member of Japanese Society for Information and Systems in Education(JSiSE), Japanese Society for Artificial Intelligence, the Institute of Electronics, Information and Communication Engineers(IEICE), The association for Natural Language Processing(NLP), Japanese Cognitive Science Society, Information Processing Society of Japan(IPSJ).