

RavenCalendar: A Multimodal Dialog System for Managing a Personal Calendar*

Svetlana Stenchikova and Basia Mucha and Sarah Hoffman and Amanda Stent

Department of Computer Science

Stony Brook University

Stony Brook, NY 11794-4400

sveta,basia,shoffman,stent@cs.sunysb.edu

1 Introduction

Dialog applications for managing calendars have been developed for every generation of dialog systems research (Heidorn, 1978; Yankelovich, 1994; Constantinides and others, 1998; Horvitz and Paek, 2000; Vo and Wood, 1996; Huang and others, 2001). Today, Web-based calendar applications are widely used. A spoken dialog interface to a Web-based calendar application permits convenient use of the system on a hand-held device or over the telephone.

In this demo, we present *RavenCalendar*, a multimodal dialog system built around the Google Calendar and Google Maps Web applications. *RavenCalendar* allows the user to create, modify and remove calendar events, query for events, and hear descriptions of events. In our demonstration we will focus on two aspects of *RavenCalendar*: its flexible approach to language understanding and dialog management, and its multimodal interface.

Flexible dialog management In *RavenCalendar*, during event management or event querying a user can provide event information either in a single utterance (e.g. “Add a meeting for tomorrow from two to three in the computer science lounge” or by engaging in a subdialog with the system (Table 1). This functionality is made possible because we use the Ravenclaw dialog manager (Bohus and Rudnicky, 2003).

We would like to thank Dan Bohus, Antoine Raux, and Thomas Harris for providing Olympus, and for answering many questions about Ravenclaw. This research is based on work supported by the National Science Foundation under grant no. 0325188.

S: Welcome to RavenCalendar, the automated dialog calendar system.
S: What can I do for you today?
U: add a meeting for Monday
S: For what time do you want to add the meeting?
U: from ten until eleven
S: The event was added type meeting for tomorrow starting at ten a.m.
U: List all events for the next week

Table 1: Sample dialog with *RavenCalendar*

Multimodality A *RavenCalendar* user may interact with the calendar directly using the Google Calendar interface, or may interact through *RavenCalendar* using text, speech, map gestures or a combination of these media. A user may use the Google Maps interface to specify the location of an event; the system uses Google Maps to display the locations of events.

2 System Description

RavenCalendar, whose architecture is shown in Figure 1, is developed using Ravenclaw and Olympus (Bohus and others, 2007). Olympus is a dialog system shell; Ravenclaw is the Olympus dialog manager. In developing *RavenCalendar*, we chose to use an existing dialog shell to save time on system development. (We are gradually replacing the Olympus components for speech recognition, generation and TTS.) *RavenCalendar* is one of the first dialog systems based on Olympus to be developed outside of CMU. Other Olympus-based systems developed at CMU include the Let’s Go (Raux and others, 2005), Room Line, and LARRI (Bohus and Rudnicky, 2002) systems.

Flexible dialog management The Ravenclaw dialog manager (Bohus and Rudnicky, 2003) allows “object-oriented” specification of a

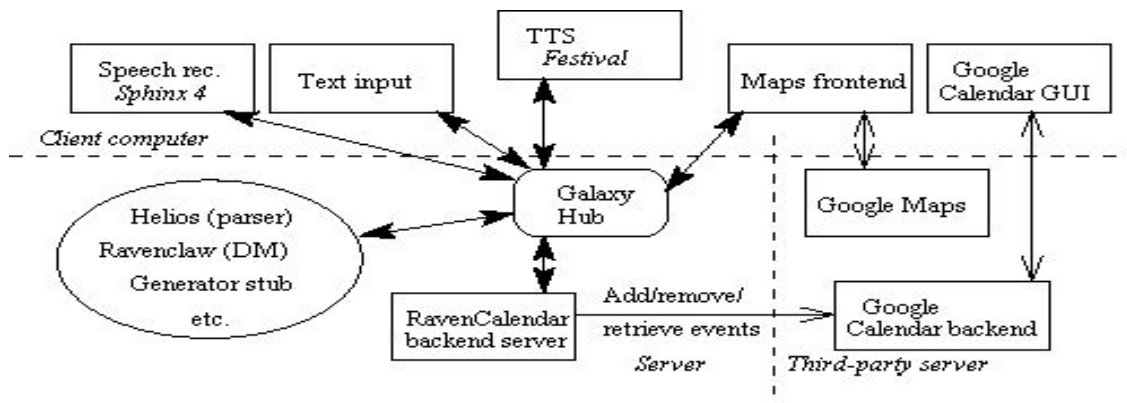


Figure 1: RavenCalendar Design

dialog structure. In *RavenCalendar*, we define the dialog as a graph. Each node in the graph is a minimal dialog component that performs a specific action and has pre- and post-conditions. The dialog flow is determined by edges between nodes. With this structure, we maximize the reuse of minimal dialog components. Ravenclaw gives a natural way to define a dialog, but fine-tuning the dialog manager was the most challenging part of system development.

Multimodality In *RavenCalendar*, a back-end server integrates with Google Calendar for storing event data. Also, a *maps front end* server integrates with Google Maps. In addition to the locations recognized by Google Maps, an XML file with pre-selected location-name mappings helps the user specify locations.

3 Current and Future Work

We are currently modifying *RavenCalendar* to use grammar-based speech recognition for tighter integration of speech recognition and parsing, to automatically modify its parsing grammar to accommodate the words in the user's calendar, to permit trainable, adaptable response generation, and to connect to additional Web services and Web-based data resources. This last topic is particularly interesting to us. *RavenCalendar* already uses several Web-based applications, but there are many other Web services of potential utility to mobile users. We are now building a component for RavenClaw that searches a list of URLs for event types of interest to the user (e.g. sports events, music events), and automatically notifies

the user of events of interest. In the future, we plan to incorporate additional Web-based functionality, with the ultimate goal of creating a general-purpose dialog interface to Web applications and services.

References

- D. Bohus et al. 2007. Olympus: an open-source framework for conversational spoken language interface research. In *Proceedings of the Workshop "Bridging the Gap" at HLT/NAACL 2007*.
- D. Bohus and A. Rudnicky. 2002. LARRI: A language-based maintenance and repair assistant. In *Proceedings of IDS*.
- D. Bohus and A. Rudnicky. 2003. Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Proceedings of Eurospeech*.
- P. Constantinides et al. 1998. A schema based approach to dialog control. In *Proceedings of ICSLP*.
- G. Heidorn. 1978. Natural language dialogue for managing an on-line calendar. In *Proceedings of ACM/CSCER*.
- E. Horvitz and T. Paek. 2000. DeepListener: Harnessing expected utility to guide clarification dialog in spoken language systems. In *Proceedings of ICSLP*.
- X. Huang et al. 2001. MIPAD: A next generation PDA prototype. In *Proceedings of ICSLP*.
- A. Raux et al. 2005. Let's go public! Taking a spoken dialog system to the real world. In *Proceedings of Interspeech*.
- M. Tue Vo and C. Wood. 1996. Building an application framework for speech and pen input integration in multimodal learning interfaces. In *Proceedings of ICASSP*.
- N. Yankelovich. 1994. Talking vs taking: Speech access to remote computers. In *Proceedings of the Conference on Human Factors in Computing Systems*.