Chapter 22

ROBUST MULTIMODAL DIALOG MANAGEMENT FOR MOBILE ENVIRONMENTS

Jeonwoo Ko¹, Fumihiko Murase¹, Teruko Mitamura¹, Eric Nyberg¹, Nobuo Hataoka², Hirohiko Sagawa², Yasunari Obuchi³, Masahiko Tateishi⁴ and Ichiro Akahori⁴

¹Language Technologies Institute, Carnegie Mellon University, USA
²Central Research Laboratory, ³Advanced Research Laboratory, Hitachi Ltd., Japan
⁴Research Laboratories, DENSO CORPORATION, Japan

Abstract: This chapter describes three aspects of mobile dialog management: robustness in the presence of recognition errors; dynamic behavior based on user context (e.g. network connectivity, location); and efficient scenario description for multimodal dialogs. We describe algorithmic techniques for these three aspects of mobile dialog management, and results from empirical user studies are discussed which indicate significant improvement in performance and user satisfaction when these techniques are deployed in a dialog system.

Keywords: Dialog system; mobile environment; dialog management; error handling; context-awareness; multimodal interaction

1. INTRODUCTION

In the past few years, several important studies have appeared on mobile dialog systems in pedestrian and automotive environments (Pellom et al., 2001; Buhler et al., 2002; Minker et al., 2004). Such systems include a navigation interface (to provide route guidance) and remote data access (to provide information on points of interest, tourism sites, weather, restaurant and entertainment facilities, etc.). In mobile environments, dialog systems must provide robust dialog management in order to handle recognition errors and network loss, and to support robust multimodal interactions. Our recent
work has focused on three aspects of robust dialog management in mobile environments.

First, speech recognition errors are more frequent in mobile settings due to noise in the environment, which can make it difficult to support smooth speech dialog communication. To address this problem, we propose a grammar-based error handling approach that dynamically generates correction grammars based on the dialog history, and uses these correction grammars to detect and repair recognition errors. Experimental results show that our error handling technique increased task completion rate while reducing the number of user turns.

Second, network connectivity is not stable in a mobile environment (e.g. the user goes through a tunnel or climbs up a mountain) and the dialog manager should be robust enough to handle changes in network status. When an information request is interrupted by a loss of network connectivity, the user should have to request the same information again when network connectivity is reestablished. The system should maintain a dialog history and automatically restart any pending dialogs as soon as the network becomes available, so that the user does not need to monitor the network status and then ask for information again. Such behavior is an example of a context-aware service (Dey and Abowd, 2000), which recognizes contextual events (e.g. a change in network status) and responds by providing relevant information to the user. We designed three types of context for mobile dialogs, and conducted an evaluation of their use in both laboratory and driving environments. Evaluation results show that context-aware dialog management improved user performance and user satisfaction.

Third, mobile dialog systems involve multimodal interactions using both speech and display/touch screen. This implies that dialog scenarios must contain detailed specifications about what to present in each modality (e.g., what to display, what to say, etc.) for each dialog step. In our previous work (Nyberg et al., 2002), we developed ScenarioXML, an extension to VoiceXML which supports flexible dialog switching. We recently extended ScenarioXML to support multimodal interactions, so that dialog developers can easily define a new multimodal dialog scenario for a new domain.

The remainder of this chapter is structured as follows. Section 2 introduces the CAMMIA dialog system, which is framework for our dialog research. Section 3 explains how correction grammars are created based on the dialog history, and how correction grammars detect and repair system errors. In Section 4, we describe context-aware features in mobile dialog systems and discuss the related issue of handling user interruption. Section 5 describes the extended ScenarioXML for multimodal dialog management. Valuable conclusions are drawn in Section 6.