Robust and Adaptive Architecture for Multilingual Spoken Dialogue Systems

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

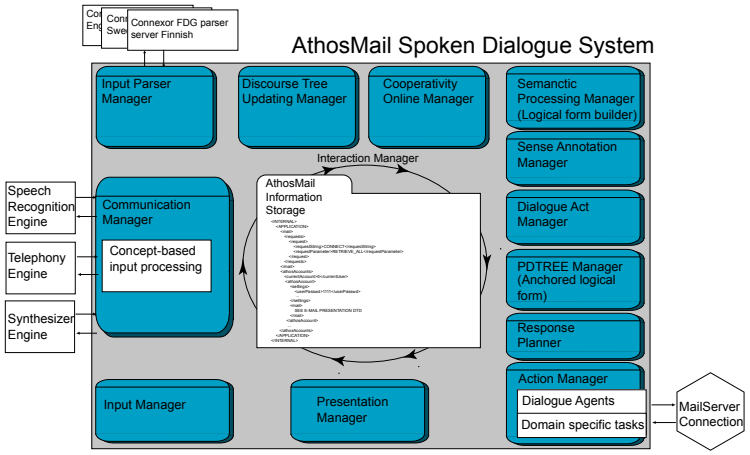
Speech-based and Pervasive Interaction Group - Tampere Unit for Human-Computer Interaction - University of Tampere

1. Introduction

We present how robustness, adaptivity and challenges of multilingual applications are addressed in the AthosMail spoken dialogue system. AthosMail has flexible system architecture supporting multiple alternative approaches for input interpretation, dialogue management and output generation. In addition, AthosMail contains several "add-ons" for help and guidance. Its general adaptation mechanism selects these components dynamically during the runtime. New components can be added without modifications to existing ones. This allows iterative development and integration of components from multiple sources. The system has been evaluated with a subjective evaluation paradigm [1].

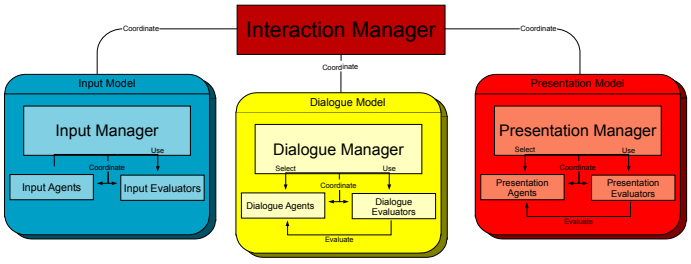
2. Architecture overview

The AthosMail application is constructed on top of the Jaspis architecture [2]. Jaspis supports highly distributed but coordinated components, shared system knowledge and system-level adaptation. The system consists of managers that each includes various amounts of agents. The agents handle different tasks, and there are also agents that handle the same tasks in different ways. All information is stored in the system knowledge base and shared between the system components. AthosMail contains more than one hundred specialized agents, many of them used in other applications as well.



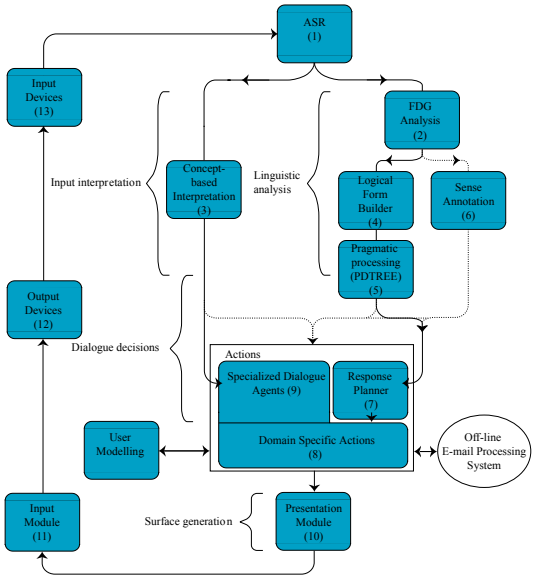
3. Adaptivity

The Jaspis architecture contains a general adaptation mechanism used across system modules and applications. Each manager uses a set of evaluators to select the most suitable agent for each situation. Each evaluator gives a score for each agent and the highest scoring agent is selected.



4. Interaction management

Natural language understanding, dialogue management and response generation are distributed to various modules in AthosMail, and the application contains multiple approaches for each of these high-level tasks. In addition, several "add-ons" are included to the system in a plug-and-play manner to bring additional robustness and adaptivity.



4.1 Input interpretation

NLU is performed in two alternative ways. In the first approach the semantic analysis agents (4) try to build a complete linguistic analysis. In the second approach the concept-spotting agents (3) identify predefined and dynamically constructed concepts and resolve their relations. The dynamic concepts are adapted for each individual user. The user input is given to both sets and the results are available for other components. Semantic analysis approach deals with linguistically rich speech inputs, while the concept-spotting approach handles incomplete and even contradictory recognition results, and thus brings robustness.

4.2 Dialogue management

Dialogue management uses two approaches. The linguistic oriented agents (5, 7) deal with complex discourse structures. Specialized dialogue agents (9) are related to the high-level functions of AthosMail (e.g., reading of e-mail messages). They utilize the object-oriented approach. Inheritance is used to separate generic dialogue tasks from domain specific actions (8). This makes it possible to combine the benefits of different dialogue control models (forms, state-based), and to implement different dialogue management strategies (system-initiative, mixed-initiative) to adapt the dialogue. The dynamic dialogue management approach is presented in detail in [3].

4.3 Output generation

NLG uses two approaches. In the first approach a single generator agent produces basic outputs by concatenating together pre-defined utterance segments. In the second approach numerous presentation agents (10) produce adaptive outputs for specific situations. The structure of the mailbox, used language, user model, and the interaction history are taken into account. Agent-based response generation is able to handle multiple output requests in each turn and produce rich combinations of multilingual outputs. When appropriate, the results are concatenated together, or multiple responses are produced, for example, when multiple synthesizers with different languages are used.

4.4 Help & Guidance

Several additional features are used to bring more adaptive features and robustness to the interaction. One such add-on feature is universal commands. Examples include context-specific help messages, which are available for all system functionality. In another example, tutoring agents bring system-initiative features to the user-initiative interface. They give guidance in appropriate situations, introducing the system and monitoring how well the user interacts with it [4]. These features are implemented using specialized NLU, DM and NLG agents (3, 8, 9 & 10). They are included to the system without modifications for other components, and can be turned dynamically on and off.

References:
[1] Turunen, M. and Hakulinen, J. "Jaspis - A Framework for Multilingual Adaptive Speech Applications". Proc. of ICSLP 2000.
[2] Hartikainen, M., Salonen, E-P. and Turunen, M. "Subjective Evaluation of Spoken Dialogue Systems Using SERVQUAL Method". Proc. of ICSLP 2004.
[3] Salonen, E-P, Hartikainen, M., Turunen, M., Hakulinen, J. and Funk, J. A. "Flexible Dialogue Management Using Distributed and Dynamic Dialogue Control". Proc. of ICSLP 2004.
[4] Hakulinen, J., Turunen, M., and Salonen, E.-P. "Agents for Integrated Tutoring in Spoken Dialogue Systems". Proc. of EUROSPEECH 2003.