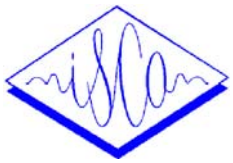
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2004 - ICSLP
8th International Conference on Spoken Language Processing
ICC Jeju, Jeju Island, Korea
October 4-8, 2004

Robust and Adaptive Architecture for Multilingual Spoken Dialogue Systems

Markku Turunen, Esa-Pekka Salonen, Mikko Hartikainen and Jaakko Hakulinen

Department of Computer Sciences, Tampere Unit for Computer Human Interaction
Speech based and Pervasive Interaction Group
University of Tampere, Finland
{mturunen,eps,mhartikainen,jh}@cs.uta.fi

10.21437/Interspeech.2004-773

Abstract

We present how robustness and adaptivity can be supported by the spoken dialogue system architecture. AthosMail is a multilingual spoken dialogue system for e-mail domain. It is being developed in the EU-funded DUMAS project. It has flexible system architecture supporting multiple components for input interpretation, dialogue management and output generation. In addition to language differences, these components have great variation in their approaches for spoken interaction. For example, the dialogue management components implement different dialogue control models. The system architecture makes the interaction robust and adaptive by utilizing different approaches for spoken interaction in a single application by selecting suitable components at runtime.

1. Introduction

There are many challenges related to the development of practical speech applications. Three common ones are the need for multilinguality, adaptive interaction and robustness. In this paper we present system architecture and an application for e-mail domain, which deals with these issues.

Robustness is one of the key issues in current spoken dialogue systems. In speech interaction the characteristics of speech, such as its serial nature, bring limitations. These limitations should be compensated using robust and adaptive techniques which take different users into account. Moreover, technical problems can be solved by using robust interfaces which allow successful interaction even when the underlying technologies fail.

Adaptivity may refer to various aspects in speech applications. For example, adaptive techniques can help different users by utilizing customized interaction methods and techniques. In speech-based human-computer interaction users have diverse ways of communication. Novice users and experienced users may want the interface to behave completely differently, for example to be system-initiative instead of mixed-initiative. An example of the benefits of interaction level adaptivity is reported by Litman and Pan [5].

Multilingual applications are examples of adaptive applications. When the need to multiple languages arises from the application domain (e.g., speech-based e-mail systems), handling multiple languages is essential. Most importantly, there should be a coordinated way to incorporate both language independent and dependent components. This should be implemented in as modular, automatic and transparent a fashion as possible.

When robustness, adaptivity and multilinguality are considered from the interaction level viewpoint, the handling of speech inputs and outputs and dialogue management are the most important areas. Examples from AthosMail include context-sensitive and personalized recognition grammars with customized vocabularies, robust input handling that combines complete linguistic analysis with concept-spotting method, the use of different dialogue control models, context-sensitive help and guidance and adaptive generation of multilingual outputs.

In this paper we present how the challenges mentioned are addressed in the multilingual AthosMail application and supported by the system architecture. We explain how adaptivity can take place on all system levels, and how the system supports multiple components for same purposes, thus allowing robust integration of different approaches. First, we introduce the AthosMail application. After that, the underlying system architecture is presented. Next, the main components of the system are described. Finally, conclusions are presented.

2. AthosMail Application

AthosMail is a multilingual (Finnish, Swedish and English) speech-based e-mail application. It allows the user to access his/her mailbox using a standard mobile or fixed-line telephone. It provides the most common e-mail client functions.

AthosMail is based on the existing Mailman application [12]. In the EU-funded DUMAS (Dynamic Universal Mobility for Adaptive Speech Interfaces) project new and elaborated components have been produced for text-processing, dialogue management, semantic template construction, user modeling, integrated tutoring, and random indexing [13].

Dialogues in the e-mail domain are rather open, and especially there are no clearly defined tasks like in many other application areas (e.g. in timetable systems). Therefore, a collaborative user initiated dialogue strategy is applied. The system supports the user by providing context-sensitive help and guidance. For example, AthosMail uses adaptive prompts, integrated tutoring and universal commands to help the user to know what to do. The system monitors the user and adjusts the amount of guidance. Still, the dialogue strategy is based on user-initiative approach, and experienced users can perform operations efficiently.

Next, the main components of the AthosMail application and the main principles of the underlying system architecture are presented.

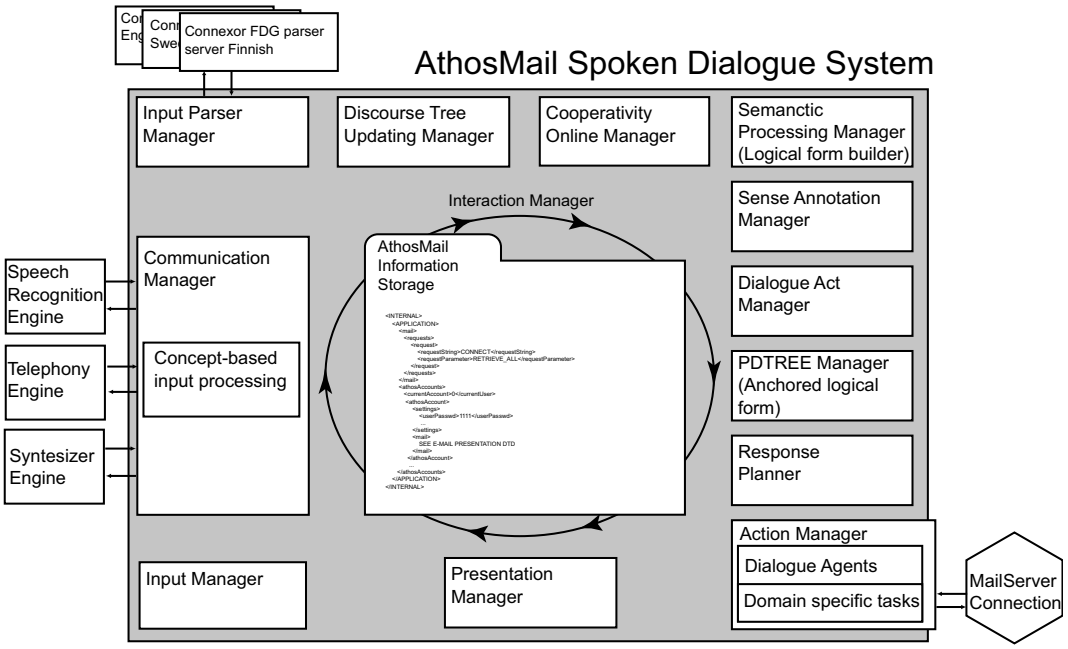


Figure 1: AthosMail system structure.

3. System Architecture

Figure 1 illustrates the main modules (collections of agents) of the AthosMail spoken dialogue system. There is no central module in the system. Instead, agents form modules corresponding to high-level tasks in spoken dialogue applications. Most notably, natural language understanding, dialogue management and output generation tasks are distributed not only to several modules, but to numerous compact agents over different parts of the system. AthosMail contains more than ten modules, each containing various number of agents to handle different tasks. In total, it contains more than hundred specialized agents.

Next, the principles of the underlying Jaspis architecture are presented.

3.1. Underlying Jaspis Architecture

The AthosMail application is constructed on top of the Jaspis architecture [11]. The Jaspis architecture supports highly distributed but coordinated components, shared system knowledge and system-level adaptation. The system consists of several managers that are under central coordination. Each of the managers includes various amounts of agents to handle different tasks, and there are also different agents that handle the same task in different ways. Evaluators are used to choose between different agents. The system architecture is distributed so that different managers and agents can run on different computers and platforms.

Jaspis-based applications store all their information in the shared Information Storage. In AthosMail it contains the discourse tree, which holds all dialogue related information, including user inputs, dialogue data and system outputs. In this way, all information is shared between various system agents and evaluators. The agent – evaluator – manager paradigm and the shared information management play major roles in the system adaptation, as presented next.

3.2. System-level Adaptation

The Jaspis architecture contains a general adaptation mechanism that can be used across system modules and applications. In a nutshell, each manager uses a set of evaluators to select the most suitable agent for each situation. Each evaluator gives a floating point score between zero and one for each agent. The agent with the highest overall score is selected to handle the current task.

The general system level adaptation mechanism is applied to various tasks. For example, the presentation agents (used for response generation) are evaluated with five different evaluators. Each evaluator use specific information to give a score for each agent. The presentation evaluators use the following information: (i) dialogue data (e.g. dialogue act), (ii) language of the output (e.g. Finnish, Swedish or English), (iii) user expertise, (iv) characteristics of the mailbox, and (v) self-evaluation results of the agents. Evaluation scores from the five evaluators are combined and the agent with the highest overall score is selected to generate the output. A similar adaptation mechanism is applied to all system components.

It is noteworthy that there is no single evaluator, nor any single component in general, which selects agents for each situation, but instead the selection is always both dynamic and distributed. This makes it possible to keep the program control dynamic and adaptive at the architectural level. Furthermore, new features can be added to applications without modifications to existing components. For example, most adaptive features of AthosMail, such as guidance prompts and interactive tutoring, are included to the system in this way.

Next we present how these general principles are realized in the AthosMail application. Here we focus on interaction management tasks.

4. Interaction Management

As illustrated in Figure 1, interaction management tasks (natural language understanding, dialogue management and response generation) are distributed to various modules in the AthosMail application. Most importantly, the application contains multiple approaches for each of these high-level tasks. In addition, several "add-ons" are included to the system to bring additional robustness and adaptivity.

4.1. Natural Language Understanding

Natural language understanding is performed in two alternative ways in AthosMail. In the first approach natural language understanding is based on the complete linguistic analysis of the user input. The linguistic analysis agents build a logical form out of the user input. In the second approach a concept-based keyword recognition method is used. The concept-spotting agents identify predefined and dynamically constructed concepts from the recognized speech and DTMF inputs, and resolve their relations. In the AthosMail application these two approaches work together so that the user input is given to both set of agents to be analyzed and the results are shared for other components to use.

Concept spotting is performed by a set of agents and evaluators, which implement the concept model on the base of concept definitions. The concepts are kept in the Information Storage, and they can be updated dynamically. The predefined concepts match the system functionality. The dynamic concepts are related to the information specific for the domain and they are adapted for each individual user. In the case of AthosMail the dynamic concepts are based on the information found from user's mailbox (e.g. sender names). Using compact agents we are able to make NLU components very specialized and distribute their workload. For example, in AthosMail highly specialized concept-spotting agents take care of most of the meta-communication and application-domain related inputs, such as sentences related to error situations, navigation inside the e-mail messages and interpretation of DTMF inputs.

The two approaches presented have different benefits. With a complete linguistic analysis we are able to deal with rich speech inputs, while the concept-spotting approach can handle incomplete and even controversy recognition results (for example, when multiple word-spotting results are returned by the recognizer). With the concept agents it is usually possible to extract the meaning behind the user utterance even if the recognition result is not grammatically correct. For example, if the user says "Mark maybe uh let's see seventh message" and recognition result is "Mark do I have quit seventh message", the AthosMail system is able to recognize the correct user intention or at least to present a clarification dialogue. In this way the concept-spotting approach brings robustness to the input processing.

Agent-based approach for NLU supports multilingual applications. There can be multiple sets of agents for each used input language depending on the available resources and used technology. For example, in the AthosMail application language specific features, such as non-trivial inflection of Finnish names are handled by a set of specialized concept agents.

Distributed agents support also dynamic use of multiple languages. In the AthosMail application it would be fully possible to handle various input languages inside a single dialogue. In the e-mail domain, however, this is not meaningful, since users are identified and their preferred language is known. For other domains, such as information kiosks and pervasive computing applications, this is a very useful feature.

4.2. Dialogue Management

Similar to the NLU, dialogue management is performed using two approaches. The linguistic oriented approach uses agents to produce an abstract plan about the next system dialogue move, regarding to the anchored logical form. These components are able to deal with complex discourse structures.

The second approach utilizes numerous specialized dialogue agents. These agents operate on the base of conceptualized user inputs and the internal state of the system. Each agent is related to one of the high-level functions of the AthosMail application, e.g., reading of e-mail messages, navigation inside the mailbox and giving context-sensitive help.

The specialized dialogue agents are implemented using the object-oriented approach. Most importantly, inheritance is used to separate generic dialogue management from domain specific actions. Similar approaches are used in other systems (e.g. [1], [7] and [8]). In AthosMail we have successfully re-used and inherited large number of agents from the existing Mailman application [12] and the underlying Jaspis architecture. For example, generic Jaspis agents are used to handle domain independent dialogue tasks (e.g., to accept incoming calls and handle certain error situations).

The modular approach to dialogue management makes it possible to combine the benefits of different dialogue control models. This is especially useful in sub-dialogues. For example, in the AthosMail application certain tasks, such as reading of e-mail messages and navigation in the mailbox are performed entirely by a set of specialized dialogue agents, which control the dialogue very differently from the linguistic oriented dialogue agents. Still, all of these agents use the Information Storage to maintain shared dialogue state.

Distributed dialogue management can be used to implement different dialogue management strategies to adapt the dialogue to the user. Different dialogue management strategies, such as the system-initiative approach and the mixed-initiative approach, have different benefits and drawbacks, which makes it useful to use both of them in an adaptive way [14]. Several techniques have been presented in previous research (e.g. [4] and [6]). In AthosMail a set of guidance agents is used to bring system-initiative features to the user-initiative interface. In addition, when there are problems (e.g. input analysis or other dialogue agents fail), a set of backup-agents are used to keep the interaction on track. Dynamic selection of these agents brings robustness and adaptivity to the interaction. The dynamic dialogue management approach is presented in detail in [10].

4.3. Response Generation

Like in other parts of the AthosMail system, response generation uses two approaches. In the first approach a single generator agent produces surface strings by concatenating together pre-defined utterance segments. Many of the basic output generation tasks can be handled using this approach.

In the second approach numerous presentation agents are used to handle different generation tasks. These agents are capable of producing outputs for specific situations in different ways. Most importantly, the language of the user, the structure of the mailbox and the interaction history are taken into account, as demonstrated in Section 3.2. The use of agents and evaluators is especially suitable for NLG tasks. This way, new components can be added without modifications to existing ones, resulting in outputs that are adaptive and easy to extend. This allows iterative development that is typical for practical speech applications.

Agent-based response generation is able to handle multiple output requests in each turn. The results of different agents are concatenated together, when appropriate, to form a single system response. Alternatively, multiple responses can be produced, for example when multiple synthesizers with different languages are used. Because of the technical limitations and cognitive capabilities of the users (e.g. it is not meaningful to change synthesizer for each individual word) the generation of multilingual outputs is far from trivial in the e-mail domain. For example, single e-mail message may contain various languages, and proper agent is needed for each element. In this way, rich combinations of multilingual outputs can be produced.

4.4. Context-sensitive Help and Guidance

To bring more adaptive features and robustness to the interaction AthosMail supports a set of additional features. One set of additions are universal commands. The 'Tell me more' command gives a detailed description of the current dialogue state. For example, when a message is selected, only the main information is spoken. With this command the user hears additional information, such as length of the message etc. Similarly, the 'What now?' command gives relevant guidance to the user, taking into account the current dialogue situation. Similar approach is used in [9].

Second addition to the system is tutoring agents [2], which give guidance to the user by introducing the system and monitoring how well the user interacts with it. Tutoring components take the initiative in certain situations. They utilize the dialogue history, the user model and their own tutoring plan in this process.

The features mentioned are implemented using specialized NLU, DM and NLG agents. Using the agent approach together with the system level adaptation mechanism we have been able to include these features to the AthosMail application without modifications for other components. In addition, these features can be turned dynamically on and off.

5. Conclusions

We have presented how robustness, adaptivity and challenges of multilingual applications are addressed in the AthosMail spoken dialogue system. Especially, we have demonstrated how the underlying system architecture supports these tasks with its general adaptation mechanism, and how components from multiple sources can be integrated to aid development of sophisticated spoken dialogue systems. The system has been evaluated with a subjective evaluation paradigm, and preliminary results from a user study are promising [3].

6. Acknowledgements

This research was carried out within the EU-funded project DUMAS (Dynamic Universal Mobility for Adaptive Speech Interfaces), IST-2000-29452.

7. References

[1] Bohus, D. and Rudnicky, A., "RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda", *EUROSPEECH 2003 Proc.*, Geneva 2003.

[2] Hakulinen, J., Turunen, M., and Salonen, E.-P, "Agents for Integrated Tutoring in Spoken Dialogue Systems", *EUROSPEECH 2003 Proc.*, Geneva 2003.

[3] Hartikainen, M., Salonen, E-P. and Turunen, M., "Subjective Evaluation of Spoken Dialogue Systems Using SERVQUAL Method", *ICSLP 2004 Proc.*, Jeju 2004.

[4] Larsen, L. B., "A Strategy for Mixed-Initiative Dialogue Control", *EUROSPEECH 1999 Proc.*, Budapest 1999

[5] Litman, D. and Pan, S., "Designing and Evaluating an Adaptive Spoken Dialogue System", *User Modeling and User-Adapted Interaction*, 12, 2/3, 2002.

[6] McTear, M., Allen, S., Clatworthy, L., Ellison, N., Lavelle, C. and McCaffery, H., "Integrating Flexibility into a Structured Dialogue Model: Some Design Considerations", *ICSLP 2000 Proc.*, Beijing 2000.

[7] O'Neill, I. M., McTear, M. F., "Object-Oriented Modelling of Spoken Language Dialogue Systems Natural Language Engineering", *Best Practice in Spoken Language Dialogue System Engineering, Special Issue*, 6, 3, 2000.

[8] Pakucs, B., "Towards Dynamic Multi-Domain Dialogue Processing", *EUROSPEECH 2003 Proc.*, Geneva 2003.

[9] Rosenfeld, R., Olsen, D., Rudnicky, A., "Universal speech interfaces", *ACM Interactions*, 8, 6, ACM 2001.

[10] Salonen, E-P, Hartikainen, M., Turunen, M., Hakulinen, J. and Funk, J. A., "Flexible Dialogue Management Using Distributed and Dynamic Dialogue Control", *ICSLP 2004 Proc.*, Jeju 2004.

[11] Turunen, M. and Hakulinen, J., "Jaspis - A Framework for Multilingual Adaptive Speech Applications", *ICSLP 2000 Proc.*, Beijing 2000.

[12] Turunen, M. and Hakulinen, J., "Mailman - a Multilingual Speech-only E-mail Client Based on an Adaptive Speech Application Framework", *Workshop on Multi-Lingual Speech Communication Proc.*, MSC 2000: 7-12, 2000.

[13] Turunen M., Salonen, E.-P., Hartikainen, M., Hakulinen, J., Black, W., Ramsay, A., Funk, A., Conroy, A., Thompson, P., Stairmand, M., Jokinen, K., Rissanen, J., Kanto, K., Kerminen, A., Gamback, B., Cheadle, M., Olsson, F., and Sahlgren, M., "AthosMail – a Multilingual Adaptive Spoken Dialogue System for E-mail Domain". *Workshop on Robust and Adaptive Information Processing for Mobile Speech Interfaces Proc.*, Geneva, 2004.

[14] Walker, M.A., Fromer, J., Di Fabbrizzio, G., Mestel, C., and Hindle, D., "What can I say?: evaluating a spoken language interface to Email", *SIGCHI Conference on Human Factors in Computing Systems Proc.*, 1998.