

# A Conversational In-car Dialog System

Baoshi Yan<sup>1</sup> Fuliang Weng<sup>1</sup> Zhe Feng<sup>1</sup> Florin Ratiu<sup>2</sup> Madhuri Raya<sup>1</sup> Yao Meng<sup>1</sup>  
Sebastian Vargas<sup>2</sup> Matthew Purver<sup>2</sup> Annie Lien<sup>1</sup> Tobias Scheideck<sup>1</sup> Badri Raghunathan<sup>1</sup>  
Feng Lin<sup>1</sup> Rohit Mishra<sup>4</sup> Brian Lathrop<sup>4</sup> Zhaoxia Zhang<sup>4</sup> Harry Bratt<sup>3</sup> Stanley Peters<sup>2</sup>  
Research and Technology Center, Robert Bosch LLC, Palo Alto, California<sup>1</sup>  
Center for the Study of Language and Information, Stanford University, Stanford, California<sup>2</sup>  
Speech Technology and Research Lab, SRI International, Menlo Park, California<sup>3</sup>  
Electronics Research Lab, Volkswagen of America, Palo Alto, California<sup>4</sup>

## Abstract

In this demonstration we present a conversational dialog system for automobile drivers. The system provides a voice-based interface to playing music, finding restaurants, and navigating while driving. The design of the system as well as the new technologies developed will be presented. Our evaluation showed that the system is promising, achieving high task completion rate and good user satisfaction.

## 1 Introduction

As a constant stream of electronic gadgets such as navigation systems and digital music players enters cars, it threatens driving safety by increasing driver distraction. According to a 2005 report by the National Highway Traffic Safety Administration (NHTSA) (NHTSA, 2005), driver distraction and inattention from all sources contributed to 20-25% of police reported crashes. It is therefore important to design user interfaces to devices that minimize driver distraction, to which voice-based interfaces have been a promising approach as they keep a driver's hands on the wheel and eyes on the road.

In this demonstration we present a conversational dialog system, CHAT, that supports music selection, restaurant selection, and driving navigation (Weng et al., 2006). The system is a joint research effort from Bosch RTC, VW ERL, Stanford CSLI, and SRI STAR Lab funded by NIST ATP. It has reached a promising level, achieving a task completion rate of 98%, 94%, 97% on playing music, finding restaurants, and driving navigation respectively.

Specifically, we plan to present a number of features in the CHAT system, including end-pointing with prosodic cues, robust natural language understanding, error identification and recovery strategies, content optimization, full-fledged reponse generation, flexible multi-threaded, multi-device dialog management, and support for random events, dynamic information, and domain switching.

## 2 System Descriptions

The spoken dialog system consists of a number of components (see the figure on the next page). Instead of the hub architecture employed by Communicator projects (Seneff et al., 1998), it is developed in Java and uses flexible event-based, message-oriented middleware. This allows for dynamic registration of new components. Among the component modules in the figure, we use the Nuance speech recognition engine with class-based  $n$ -grams and dynamic grammars, and the Nuance Vocalizer as the TTS engine. The Speech Enhancer removes noises and echo. The Prosody module will provide additional features to the Natural Language Understanding (NLU) and Dialog Manager (DM) modules to improve their performance.

The NLU module takes a sequence of recognized words and tags, performs a deep linguistic analysis with probabilistic models, and produces an XML-based semantic feature structure representation. Parallel to the deep analysis, a topic classifier assigns  $n$ -best topics to the utterance, which are used in the cases where the dialog manager cannot make any sense of the parsed structure. The NLU module also supports dynamic updates of the knowledge base.

The DM module mediates and manages interac-

tion. It uses an information-state-update approach to maintain dialog context, which is then used to interpret incoming utterances (including fragments and revisions), resolve NPs, construct salient responses, track issues, etc. Dialog states can also be used to bias SR expectation and improve SR performance, as has been performed in previous applications of the DM. Detailed descriptions of the DM can be found in (Lemon et al., 2002) (Mirkovic and Cavedon, 2005).

The Knowledge Manager (KM) controls access to knowledge base sources (such as domain knowledge and device information) and their updates. Domain knowledge is structured according to domain-dependent ontologies. The current KM makes use of OWL, a W3C standard, to represent the ontological relationships between domain entities.

The Content Optimization module acts as an intermediary between the dialog management module and the knowledge management module and controls the amount of content and provides recommendations to user. It receives queries in the form of semantic frames from the DM, resolves possible ambiguities, and queries the KM. Depending on the items in the query result as well as configurable properties, the module selects and performs an appropriate optimization strategy (Pon-Barry et al., 2006).

The Response Generation module takes query results from the KM or Content Optimizer and generates natural language sentences as system responses to user utterances. The query results are converted into natural language sentences via a bottom-up approach using a production system. An alignment-based ranking algorithm is used to select the best

generated sentence.

The system supports random events and dynamic external information, for example, the system prompts users for the next turn when they drive close to an intersection and dialogs can be carried out in terms of the current dynamic situation. The user can also switch among the three different applications easily by explicitly instructing the system which domain to operate in.

### 3 Acknowledgement

This work is partially supported by the NIST Advanced Technology Program.

### References

- Oliver Lemon, Alex Gruenstein, and Stanley Peters. 2002. Collaborative activities and multi-tasking in dialogue systems. In *Traitement Automatique des Langues (TAL)*, page 43(2).
- Danilo Mirkovic and Lawrence Cavedon. 2005. Practical Plug-and-Play Dialogue Management. In *Proceedings of the 6th Meeting of the Pacific Association for Computational Linguistics (PACLING)*, page 43(2), Tokyo, Japan.
- National Highway Traffic Safety Administration NHTSA. 2005. *NHTSA Vehicle Safety Rulemaking and Supporting Research Priorities: Calendar Years 2005-2009*. January.
- Heather Pon-Barry, Fuliang Weng, and Sebastian Varges. 2006. Evaluation of content presentation strategies for an in-car spoken dialogue system. In *Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP)*, pages 1930–1933, Pittsburgh, PA, September.
- Stephanie Seneff, Ed Hurley, Raymond Lau, Christine Pao, Philipp Schmid, and Victor Zue. 1998. GALAXY-II: A Reference Architecture for Conversational System Development. In *International Conference on Spoken Language Processing (ICSLP)*, page 43(2), Sydney, Australia, December.
- Fuliang Weng, Sebastian Varges, Badri Raghunathan, Florin Ratiu, Heather Pon-Barry, Brian Lathrop, Qi Zhang, Tobias Scheideck, Harry Bratt, Kui Xu, Matthew Purver, Rohit Mishra, Annie Lien, Madhuri Raya, Stanley Peters, Yao Meng, Jeff Russel, Lawrence Cavedon, Liz Shriberg, and Hauke Schmidt. 2006. CHAT: A conversational helper for automotive tasks. In *Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP)*, pages 1061–1064, Pittsburgh, PA, September.

