Designing a Multimodal Phone Interface for Car

Martin Čmejrek, Martin Fanta, Ladislav Seredi, Vladimír Bergl
IBM Czech Republic, Voice Technologies and Systems
The Park, V Parku 2294/4, 148 00 Prague 4, Czech Republic
{martin.cmejrek, martin.fanta, ladislav_seredi, vladimir_bergl}@cz.ibm.com

Abstract

This paper describes the functional prototype of CarDialer - a multimodal application for controlling a cellphone while driving a car - designed and implemented by the authors.

The user interface of CarDialer integrates two modalities: voice (including speech recognition, speech synthesis and pre-recorded prompts) and GUI (touch sensitive LCD panel in dashboard and buttons mounted on a steering wheel). The system facilitates common tasks associated with a mobile phone as name and number dialing, adding new entries to the contact list, operations with lists of calls and messages, notification of presence, etc. The application is fully functional as soon as the user connects its mobile phone to the system. The UI is built on and open framework independent of the application logic. It includes generic modules as dialog manager, multimodal render kit as well as multimodal event fusion and other components.

The architecture of CarDialer allows both symmetric and complementary use of those modalities. the voice modality is the primary one, the user interface is designed so that the user can complete any task only by using voice, without looking at the screen. The main function of GUI is to present auxiliary information, which may speed-up user interaction. Nevertheless, almost all actions can be performed using just GUI, and there are certain actions (e.g. confirmation vs. rejection), where using GUI is faster and less intrusive than voice interaction.

The system allows reasonable permutations of names, using initials, or elisions, such as *Brown James*, *J. Brown*, *James B.*, or *James*.

Some underspecified utterances may result in ambiguities in case of more contacts with the same name or surname. In order to quickly explain the ambiguity to the user, the prompt summarizes, what are the common properties of the ambiguous contacts and what are the distinguishing features, such as "H: *Dial Smith.* → C: *I have two contacts with name Smith: John and Kevin, say one of them.*"

Another problem is related to acoustically similar names, since repeating the whole name would not do any progress. In addition to a multimodal strategy using the buttons for making a selection, the voice interaction offers an alternative solution: "H: *Dial John Thorn* → C: *I have 2 contacts that sound similar, please do not repeat the name, rather say 'Dial the first one' or 'Dial the second one'.*"

From the architectural point of view the system considers the following modalities: speech, GUI, and – since the drive may use it independently and the system has to monitor and control it – the phone.

Render kits are responsible for translating platform-independent commands (specific to a given modality) into an implementation-specific protocol. The speech render kit, for example, translates the definition of the TTS prompt, the list of ASR grammars, the ASR confidence score, and other speech-specific data into VXML. Similarly, the GUI render kit translates GUI-related commands into a set of implementation specific (in our case either Java SWT or Adobe Flash) ones. Events issued by modalities are transformed inversely into *generic modality events*.

The *dialog manager* interprets the *dialog state machine* – the computational form of the call flow. Each dialog state is associated with an implementing *dialog component*. The dialog manager traverses the state machine and executes component's code associated with entering and exiting the call flow states. This activity results in preparing platform-independent commands, such as prompts to be played, grammars to be recognized, or GUI content to be displayed. Handling of generic modality events results in transitioning between dialog states.

References

[1] Statement of principles, criteria and verification procedures on driver interactions with advanced in-vehicle information and communication systems. Technical report, Washington, D.C., 2002.

[2] T. Beran, V. Bergl, R. Hampl, P. Krbec, J. Šedivý, B. Tydlitát, and J. Vopička. Embedded viavoice. In *Proceedings of TSD 2004*, Brno, 2004.

[3] V. Fischer, C. Gunther, J. Ivanecký, S. Kunzmann, J. Šedivý, and L. Ureš. Multi-modal interface for embedded devices. In *Elektronische Sprachsignalverarbeitung*, Dresden, 2002.

[4] P. Green, W. Levison, G. Paelke, and C. Serafin. Preliminary human factors guidelines for driver information systems (technical report umtri-93-21). Technical report, Ann Arbor, MI, 1993.

[5] J. Kleindienst, T. Macek, L. Seredi, and J. Šedivý. Vision-enhanced multi-modal interactions in domotic environments. In *Proceedings of the 8th ERCIM Workshop "User Interfaces For All"*, 2004.