Characterizing User Habituation in Interactive Voice Interface – Experience Study on Home Network System

Noriyuki MATSUBARA, Shinsuke MATSUMOTO, Masahide NAKAMURA
Graduate School of System Informatics, Kobe University
1-1, Rokkodai-cho, Nada, Kobe, Hyogo 657-8501, Japan
matsubara@ws.cs.kobe-u.ac.jp, {shinsuke, masa-n}@cs.kobe-u.ac.jp

ABSTRACT

In this paper, we try to empirically characterize user's habituation effect of the voice control in the Home Network System (HNS). We propose three kinds of metrics that capture the user's habituation quantitatively: (M1) the time of system speech, (M2) the number of support commands and (M3) the number of mistakes. The experimental results show that the metrics M1 and M2 are reasonable to capture the habituation of the user.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Voice I/O*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Training, help, and documentation*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Evaluation/methodology*

General Terms

Experimentation, Human Factors, Measurement

Keywords

Home Network System, remote controller, voice interface, feedback voice, habituation effect

1. INTRODUCTION

The Home Network System (HNS) has been studied extensively as the next-generation ubiquitous application. The HNS connects various home appliances and sensors to the network, and provides comfortable services [3]. One of the important research topics in the HNS is the user interface (UI). The conventional remote controllers and the integrated control panels have limitations in operating a number of heterogeneous appliances and services. The *voice interface* is a promising UI for the HNS, which is being focused recently. Several methods to operate home appliances via voice have been proposed. Basically, the voice interface for the HNS can be implemented by associating certain words spoken by a user (i.e., *voice command*) to an API of the HNS that executes an appliance operation. The voice interface can be adapted easily to various configurations of the HNS. This is because the behavior of the voice interface is determined by the association of a voice command and an API.

In our previous work [2], we have developed the voice interface for our practical HNS, called CS27-HNS. The method employs *mixed-initiative interaction* [1]. The proposed method constructs voice commands through the interaction with a user. So it is unnecessary for the user to memorize a lot of voice commands.

However, for those who got used to the voice interface, it is not always comfortable to use it. Every time a user operates an appliance, the system speaks the feedback. The feedback occurs whenever the user selects an appliance, an operation, and parameters. As a result, the interaction may annoy the user, because the system spends several tens of seconds to finish an appliance operation. To achieve high usability for the user who get used to the interface, we have to introduce a method that detects *user habituation* and optimizes the system feedback.

In this paper, we empirically evaluate the habituation effects of the voice interface in the HNS. We propose three kinds of metrics that capture the user's habituation quantitatively: (M1) the time of system speech, (M2) the number of support commands and (M3) the number of mistakes. In the experiment, we instructed subjects to repeat using the proposed voice interface in the CS27-HNS [4]. We then observe the change of time to spend to interact with the system. Through the analysis of the habituation to the system, we reveal how the habituation occurs and can be detected.

2. INTERACTIVE VOICE INTERFACE FOR HOME NETWORK SYSTEM

2.1 Previous Work

In our previous work [2], we have developed a voice interface with mixed-initiative interaction for the HNS. The advantage of the mixed-initiative interaction is that the users don't need to have prior knowledge of the voice commands. The voice commands are built dynamically through interactions between the user and the system. In the proposed

iiWAS2011, 5-7 December, 2011, Ho Chi Minh City, Vietnam.

system, the users, even if they have never used the voice interface, just speak an appliance name, an operation name and a parameter name, according to the *feedback voice* from the system. The following sequence shows a typical workflow of the proposed voice interface.

1. A user activates the voice control system and registers his name.
2. The user speaks an appliance name which he wants to operate.
3. The system feeds back the list of operation names of the appliance by speech.
4. The user speaks an operation name.
5. If the operation has parameters, the system speaks the list of parameter names.
6. The user speaks a parameter name.
7. The system executes the appliance operation.
8. The user repeats the interaction from Step 2. Otherwise, the user exits the voice control system.

To achieve high usability, the proposed system also allow *asynchronous interaction*, where the system can accept a user speech even while the system is speaking the feedback.

2.2 Problem

The advantage of the proposed interface is that the users do not need to learn a lot of voice commands. However, too much system feedback is not always comfortable for those who got used to the voice interface. Thus, a major limitation of our current implementation is that the system does not count the *habituation* of the user. The voice interaction occurs in every step of selections of an appliance and an operation (and parameters if any). Although the proposed interface adopts the asynchronous interaction, the system requires at least the two steps to execute every appliance operation. Especially, as for the daily routine operations, the habituated user must be bored with the voice interaction itself.

To achieve high usability for the habituated users, we need to introduce a method that detects user habituation automatically, and changes the system behaviors.

3. PROPOSED METHOD

3.1 Key Idea

To cope with the problem in Section 2.2, we aim to detect the habituation effect in the appliance operations by the voice control system in this paper. As a user repeats the appliance operations, the user learns and memorizes the feedback. Eventually, he gets used to appliance operations and he no longer needs the careful feedback.

Our key idea is to capture such habituation effects by some quantitative metrics that can be measured in the voice control system. If the system can detect the habituation of the user, the system can omit or reduce the wasteful long feedback, and optimizes the interaction overhead. We consider that such optimization is quite important to reduce the stress of the users. In this paper, we propose the following three metrics to characterize the habituation effect.

M1: Time of system speech

M2: Number of support commands

M3: Number of mistakes

We explain the details of these metrics in the next sections.

3.2 M1 : Time of System Speech

According to our observation, habituated users tend to respond the feedback *quickly*, and speak the next command at the *early timing*. Conversely, the user who hasn't got used to the interaction tends to listen the feedback carefully until it is finished. Therefore, a promising metric to detect the habituation is the *time interval from one voice command to the next command*. As the user is getting used to the interface, the time interval is expected to become shorter.

At this point, we introduce three kinds of time spent within the voice interface with mixed-initiative interaction.

Time of user speech (T_{user_speech}) **:** Time from the user starts speaking a voice command till he finishes speaking. T_{user_speech} may vary depending on the length of the voice commands, but the variation is not so significant.

Time of system process ($T_{sys\_process}$) **:** Time spent by the system to recognize a voice command from the user, synthesize the feedback voice and execute an appliance operation. $T_{sys\_process}$ has no significant variation, since it depends on the system implementation.

Time of system speech (T_{sys_speech}) **:** Time from the system starts speaking a feedback until the system receives the next voice command from the user.

For every voice interaction, the above three types of time (intervals) appear in the order of $T_{user\_speech} \rightarrow T_{sys\_process} \rightarrow T_{sys\_speech}$, periodically. The first two ($T_{user\_speech}, T_{sys\_process}$) do not vary so much, but T_{sys_speech} varies depending on the user's experience.

Thus, we propose to use *the sum of* T_{sys_speech} *within a complete sequence of voice interactions*, denoted by M_{sys_speech}, to evaluate the habituation effects. The metric M_{sys_speech} would characterize user's habituation from a perspective of *time efficiency*.

3.3 M2 : Number of Support Commands

In addition to the voice commands, our system can recognize a set of *support commands* to help the user learn the usage of the system. Typical support commands include;

Menu : If the user speaks this command, the system enumerates a list of available appliances by speech.

Help : If the user speaks this command, the system feeds back the usage of the voice interface.

We suppose that the non-expert users tend to use these support commands, frequently. Therefore, counting the number of support commands may capture the habituation of the user.

Therefore, we define a metric, *the number of support commands* (M_{sup}), as the total number of the support commands issued within a complete sequence of voice interactions. As the user is getting used to the voice interface, M_{sup} is expected to be reduced. The metric M_{sup} would characterize user's habituation from the viewpoint of *usage*.

Table 1: Appliance operations of each task

appliance	Task 1	Task 2
curtain	Open	Open
fan	Power on Maximize air volume –	Power on Maximize air volume Swing mode
TV	–	Power on Switch input mode to PC
air-conditioner	–	Turn on cooling mode

3.4 M3 : Number of Mistakes

We consider that *mistakes* in the voice operation would reflect the experience of the user. That is, we suppose that non-expert users tend to make more mistakes. Hence, counting the *number of mistakes* occurred would be a good metric to capture the habituation.

In the proposed system, the mistakes are caused by the following two kinds of matters.

Mis-recognition by system : The system fails to recognize the user speech due to the accuracy of the voice recognition engine, or the ambiguity of user's pronunciation.

Wrong vocabulary or timing of user : The system fails to recognize the user speech due to the poor vocabulary of the system, or the speech timing of the user.

We define a metric, *the number of mistakes* (M_{miss}), as the total number of mistakes count within a complete sequence of voice interactions. We suppose that M_{miss} will be reduced as the user is getting used. The metric M_{miss} would characterize user's habituation from the viewpoint of *reliability*.

4. EVALUATION EXPERIMENT

4.1 Overview

To evaluate the proposed three metrics empirically, we have conducted an experiment, where subjects operate the CS27-HNS (see Section 2.1) using the proposed system.

Seven subjects participated in the experiment. All of them were 20's years old, and none of them was familiar with the voice interface. First, the subjects practiced the voice operation to turn on a light. Next, we asked them to perform two kinds of tasks: Task 1 and Task 2. Table 1 summarizes the concrete appliance operations to be executed in the tasks. For each of Task 1 and Task 2, we instructed the subject to try the task for *five times*, in order to see how the subjects familiarized themselves to the system. We reset all the appliances after every trial was finished.

In the experiment, we measured the proposed three metrics.

M_{sys_speech} **:** The total time of system speech

M_{sup} **:** The total number of the support commands

M_{miss} **:** The total number of the mistakes

The metric M_{sys_speech} measured in the i-th trial of the task is denoted by $M^i_{sys\_speech}$ $(1 \leq i \leq 5)$. Similarly, we introduce the notations M^i_{sup} and M^i_{miss}.

4.2 Procedure of Experiment

1. We explained background and objective of the experiment to all the subjects, and obtained their informed consents.
2. We explained the usage of the voice interface and gave instruction manuals of the tasks to the subjects.
3. The subjects performed the preliminary practice, Task 1 and Task 2 with the voice interface.

4.3 Results

Figure 1 shows a graph plotting the total time of system speech, $M^i_{sys\_speech}$, taken by each user to perform the i-th trial of Task 1 $(1 \leq i \leq 5)$. Figure 2 plots the total number of support commands, M^i_{sup}, issued by each subject in i-th trial of Task 1. Figure 3 plots the total number of mistakes, M^i_{miss}, caused by each subject in the i-th trial of Task 1. In the graphs, A, B, ... ,G represent the 7 subjects.

We explain the result of Task 1. It can be seen in Figure 1 that for every subject, the value of $M^i_{sys\_speech}$ is significantly reduced as the number of trials increases. We can also see in Figure 2 that for every subject, the value of M^i_{sup} converges to 0 as the subjects repeated the task. Thus, our hypothesis in Sections 3.2 and 3.3 are validated. However, as for the number of mistakes M^i_{miss} in Figure 3, we cannot see any significant correlation of M^i_{miss} to the habituation. We found the similar observation in Task 2 as well, which was omitted due to limited space.

4.4 Metrics for Characterizing Habituation

As seen in Figures 1, the total time of system speech (M_{sys_speech}) of all subjects were reduced as they repeated the tasks. Although some individual differences exist, M_{sys_speech} can well characterize the habituation effects. Thus, in the proposed voice interface, we can say that the subjects improve their performance with respect to the *time efficiency*, as they get used to the interface. Indeed, we expected that M_{sys_speech} in Task 2 would be larger than that in Task 1 (in Figure 1). However, there was no significant difference. We consider that this is because the habituation obtained in Task 1 was taken over to the Task 2. From these observations, there seems to be two kinds of habituation. One is the *habituation within a task*, which is obtained by the repeat of the same routine. Another is *habituation between tasks*, which is globally obtained by performing various tasks.

The total number of support commands (M_{sup}) in Figure 2 converged to 0 as the subjects repeated the tasks. Therefore, we consider that M_{sup} can characterize the habituation effects as well. There was no significant differences between in two tasks. This fact implies that the subjects were able to learn the usage of the system and the appliance commands very quickly, with at most one execution of the support command. We consider that this is because the usage of our system was quite easy and we did not use so many appliances in the experiment.

On the other hand, the total number of mistakes (M_{miss}) shown in Figures 3 could not explain the habituation well, despite of our expectation. Through our investigation, we observed that every subject was likely to make different kinds of mistakes before and after the habituation. Before the habituation, the mistakes often occurred by misrecognition of the system. In the early stage of the experi-

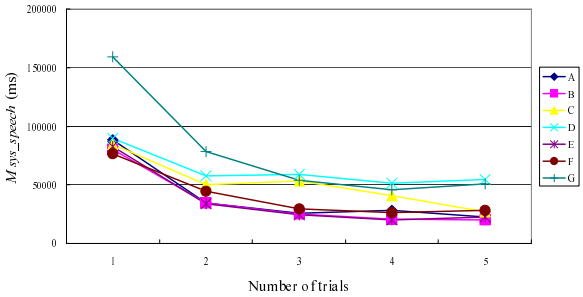

Figure 1: $M^i_{sys\_speech}$ in Task 1

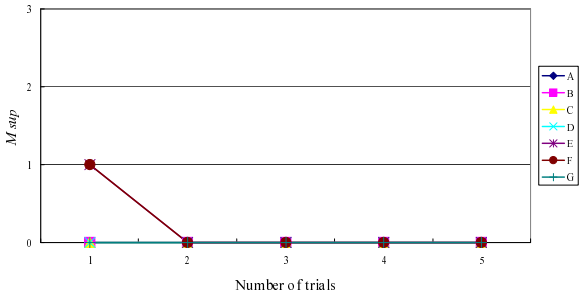

Figure 2: M^i_{sup} in Task 1

ment, words of the subjects were often unclear and the system could not understand the words correctly. As the subjects repeated the tasks, this problem was resolved by subject's intention to speak the commands clearly and slowly, especially for confusing words like "on" and "off". On the other hand, after the subjects got used, the mistakes occurred by the lack of system capacity. The input from the habituated subjects was so fluent that the system could not catch up with the input. Specifically, while the system was in the speech recognition process, the system could not accept the next speech. However, the fluent subjects were likely to speak the next command without waiting for the appropriate timing, which resulted in the failure of interaction. To cope with the problem, we need to optimize the performance of voice recognition, or extend the system so that it can process the input and recognition in parallel.

4.5 Automatic Detection of Habituation

We interviewed the subjects about "at which trial (out of five) in each task they felt the habituation". As a result, subjects A to G respectively felt the habituation at 3rd, 2nd, 2nd, 3rd, 2nd, 4th, 3rd trials in Task 1. As for in Task 2, they felt the habituation at 3rd, 3rd, 2nd, 5th, 3rd, 4th, 3rd trials, respectively. Based on this, we try to consider the *automatic detection* of habituation. Suppose now that a subject felt the habituation at k-th trial, and that we try to detect the habituation effect by using the metric M_{sys_speech}. We here introduce a difference $H_{time}(k) = M^{k-1}_{sys\_speech} - M^{k}_{sys\_speech}$, which reflects the *gain* of the habituation effect obtained within the k-th trial. When this gain is converged to a certain small value, we would say that the subject reached a certain level of the habituation. Comparing the experimental results and the answers of the

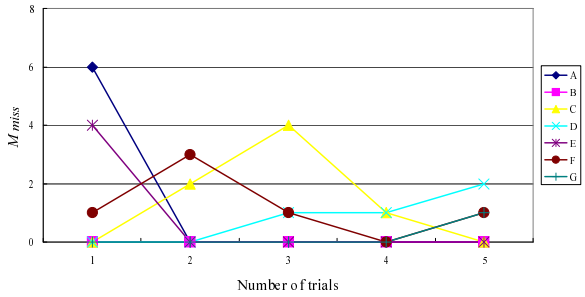

Figure 3: M^i_{miss} in Task 1

interview, we have derived a fact that the average value of $H_{time}(k)$ was about 5.6 seconds. This allows us to detect the habituation empirically, when $H_{time}(i)$ is close to 5.6 seconds for arbitrary i. Thus, the system can detect the habituation automatically, by recording and monitoring the value of the metric of M_{sys_speech}.

To use M_{sup} for the automatic detection, we need more data to establish the detection logic. We leave this for our future study.

5. CONCLUSION

In this paper, we have presented a method to evaluate the user's habituation in interactive voice interface for the home network system (HNS). We proposed three kinds of metrics, (M_{sys_speech}) the total time of system speech, (M_{sup}) the total number of support commands, and (M_{miss}) the total number of mistakes, to capture the habituation effect quantitatively. The experimental results show that M_{sys_speech} and M_{sup} can well characterize the habituation effect of the voice interface for the HNS. Our future works include determining the specific threshold value to detect the user habituation, and implementing the concrete adaptation mechanism of the system feedback.

6. ACKNOWLEDGMENTS

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research(B) (No.23300009), Young Scientists (B) (No.21700077), Research Activity Start-up (No.22800042)], and Hyogo Science and Technology Association.

7. REFERENCES

[1] T. Kawahara and M. Araki. *The voice interaction system*. Ohmsha, 2006.

[2] N. Matsubara, K. Egami, H. Igaki, and M. Nakamura. Interactive voice interface for eliciting and estimating implicit user requirements in home network system. *TECHNICAL REPORT OF IEICE*, 109:061–066, 2010.

[3] H. Morikawa. Ubiquitous network and the role of wireless. *The Journal of the Institute of Electronics, Information, and Communication Engineers*, 87(5):356–361, 2004.

[4] M. Nakamura, A. Tanaka, H. Igaki, H. Tamada, and K. Matsumoto. Constructing home network systems and integrated services using legacy home appliances and web services. *International Journal of Web Services Research*, 5(1):82–98, January 2008.