Robust Speech Dialog Interface for Car Telematics Service

Nobuo Hataoka and Yasunari Obuchi
Central Research Laboratory
Hitachi Ltd.
Kokubunji, Tokyo 185-8601, JAPAN
hataoka@crl.hitachi.co.jp, obuchi@rd.hitachi.co.jp

Teruko Mitamura and Eric Nyberg
Language Technologies Institute,
Carnegie Mellon University
Pittsburgh, PA 15213, U.S.A.
{teruko , ehn}@cs.cmu.edu

Abstract— **In this paper, we describe new consumer services based on speech processing technologies to support a new digital/mobile era of ubiquitous communication. First, we propose a compact and noise robust embedded speech recognition middleware implemented on microprocessors focused on sophisticated HMIs (Human Machine Interfaces) for car information systems (i.e. Car Telematics). Second, we report on a novel and sophisticated Dialog Management/Manager (DM) system based on VoiceXML (Voice eXtensible Markup Language), called CAMMIA (Conversational Agent for Multimedia Mobile Information Access). The proposed DM will handle two important issues: an automatic generation scheme for lexicons and grammars, and an effective combination/merger between Automatic Speech Recognition (ASR) and Natural Language Processing (NLP). The new DM scheme has been evaluated for an application of the Car Telematics service task after integration with ASR and a VoiceXML interpreter (VXI).**

Keywords- Human Machine Interfaces (HMIs), speech technologies, Automatic Speech Recognition (ASR), Text-to-Speech (TTS), embedded middleware, Natural Language Processing (NLP), VoiceXML Interpreter (VXI), Dialog Management/Manager (DM)

I. INTRODUCTION

As information technology expands into the mobile environment to provide ubiquitous communications, an intelligent interface will be a key element to enable mobile access to networked information. For mobile information access, HMIs using speech might be the most important and essential application, as speech interfaces are more effective for small, portable devices. Mobile terminals such as cellular phones, PDAs (Personal Digital Assistants) and Hand-held PCs (Personal Computers) are already connected to networks such as the Internet to access information from web services. For effective use of mobile information access, speech processing and image processing will be key technologies for intelligent mobile terminals.

Due to improvements in microprocessor performance, it has been possible to implement multimedia-processing technologies using software on microprocessors and/or DSP (Digital Signal Processing) chips. This software, called *middleware*, is a kind of code library that connects hardware and end-user applications. Middleware enables developers and users to use various media processing technologies in different mobile applications, such as car navigation systems and hand-held PCs, using a single microprocessor.

In this paper, we first report on embedded speech middleware[1][2] which enables sophisticated HMIs for multimedia systems. The speech middleware has been used widely for car navigation systems, mobile information equipment, and game machines.

Second, we discuss the creation of a speech dialog interface for HMIs, and report on a new dialog management system called CAMMIA[3][4]. Speech Dialog is an essential function to realize sophisticated and human-centered HMIs. A complete system based on dialog management needs not only ASR and Text to Speech (TTS) technologies, but also an efficient integration with NLP. VoiceXML[5] has been proposed as a standard markup language by the W3C (WWW Consortium) to realize dialog sequences that integrate ASR/TTS with Speech DM[6].

Our goal is to extend VoiceXML for large-scale dialog systems, which require 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 and VXI are connected to remote servers that support a variety of information-seeking tasks (car navigation, restaurant information, voice-activated control, etc.). While we prefer the simplicity of VoiceXML for run-time processing, a more supportive representation is required for creation of dialog scenarios.

We have constructed a prototype that integrates the DM System with ASR and a VXI. The proposed architecture is independent and neutral with respect to ASR; any approach that can be integrated via the VoiceXML grammar mechanism can be utilized. This paper describes DialogXML, an extension to VoiceXML that supports a declarative language for dialog scenarios. We also introduce ScenarioXML, a straightforward combination of DialogXML with the template-filling mechanism of Java Server Pages (JSP)[7]. ScenarioXML provides a mechanism for dynamic generation of content (e.g. navigation directions accessed from a remote database).

II. CAR TELEMATICS SERVICE

A. System Concept for Network Applications

Car Telematics refers to a new service concept where mobile terminals (e.g. car navigation systems, cellular phones) are used to connect to networked information services. Figure 1 illustrates the total service system concept, which consists of three parts, e.g. Terminal/Client, Network, and Center/Server. For the Terminals, sophisticated HMIs are required to handle various inquiries and to deliver information from the Center using speech and image input/output. The Network is typically the Internet; and via the Internet, the user's requests are transferred to related Web servers at the Center, and required information will be provided from the Center to users via Networks and Terminals.

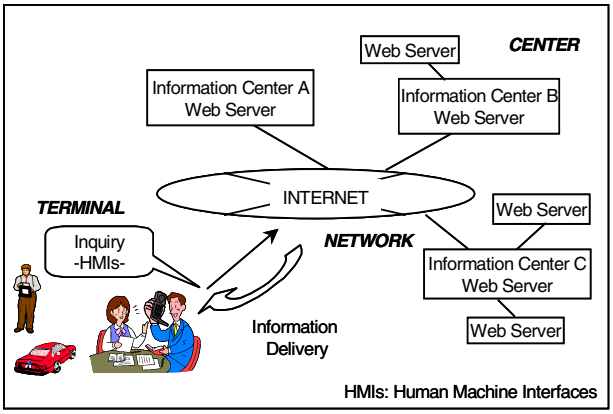

***Figure 1**: Service System Image(Terminal, Network, and Center)*

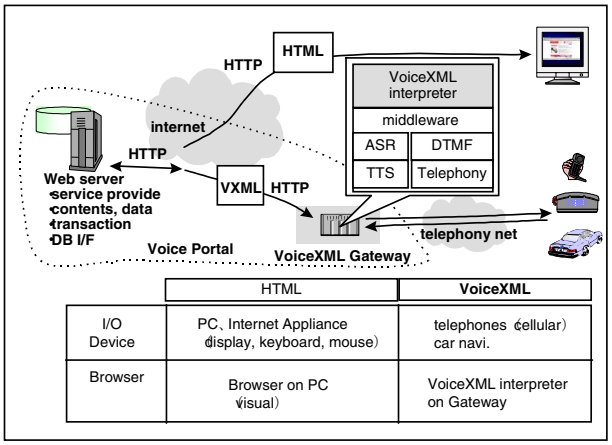

***Figure 2**: Voice Portal Concept based on VoiceXML Gateway*

B. Voice Portal Architecture

In the car, HMIs based on speech processing such as ASR and TTS are essential to provide a safe driving environment. Car Telematics using speech technology is thought of as one of the Voice Portal services. Figure 2 shows the Voice Portal service concept 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 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.

III. EMBEDDED SPEECH MIDDLEWARE

A. Speech Middleware Specifications

Middleware is a kind of code library which connects hardware and user applications. We have developed speech recognition middleware on RISC microprocessors. The middleware helps service providers to easily create applications using speech recognition. We have implemented middleware on our RISC microprocessor called the *SuperH* RISC Engine (SH-3 and SH-4). Table 1 shows a specification for the SH-3 and ASR middleware.

The operation speed and memory size are limited. We use phonemic speech segments such as HMM (Hidden Markov Model) units. To reduce calculation burden, semi-continuous HMMs and tied mixtured 3-dimensional models have been used. Moreover, we developed several approximation search techniques and memory assignment of acoustic models to save calculation time. Thus, the middleware achieved a performance of around 93% recognition rate for a 2000-word vocabulary with 0.6-second response time.

item	specification
Operation Speed	SH-3: 60 MHz
External Bus	SH-3: 60 MHz / 32 bit
Speech Model	Phonemic Speech Units Semi-continuous HMM
Sampling	11.025 kHz / 16 bit
Frame Length / Period	20 ms / 10ms
Processing Time	14 ms / frame
Response Time	~ 0.6 sec
Vocabulary Size	2000 words
Memory Size	256 kByte (phonemic model etc.) 500kByte (work memory)

***Table1**: Specification of ASR Middleware*

Figure 3 shows a block diagram for the ASR middleware. The speech signal is processed by the A/D (Analog to Digital) Converter with an LPF (Low Pass Filter) for sampling. Speech parameters are extracted via speech analysis. In this system, 13-order LPC (Linear Predictive Coding) cepstrums and 13-order differential cepstrums are used as speech parameter. After speech detection, the speech interval detected and

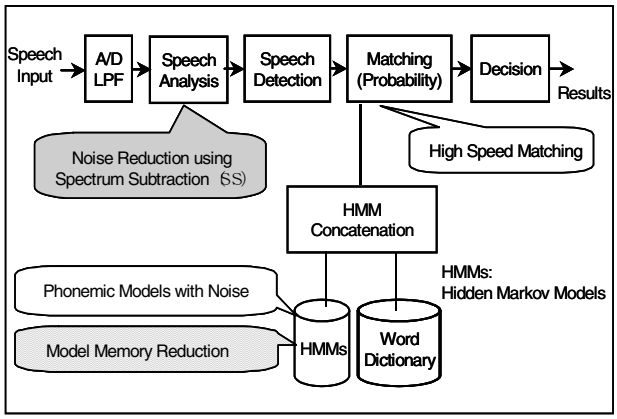

***Figure 3**: Block-diagram of ASR System*

associated speech data are processed via a HMM matching/search. The HMM matching method is popular in the speech recognition field. For the middleware, an effective noise reduction method using Spectral Subtraction (SS) and a novel HMM memory reduction technique has been developed.

B. Evaluation Board Architecture

Figure 4 shows a photo image of an evaluation board that the developed speech middleware has been implemented on using SH-3. Figure 5 shows the system architecture of the evaluation board. The SH-3 has 60MHz cycle process power. The fundamental middleware, which has 2000-word speech recognition ability, needs 256kByte ROM (Read Only Memory) as data/program memory and 500kByte RAM (Random Access Memory) as working memory. The input speech is digitized by an 11.025kHz-sampling A/D converter, and processed by the middleware via the interface bus. Finally, the recognized results are shown to display terminals through RS232C.

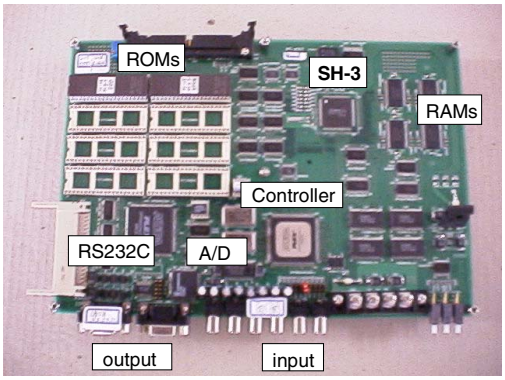

***Figure 4**: Photo Image of Evaluation Board*

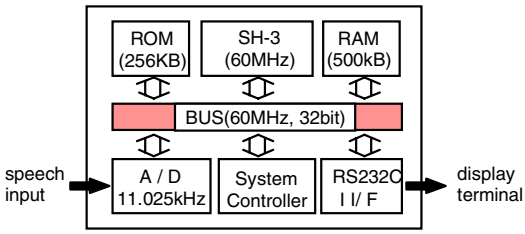

***Figure 5**: System Architecture of Evaluation Board*

C. Evaluation Results for Environmental Noise

To realize speech recognition middleware on microprocessors that have low CPU power and low memory capacity, compact noise handling techniques must be developed. In our evaluation, two types of robust speech recognition methods against noise have been developed. The first one is a noise inserted acoustic model and the second is the Spectral Subtraction (SS) method[1][2].

We evaluated the total performance of the recognition middleware/system using the prototype board shown in Figure 4. Table 2 shows evaluation results for car idling and speedway conditions. The recognition task was for 1700 words (Japanese railway station names) and the number of test words was 200 for idling and 100 for speedway-driving environments. The results of idling conditions showed no recognition-rate degradation on the evaluation board compared with simulation results.

***Table 2**: Evaluation Results using Prototype SuperH Board*

#	condition (no. of words)	S/N ratio(dB)	recognition rate	
			simulation	board
1	idling(200)	34.9dB	92.0%	92.0%
2	speedway(100)	16.6dB	85.0%	82.0%

IV. CAMMIA: DIALOG MANAGEMENT BASED ON VOICEXML

A. Current Challenges for Dialog Management

The use of VoiceXML with a suitable interpreter is an effective way to develop simple command and control systems. However, in order to use an existing VXI in a more complex environment with dynamic context and context switching, some enhancements to the VoiceXML representation and interpretation process are desirable.

Most VoiceXML dialog systems use simple context-free language models for user utterances. In order to support full natural language parsing, the VXI must be extended to handle grammar and lexicon elements that contain references to more sophisticated grammar and lexicon modules.

Natural dialog has a notion of state, and dialog designers often think in terms of state-transition networks. VoiceXML supports simple form and action pairs, but does not explicitly

model states and transitions among states. Although it is possible to author and maintain a set of large-scale dialogs directly using VoiceXML, it would be tedious to do so.

B. Desirable DM Architecture

We investigated a possible DM architecture that can smoothly handle task switching. The architecture should allow the system to "push" a subdialog or "jump" to another dialog on demand. However, the current VoiceXML only supports a single active dialog, and cannot handle transparent task switching. Therefore, we have investigated extensions to VoiceXML to address this issue. We propose a 3-tier hierarchical structure consisting of ScenarioXML, DialogXML, and VoiceXML [3][4].

C. Extending Dialog Management with NLP

One challenge is to incorporate NLP techniques for voice applications to improve flexibility and to generate grammars and lexicons for more realistic (non-menu) interaction. In this section we propose a new DM approach which incorporates NLP.

1) Total DM System

Figure 6 shows the CAMMIA system architecture, consisting of three functions (ASR, DM, and NLP) integrated via VoiceXML-based structures. The first prototype we have designed makes use of NLP as an offline process. The Speech Recognition Grammar (SRG) (which includes both a grammar and lexicon) is created in a pre-compilation step before actual use. The DM uses a suitable grammar and lexicon that are

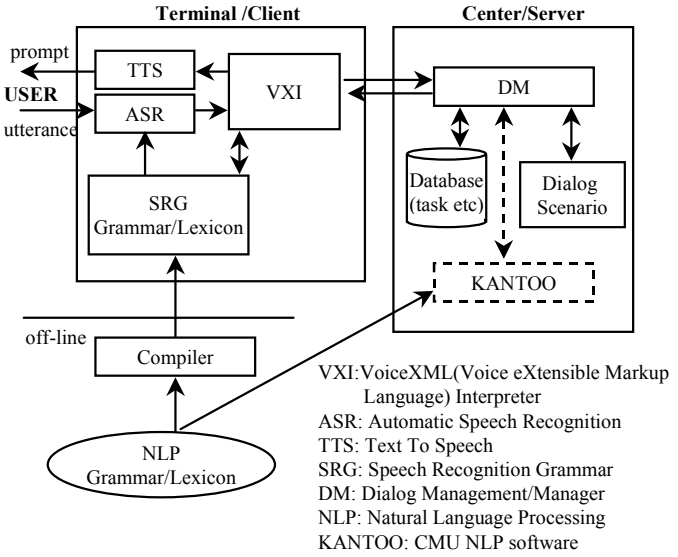

***Figure 6**: CAMMIA System Architecture*

pre-defined and stored in a table according to dialog task. We are currently exploring how to use NLP as an online capability, so that SRGs can be generated on the fly during the dialog. Currently we use the KANTOO system [8] for NLP.

2) 3-Tier DM Architecture

Figure 7 shows a proposed DM structure consisting of three hierarchical XML representations.

ScenarioXML is used to define the interactions required for a particular dialog. DialogXML (Figure 8) is used to represent all possible dialog sequences, including task switching interactions which link to other dialogs. The dialog designer writes ScenarioXML, which is compiled into DialogXML. Finally the DialogXML is compiled to VoiceXML to be interpreted by a speech recognition engine. The VoiceXML represents dialog utterances and actions for a single state.

In order to utilize NLP, the **grammar** must be handled freely and automatically. This means that the system must handle states and state-transitions. From this viewpoint, the dialog sequences are expressed by **arc** elements and **grammar definition,** and transitions by **push** elements are possible to be expressed in the DialogXML as shown in Figure 8.

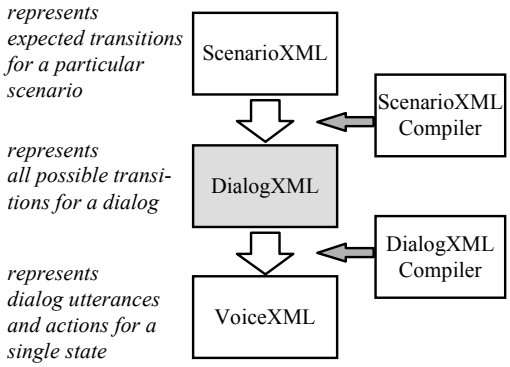

***Figure 7**: Proposed DM Structure*

```
<DialogXML>
.....
  <arc>
    <grammar src="directions_request.pat"
                        type="kantoo/patrick"/>
    <push dialog="directions-request.jsp"/>
  </arc>
  <arc>
    <grammar src="parking_request.pat"
      ........
```

***Figure 8**: Example of DialogXML*

Figure 9 shows an example of dialog sequence. Using the proposed DM algorithm, we have found that the task change from Route Guidance to Parking Lot Info is processed correctly. 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.

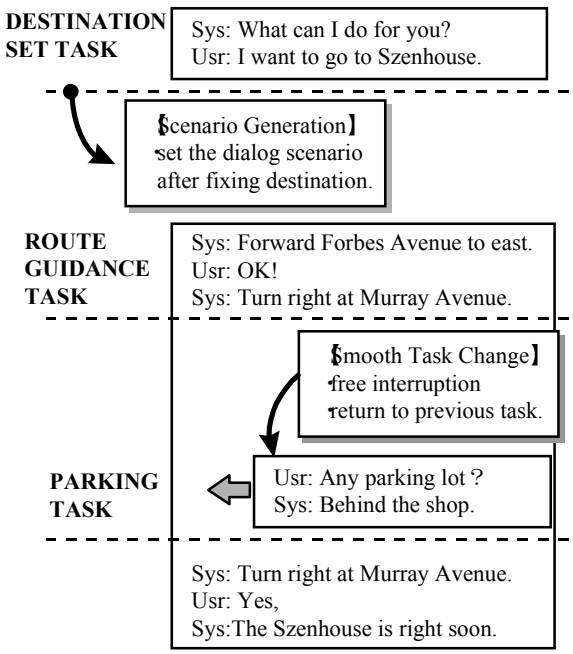

***Figure 9**: Example of Dialog Sequence*

V. SUMMARY

This paper described new consumer services based on speech processing technologies connected to the Internet. First, the compact and noise robust embedded speech recognition middleware has been developed. After large-scale evaluation using real noisy data, we have demonstrated noise robust speech recognition abilities. Second, we proposed the novel Dialog Management (DM) architecture based on a three-tiered extension of VoiceXML that is make it easier to represent complex combinations of dialogs. An evaluation has been conducted to demonstrate smooth transition between multiple active tasks.

ACKNOWLEDGMENT

The authors thank Mr. I. Akahori and Mr. M. Tateishi of DENSO Corporation for their valuable support.

REFERENCES

[1] N. Hataoka, K. Kokubo, Y. Obuchi, and A. Amano, "Development of Robust Speech Recognition Middleware on Microprocessor," *Proc. of IEEE ICASSP98*, pp.II837-II840, 1998.

[2] N. Hataoka, H. Kokubo, Y. Obuchi, and A. Amano, "Compact and Robust Speech Recognition for Embedded Use on Microprocessor," *Proc. of IEEE Multimedia Signal Processing(MMSP02)*, Dec.,2002.

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

[4] Y. Obuchi, E. Nyberg, T. Mitamura, M. Duggan, S. Judy, and N. Hataoka, "Robust Dialog Management Architecture using VoiceXML for Car Telematics Systems," *Proc. IEEE Workshop on DSP in Mobile and Vehicular Systems*, April, 2003.

[5] VoiceXML 2.0: http://www.w3.org/

[6] B. Carpenter, S. Caskey, K. Dayanidhi, C. Drouin, and R. Pieraccini, "A Portable, Server-Side Dialog Framework for VoiceXML," *Proc. of ICSLP02, Sept.*, 2002

[7] Jakarta Tomcat, http://jakarta.apache.org/tomcat.

[8] E.Nyberg, T.Mitamura, "The KANTOO Machine Translation Environment, " *Proc. of AMTA2002*, 2002.