Robust Speech Dialog Management System for Mobile and Car Applications

Nobuo Hataoka[1], Hirohiko Sagawa[1], Yasunari Obuchi[2]
Masahiko Tateishi[3], Ichiro Akahori[3]
Jeongwoo Ko[4], Fumihiko Murase[4], Teruko Mitamura[4] and Eric Nyberg[4]

[1]Central Research Laboratory, [2]Advanced Research Laboratory, Hitachi Ltd, Kokubunji, Tokyo 185-8601, JAPAN, {hataoka, h-sagawa}@crl.hitachi.co.jp, obuchi@rd.hitachi.co.jp
[3]Research Laboratories, DENSO CORPORATION, Nisshin, Aichi 470-0111, JAPAN, mtatei@rlab.denso.co.jp, iakahori@its.denso.co.jp
[4]Language Technologies Institute, Carnegie Mellon University Pittsburgh, P.A. 15213, U.S.A. {jko, fmurase, teruko, ehn}@cs.cmu.edu

ABSTRACT

In Car Information Service Systems (Car Telematics), Speech Dialog Interfaces are important from safety viewpoints. However, there are many technical problems such as flexible dialog management, robust task management, and intelligent user assistance. We have proposed the CAMMIA (Conversational Agent for Multimedia and Mobile Information Access) to solve these technical problems[1][[2][3].

In this paper, we propose robust dialog management strategy and system architecture consisting of a Dialog Management and a Task Management, separately. Finally, we report the system evaluation results to confirm effectiveness of the proposed system architecture and dialog management for the mobile use.

Keywords: Car Telematics, Human Machine Interfaces (HMIs), Dialog Management (DM), Automatic Speech Recognition (ASR), Text-to-Speech (TTS), VoiceXML (Voice eXtensible Markup Language) Interpreter (VXI), Task Management (TM)

1. INTRODUCTION

A dialog system might utilize different dialog representations of varying degrees of complexity, depending on the nature of the task. Simple approaches involve the use of nested menus or transition networks, which restrict the user to a fixed set of pre-defined dialog exchanges. More sophisticated systems support mixed initiative dialogs, where the exact number of exchanges is not known in advance. Some tasks (e.g. navigation or route guidance) require dynamic creation of dialogs at run time, based on access to a remote information server.

Our final goal is to provide a flexible and efficient Human Machine Interface (HMI) using Speech Dialog for mobile applications. To do so, we extend VoiceXML[4] for large-scale dialog systems, which need large vocabularies and a variety of grammars. We envision a mobile application environment (e.g. a mobile information service system) where an embedded speech recognizer[5] and VoiceXML Interpreter (VXI) are connected to remote servers that support a variety of information-seeking tasks (car navigation, restaurant information, voice-activated control, etc.). Some similar research works have been reported[6], but the flexible mobile interface of our approach is useful for the actual environments.

To realize flexible speech dialog interfaces, the following issues are typical technical challenges:

- *Flexible dialog management.*
  The system must deal with a task/topic change/switching smoothly. To do this demand, the architecture should have a push&jump abilities/functions. However, the current VoiceXML cannot cope with this smooth task movement. Therefore, we have investigated the extended VoiceXML to handle this issue[2]. Moreover, the dialog system should have abilities in that the system provides useful new information to the users depending on the dialog context using user profile and local area information.
- *Robust system architecture for network connection loss.*
  In the real applications, the network connection is not stable in mobile environments such as in a tunnel etc. The speech dialog system should cope with this kind of bad situations. The desirable system should keep previous dialog history and when the communication network comes back and becomes available, the system can return to the previous dialog soon. This ability/function can be implemented from both architecture viewpoints and dialog managemant viewpoints.

We have proposed the CAMMIA system introducing DialogXML, an extension to VoiceXML that supports a declarative language for dialog scenarios and ScenarioXML, a straightforward combination of DialogXML with the template-filling mechanism of Java Server Pages (JSP)[1]. In this paper, we describe the extension of the CAMMIA to handle the flexible dialog management and network loss. To do so, we propose a robust system architecture consisting of the Dialog Management(DM) and Task Management(TM) separately so that tasks interrupted can be retried when a network connection-loss occurs between a client and a server.

2. CAR TELEMATICS SERVICES

2.1 System Concept for Network Applications

The Car Telematics is a new service terminology using car navigation terminals to connect networks. Figure 1 illustrates the total service system concept, which consists of three parts, e.g. Terminal/Client, Network, and Center/Server. For the Terminals, the sophisticated HMIs are required to input various inquiries and to get information delivered from the Center. The Network usually is the Internet and via the Internet, the user's requests are transferred to a related Web server on the Center and the required information will be provided from the Center to users via Networks and Terminals.

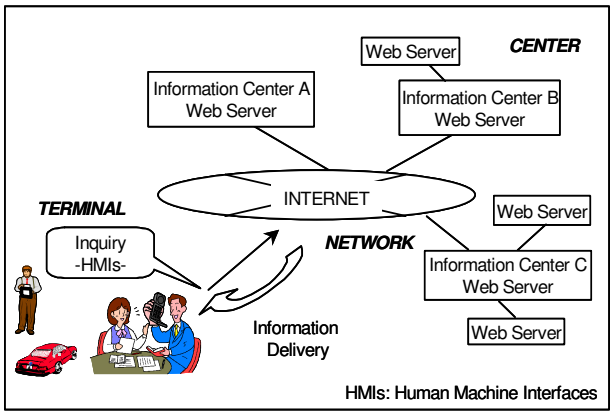

Figure 1: Car Telematics System Concept

2.2 Voice Portal Architecture

In the car, HMIs based on speech processing such as Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) are essential to provide a safe driving environment. Car Telematics using speech technology is thought of as one of the Voice Portal services. The Voice Portal service concept is based on the VoiceXML gateway. In the VoiceXML gateway, VXIs are implemented for the WWW to be accessed by voice. The VoiceXML is a W3C (WWW Consortium) standard to provide dialog functions to voice systems. The VoiceXML Gateway also incorporates ASR/TTS engines; sometimes, terminals such as car navigation systems also incorporate ASR/TTS engines.

3. CAMMIA SYSTEMS

3.1 CAMMIA ARCHITECTURE

Figure 2 shows a CAMMIA system architecture consisting of four layers, a User Interface Layer, a Dialog Management Layer, a Task Management Layer, and an Application Layer. By this architecture, the speech dialog system can handle dialog management issues between a user and the system and task management issues separately. This architecture leads many advantages from the software handling viewpoints.

(1) User Interface Layer: This layer is responsible for interaction with the user via a variety of input/output modalities such as speech and a touch panel. VXI is an interpreter of VoiceXML sequences written using speech interactions between users and systems.

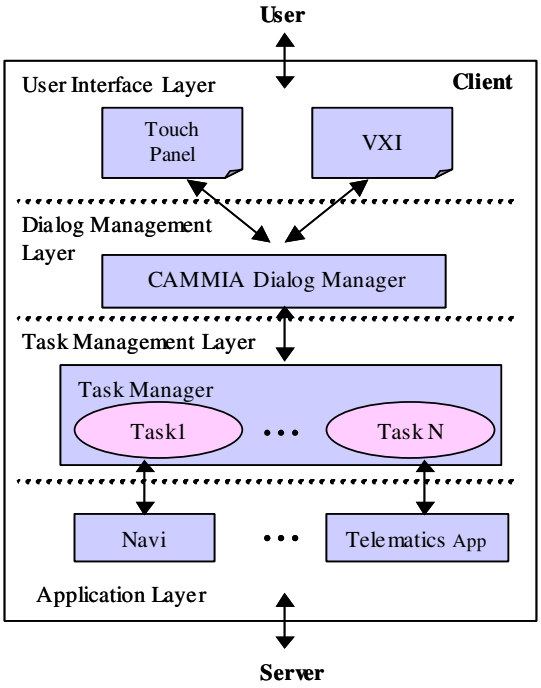

Figure 2: CAMMIA Architecture

(2) Dialog Management Layer: The Dialog Management layer is responsible for supporting multiple ongoing topics of dialogs between users and systems. Dialogs are represented as ScenarioXML and DialogXML[1][2]. DialogXML is used to represent all possible dialog sequences, including task-switching interactions which link to other dialogs. Input from the user may interrupt or cancel the current dialog. This Dialog Management feature can handle the task/topic changes/switching such as "Find Japanese restaurant nearby" and "By the way, is there any parking lot?" This ability can be realized using a stack memory in VoiceXML.

(3) Task Management Layer: The Task Management layer has been added to explicitly represent tasks and their states, so that tasks which are interrupted or which fail can be retried or restarted when a network connection becomes available.

(4) Application Layer: The Application layer is between the Task Managemen Layer and the Server via. remote networks (e.g. the Internet). This Layer is responsible for servicing a particular task such as a navigation task or Telematics applications, and also servicing all information requests from the Dialog Manager and the Task Manager (e.g., "find Italian restaurants near current location"). The Application Layer does download information from remote networks.

3.2 Flexible Dialog Management

(1) Task/Topic Switching: Figure 3 shows an example of dialog task/topic changes. Using the proposed DM algorithm, we have found that the task change from Route Guidance to Parking Lot Info is processed correctly in the CAMMIA

prototype. The Destination Set Task provides a necessary prompt to users, and the dialog task for Route Guidance was set according to the recognized results by a dynamic generation of dialog state. We see that smooth task movement is possible if the DM can maintain appropriate multi-task context, and return from the Parking Task to the Route Guidance Task correctly once the subdialog is complete.

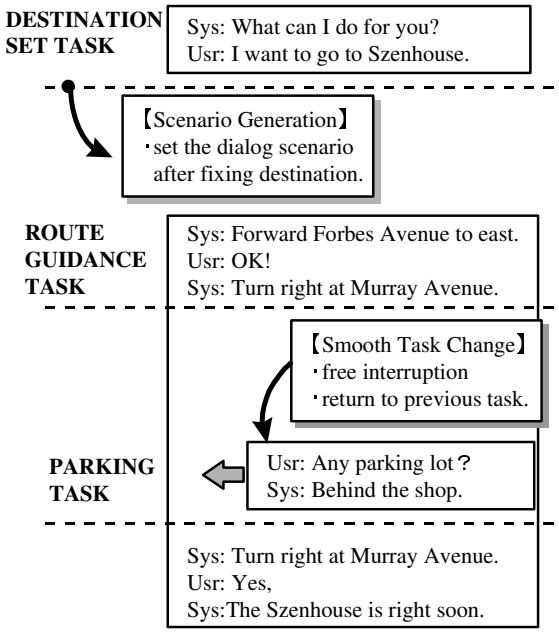

Figure 3: Example of Dialog Sequence

(2)Confirmation Push: The CAMMIA dialog system has abilities in that the system can provide useful new information to the users depending on the dialog context and locations using user profiles and local area information. For example, the system may suggest famous restaurants for lunch or suggest famous sightseeing places near the current location. We call this feature as "confirmation push." The user can modify confirmation push items in user preferences.

4. EVALUATION EXPERIMENTS

An evaluation experiment has been carried out to evaluate flexible dialog management such as Task/Topic Switching and Confirmation Push, and robust task management for network connection loss.

4.1 Experiment Setup

The number of subjects in the experiments was 10 Japanese adults, 5 males and 5 females. Each subject used the CAMMIA evaluation prototype for about 10 minutes as a training session before the experiment. The subjects were asked to press a push-to-start button (a function key in the keyboard) to speak. To minimize the effect of task ordering, half of the subjects started the tasks in the opposite order. The subjects were able to look at the screen, but were asked to use only speech interface.

We used a Microsoft speech recognition software to recognize continuous speech using a network grammar. The recognition vocabularies are 4 locations (Tokyo, Shizuoka, Izu, Nagoya), 10 restaurant genre (Japanese, Italian, Italia, French, France, Chinese, etc.), 2 dates (today, tomorrow) and various expressions such as "Please le me know …," "Please show me …," "Is there any …?," "Any …?," and so on.

4.2 Evaluation Tasks

Three tasks were used as below. After experiments, subjects filled out satisfaction questionnaire using a 7-point MOS scale.

Task 1: Confirmation push

The subjects were asked to find a nearby Japanese (and Italian) restaurant which had a parking lot and set it as a destination for navigation. Figure 4 shows a task image which was shown to the subject before the evaluation experiment start. Five restaurants found in the database were shown to the subject and the subjects had to search them to set a restaurant which had a parking lot and was open. Among the five restaurants, the only one restaurant had a parking lot and was open.

In the Task 1, there were two subtasks such as Task A and Task B. The Task A was without the confirmation push feature, and the Task B had the confirmation push feature. The system with the confirmation push feature (Task B) showed subjects the reason why the search was failed because no parking lots and closed time. The system without the confirmation push feature (Task A) did not show the reason the search was failed and came back to the original search starting point and the subject had to continue to search for another restaurant.

Task 2: Task/Topic Switching

The subjects were asked to find a nearby fruit garden which had a parking lot and set it as a destination to the navigation map. To set the destination, the subject had to change a task asking the weather information, and after getting the necessary information the subject had to return the previous task.

There were two subtasks, Task I and Task K. The Task I did not have the Task/Topic Switching feature, and the Task K had the feature.

Figure 4: Example of Task

Task 3: Robust Task Management for Network Loss

The subjects were asked to find the weather in Tokyo for today and for tomorrow. In this task, the communication network connection loss happened for about 20 seconds, and the subject had to wait until the network connection became available.

There were two subtasks: one with task management (Task Y) and the other without task management (Task X). Without task management, the subject had to start again from the beginning. With task management, the Task Manager kept the user's requests and showed the search results to the subject when the network connection became available.

4.3 Evaluation Results

Recognition Rate

Table 1 shows recognition rates for 10 subjects (5 males and 5 females). In this table, 441 and 407 show the number of subject utterances, male and female respectively. The recognition rate of females was lower than that of males because of low microphone level.

The number of dialog turns

Table 2 shows the average number of user utterances (turns). Task A, Task I, and Task X do not have the proposed features, confirmation push, task switching, and task management for network loss, respectively. As can be seen, the subjects could finish the task B, task K, and Task Y with less user dialog turns comparing to Task A, Task I, and Task X, respectively.

Satisfaction Opinion Comments

Most subjects showed unsatisfactory comments because tasks were not reasonable. For example, the confirmation push is a baseline function, so to compare to without-confirmation-push is not meaningful. However, a couple of subjects made comments that the system would be useful if enough training was performed.

Table 1: Recognition Rate

5 males	91.4%(403/441)	5 females	81.1%(330/407)

Table 2: Average No. of User Dialog Turns

Task		Average no. of dialog user turns	
		male	female
Task 1	Task A	30.8	25.6
	Task B	26.4	17.2
Task 2	Task I	10.0	14.4
	Task K	5.6	7.8
Task 3	Task X	5.0	6.2
	Task Y	2.0	3.0

5. SUMMARY and FUTURE WORK

This paper described the flexible and robust Dialog Management (DM) system based on the four-layer architecture. The extension of VoiceXML makes the system more eligible to handle real dialog sequences. We have implemented the proposed architecture as the evaluation prototype and found real-time processing performance. By the evaluation experiments, we have found the effectiveness of the proposed flexible dialog management and robust task management to handle real user dialogs/utterances and the network connection loss.

For the future work, we are planning to evaluate the system by more subjects and using other types of intelligent user assistance with driving simulator where the user is moving to new places. To check the driver distraction problems using the proposed system architecture is an interesting future research issue.

ACKNOWLEDGEMENTS

The authors thank Dr. N. Sato and Dr. T. Honma of Hitachi Central Research Lab, for their valuable support.

REFERENCES

[1] E. Nyberg, T. Mitamura, P. Placeway, M. Duggan and N. Hataoka, "DialogXML: Extending Voice-XML for Dynamic Dialog Management, " Proc. of HLT-2002 (2002)

[2] Y.Obuchi, E. Nyberg, T. Mitamura, S. Judy, M. Duggan and N. Hataoka, "Robust Dialog Management Architecture using VoiceXML for Car Telematics Systems, " pp.83-96, DSP for In-Vehicle and Mobile Systems, edited By H. Abut, J.H.L. Hansen and K. Takeda, Springer (2004)

[3] M. Tateishi, K. Asamai, I. Akahori, S. Judy, Y. Obuchi, T. Mitamura, E. Nyberg, and N. Hataoka, "A Spoken Dialog for Car Telematics Services," pp.47-64, DSP for In-Vehicle and Mobile Systems, edited By H. Abut, J.H.L. Hansen and K. Takeda, Springer (2004)

[4] VoiceXML2.0, 04/24/02, http://www.w3.org.

[5] N. Hataoka, K. Kokubo, Y. Obuchi, and A. Amano, "Development of Robust Speech Recognition Middleware on Microprocessor," Proc. of IEEE ICASSP1998, pp.II837-II840 (1998)

[6] H. Meg, e al., "Bilingual Chinese/English Voice Browsing based on a VoiceXML Platform," Proc. of IEEE ICASSP2004, pp.III-769-III772 (2004)