OFFICE MESSAGE CENTER - A SPOKEN DIALOGUE SYSTEM

Jiang Han, Yonghong Yan, Zhiwei Lin, Yong Wang, Jian Liu, Danjun Liu, Zhihui Wang

Intel China Research Centre,
Intel China Ltd.
6th Floor, North Office Tower, #06-01 Beijing Kerry Centre
1, GuangHua Road, ChaoYang District,
Beijing 100020, PRC
Email: yonghong.yan@intel.com
Tel: (8610)85298800, Fax: (8610)85298717

Abstract
This paper describes the experience gained from the structuring of a spoken dialogue system and its key components during the design and development of a telephony based office message center, it integrates auto-attendant, email accessing, meeting scheduling capabilities through spoken dialogue interface. A building block dialogue toolkit has been designed based on these experiences and efficiently applied to the office message center with the capabilities of automatic call routine, meeting scheduling and email access. This paper also introduces some of the key components constructed for the spoken dialogue system: ASR engine and robust semantic parser, dialogue management module with consideration of domain portability, configurable template based sentence generation, and text to speech strategy.

Keywords: spoken dialogue, telephony, message center, building blocks.

1. Introduction
Spoken dialogue system is a voice enabled human computer interface. Enhanced with computer telephony interface and domain application back-ends, it can be used to provide complete messaging and answering solutions for your home and office. This paper introduce an office message center, which based on a building block dialogue technologies, provides mobile information access and control such as automatic call routing, email accessing and meeting scheduling by telephones.

![Diagram of an office message center](image-url)

Fig. 1: Diagram of an office message center
2. System Overview
Above diagram shows an example of the office message center. The dialogue control center plays important roles in receiving the parsed input, interacting with different application agents and conveying the message out through output generator. The distributed architecture and universal interface will facilitate the incremental creation of new application agents for information access and service providing. This experimental office message center is built as a test bed for the design of a dialogue toolkit for research and development.

2.1. Dialogue Toolkit Design
Designing a spoken dialogue system is typically a very time-consuming process, it involves many efforts from speech recognition, language understanding, discourse analysis, dialogue management, domain applications and user interface development. Especially at the initial stage, collecting conversational dialogue data, summing up spoken language grammar rules and conversational dialogue control rules for the application domain usually require a lot of hand crafted works. The building block architecture is introduced that hierarchically partitions the entire system into different modules and components by their different functions and capabilities. The design will facilitate the components sharing and inheritance of generic objects for dialogue management, hence speed up the process for designing a dialogue system of a new task.

There are two levels of interface for the build block dialogue design. At lower level it provides a C++ class interface for the key components of a typical dialogue system. For example, a frame-based, self-organizing slot filling class and a finite state machine (FSM) based dialogue control class are carefully designed for the dialogue management (DM) and flow control. Also, an enhanced ASR API is available to construct speech recognition engine, grammar specification and dialogic telephony interface. At higher level it provides useful engines and tools for rapid system deployment, such as generic ASR engines, scripting language supported authoring tools for dialogue control design, debugging and testing tools for development.

2.2. Experiments
An automatic call routine - Auto Attendant system is the most simple and efficient feature in the office message center that deployed in Intel Beijing office. Within the office, by simply dialing 1788 and speak the name of the person you are looking for, the system will recognize your speech and automatically make the phone connection for you. By testing upon the real world data collected during the Auto Attendant trial period, the system reaches successful connection accuracy higher than 90%.

A meeting scheduling system is another useful feature integrated in the office message center, besides of the basic components of a dialogue system, there is a backend application server named conference facility agent that communicates with an outlook email server and a conference room reservation server. The system is able to submit a meeting schedule through email to the required attendees based on the user's requests, it will automatically check the attendee's calendar to find the free time slot, and the availability of the conference room from the room reservation server. Based on the building block structure of the dialogue toolkit and the opened architecture of the office message center, more application features such as weather information, flight information, stock information query, telephony email accessing etc. can be added into the system incrementally to meet the increasing requirement for business.

3. Speech Recognizer and Robust Parser
A FSG based graph decoder that enhanced by semantic frame based meaning representations is introduced for the speech recognition and meaning extraction. This simple configuration is very effective for the smaller tasks that limited to a few hundred words of vocabulary size, but it becomes inefficient quickly for medium to large tasks when vocabulary size increases and grammar becomes complicated. In later case the system will slow down rapidly and consume more and more memory resource. A generic tree decoder based on the domain specific N-gram statistical language model is introduced for its efficiency and capability in handling large vocabulary speech recognition tasks. Also, a robust semantic parser could be added in order to extract the semantic frame based meaning representations so that it can pass the meaning to dialogue control center to make the decision for next step action.

4. Dialogue Management and Response Generation
The dialogue management for meeting scheduling task is based on form-filling paradigm. A hierarchical structure that contains message slots at various levels is introduced to describe task domain specifications. This is a semantic data driven dialogue manager in which the current interactive context is used to decide what to do in next step. The dialogue engine operates on the input semantic representation and the current interactive context to control the whole dialogue flow. Form-filling is an efficient and directive way for dialogue management. The key for design and implementation is to make the configuration simple and illustrative. A template based response sentence generation module is tightly coupled with the dialogue management module for easier configuration. A lot of effort has been put on making the domain dependent task specifications configurable for the dialogue toolkit.
5. **TTS for Limited Domain**
For limited domain applications, a simple implementation for Text-To-Speech can be done by waveform concatenation with large chunks of recording speech segments. The playback of concatenated sentence should be satisfied as long as all the necessary speech chunks have been recorded and correctly segmented.

6. **Discussion**
Although there are still a lot of research works need to be improved for spoken dialogue system in areas of robust speech recognition, spoken language understanding, discourse and dialogue modeling. The design and implementation of spoken dialogue applications such as the office message center show that it is possible to prototype practical and useful systems based on the prevailing technologies. The challenge remains on how could we continue to reduce the cost and shorten the developing period for building a usable dialogue system for a new domain.

7. **Reference**