

Adaptive Interaction Methods in Speech User Interfaces

Markku Turunen

Human-Computer Interaction Group

University of Tampere

Department of Computer and Information Sciences

FIN-33014 University of Tampere, Finland

mturunen@cs.uta.fi

ABSTRACT

The aim of this research is to build a model for adaptive interaction management in speech user interfaces. We are constructing a framework to aid the development of adaptive speech applications in general, and multilingual speech application in particular. We use *input agents*, *dialogue agents* and *presentation agents* to achieve our goals. In addition to the core architecture we are interested in error-handling methods and interaction in complex, structural elements. Our current work has produced an application development framework called Jaspis. We have also built a multilingual e-mail client, Mailman, based on the Jaspis architecture.

Keywords

Speech recognition, dialogue management, error handling, input agents, dialogue agents, adaptive applications

INTRODUCTION

Currently speech application development is still hard work with a lot of low-level coding. Although many development tools have been introduced, they are not ideal for the development of adaptive, such as multilingual, applications. Furthermore, most tools do not support high-level interaction methods, such as dialogue handling and error correction. In general, more interaction methodology should be included in the development tools.

When adaptive applications are constructed we need adaptive *dialogue handling*, *input handling* and *output generation* together with an adaptive *information and interaction model*. I am working to support these in our speech application development architecture called Jaspis. In this paper I focus on input handling and dialogue management components. I will describe the principles behind these components and how they are, and will be, realized in Jaspis and applications using Jaspis.

The rest of the paper is organized as follows. First, I introduce the general architecture of Jaspis. Then I explain how input handling and dialogue management work. After

that I present some application issues. Finally, conclusions and future work are presented.

GENERAL ARCHITECTURE

The general architecture of Jaspis is shown in Figure 1. The main components are *interaction management*, *information management*, *presentation management*, *dialogue management* and *communication management*.

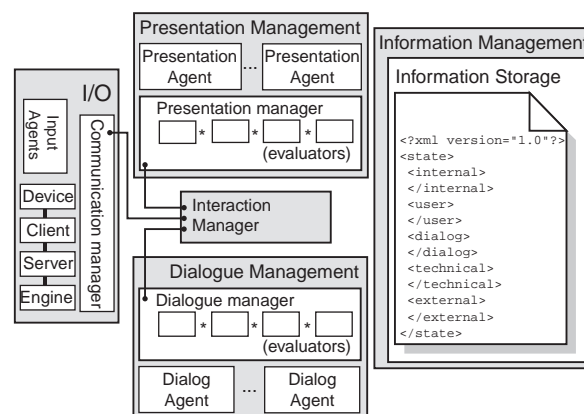


Figure 1: General Jaspis architecture.

The Jaspis architecture is based on distributed components. The components are connected to the *interaction manager* which controls interaction on the highest level. All components, except information management, are in a symmetrical position and decide for themselves when they are able to take over the control of the application. This architecture allows flexible interaction management but still makes it possible to retain close connections between the components.

Jaspis follows the principle of stateless components. All information is stored in *information storage* instead of the individual components. Every component has access to all the information the system contains. This makes it possible for components, such as input handlers, to adapt to the user and the situation based on the shared information found in the information storage. Information is stored in a form that is as conceptual level as possible.

The information storage and interaction management components are crucial to the adaptive features of the

other components. These, and the Jaspis architecture in general are described in more detail in [2]. The presentation management is described in [1].

ADAPTIVE INPUT HANDLING

We use *input agents* to handle the raw inputs between the user and the dialogue management. Input agents interpret the inputs into a conceptual form understood by the other components of the system. Input agents are specialized in certain inputs and situations. This way we can have modular and reusable interaction components, and it is also an efficient way to support adaptive applications. For example, in a multilingual application, support for new languages can be built by inheriting interpretation agents and writing only the language specific methods.

Input agents are not only meant for the interpretation of inputs. They also handle other interaction tasks, such as the detection of errors. Since they combine high-level processing of inputs with low-level device integration they can handle complex inputs and still coordinate devices in a timely manner. Such support is often missing from speech application toolkits.

In addition to input agents, we use *engines*, *servers*, *clients* and *devices* to provide a distributed environment in which input agents can operate efficiently. Input agents coordinate devices and handle multimodal inputs by using multiple input streams and by providing feedback to the devices.

ADAPTIVE DIALOGUE HANDLING

We use a dialogue model in which a monolithic dialogue structure is abandoned. Instead, a dynamic, context-sensitive and adaptive alternative is proposed. This model distributes the single dialogue flow to the multiple dialogue agents. In every dialogue turn the most suitable agent is selected to handle the current dialogue turn.

The *dialogue manager* is a component that attempts to choose the best possible dialogue agent for the situation at hand. It evaluates all available dialogue agents and chooses the most suitable one. The dialogue manager uses *dialogue evaluators* to compare the different attributes of dialogue agents and chooses the one recommended by the evaluators.

The selection of the most suitable agent is based on the overall situation which contains current inputs, dialogue context and user preferences. All information stored in the information storage can be used in the evaluation process. Therefore, the selection procedure supports specialized dialogue agents, such as the ones designed for error handling. It also supports alternative dialogue flows and adaptive dialogue strategies. Alternative and adaptive dialogue handling strategies are needed not only in multilingual applications, but also in monolingual applications, as different users prefer different dialogue handling strategies [4].

APPLICATION ISSUES

We have constructed a multilingual e-mail client, Mailman [3], to try out these interaction models in practice. We found that in the e-mail domain there are two particularly problematic issues: multilinguality and complex, structural elements, such as tables. Currently we have studied mainly output generation issues, such as how prosody could be utilized when structural elements are read to the user. We will continue to investigate these questions from the perspective of input and dialogues using input and dialogue agents.

Further applications under development address interaction issues, such as the needs of ubiquitous computing and the demands of public information services. Since the proper handling of recognition errors is a crucial issue in speech applications, we investigate how sophisticated error-handling methods could be constructed using input and dialogue agents.

CONCLUSIONS AND FUTURE WORK

We are developing a model for adaptive interaction management for speech user interfaces. This work has resulted in the first version of an adaptive speech application development environment called Jaspis. Building on Jaspis, we have implemented a multilingual e-mail client Mailman (Postimies in Finnish) and more applications are under development. In the future we will focus further on adaptive input and dialogue handling issues and especially on multilinguality, error handling and interaction in structural elements.

ACKNOWLEDGMENTS

This work was supported by the Academy of Finland (project 163356) and by the National Technology Agency (Tekes). I also like to thank Jaakko Hakulinen for his valuable comments and ideas concerning this work.

REFERENCES

1. Hakulinen, J. and Turunen, M. Presentation Agents for Speech User Interfaces. In *CHI 2000 Extended Abstracts*, ACM Press, 2000, 115-116.
2. Turunen, M. and Hakulinen, J. Jaspis - A Framework for Multilingual Adaptive Speech Applications. In *Proceedings of 6th International Conference of Spoken Language Processing*, 2000 (to appear).
3. Turunen, M. and Hakulinen, J. Mailman - a Multilingual Speech-only E-mail Client Based on an Adaptive Speech Application Framework. In *Proceedings of Workshop on Multi-Lingual Speech Communication*, 2000 (to appear).
4. Walker, M., Fromer, J., Fabbriozio, G., Mestel, C., and Hindle, D. "What can I say?: Evaluating a spoken language interface to Email". In *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, 1998: 582-589.