Flexible Dialogue Management Using Distributed and Dynamic Dialogue Control

Esa-Pekka Salonen, Mikko Hartikainen, Markku Turunen, Jaakko Hakulinen and J. Adam Funk (2)
Speech-based and Pervasive Interaction Group - Tampere Unit for Human-Computer Interaction - University of Tampere
(2) Interactive Systems Design Group - Department of Computation - UMIST

1. Introduction

AthosMail [1] is a multilingual speech-based e-mail system that has a distributed system structure including two dialogue control models and two input processing approaches.

The AthosMail application is built on top of the Jaspis architecture [2]. The architecture is based on distributed set of managers, evaluators and agents that share knowledge via Information Storage. Here agents are compact software components handling single tasks.

The resulting structure makes it possible to introduce multiple agents for same tasks and selected the most suitable depending on the situation. This makes the interaction management highly dynamic and flexible. The system architecture is presented in [3].

2. Information Storage

Every component has access to the shared Information Storage (IS). This enables the agents to be stateless, i.e., they store attributes in the IS. In practice, AthosMail maintains a discourse tree in the IS. Every user and system utterance is presented as a node in the tree. Each agent places its results to the discourse tree. The discourse tree also serves as a dialogue history.

6. Surface Generation

The responsibility of the generation agents is to generate surface forms from the abstract output requests produced by action agents. AthosMail uses a number of independent agents that respond to one or more conceptual output request(s) to produce the system's utterances. Proper agents are selected for each abstract output request to produce the surface form regarding to parameters, such as the language and amount of guidance in the surface form.

Speech Recognition Results

3. Input Interpretation

Two alternative approaches are used for NLU. Conceptual interpretation and semantic analysis are done in parallel. These two approaches provide compensating information.

3.1 Concept-based Interpretation

Concept-based interpretation agents try to identify predefined and dynamically constructed concepts from the recognized user inputs (speech and DTMF), and resolve their relations. These agents can handle incomplete recognition results and special situations.

3.2 Semantic Analysis

The semantic analysis agents (based on the Parasite system [4]) use functional dependency analysis, compositional semantics and property theory to represents the propositional content of the utterance in a logical form. These agents are able to deal with linguistically rich speech inputs.

4. Dialogue Modeling

AthosMail contains two dialogue control models: one uses specialized dialogue agents, and other linguistically motivated dialogue agents. Specialized agents handle tasks related to application functionality, while the latter ones deal with complex discourse structures. Both approaches produce an abstract suggestion of the system response.

4.1 Specialized Dialogue Agents

The specialized dialogue agents represent application functionality. AthosMail contains roughly one agent per high-level application function. For example, the application contains an agent that responds when a call is received, and an agent for giving context-sensitive help.

The dialogue agents are implemented using the object-oriented software development paradigm, and they utilize inheritance to separate generic dialogue management from domain specific actions.

In AthosMail we have successfully re-used and inherited agents from a previous e-mail application Mailman [4] and from the underlying Jaspis architecture.

4.2 Pragmatic Processing

The pragmatic agent maintains a discourse model which represents the entities in the discourse and facts about them. For each user utterance, it anchors the logical form by dereferencing referring expressions where possible to suitable entities existing in the model and by creating new entities needed to accommodate some expressions. The pragmatic agent uses its theorem prover to find solutions to questions such as "What new messages do I have from Bob?".

4.3 Response Planner

The response planning agent interprets the goal from the pragmatic agent and information from the user model using a set of declarative rules to produce a plan that describes the system's utterance as well as any appropriate non-verbal actions (such as deleting a message).

5. Domain Specific Actions

Domain dependent tasks that the dialogue agents suggest are handled with action agents. Their responsibilities are: to do the needed actions to the IS (e.g. mailbox), communication with the mail server, and conveying the dialogue suggestions to the input and output devices. Each system action is presented in a separate agent, which extend the specialized dialogue agents. Inheritance is used to maximize run-time efficiency and to separate domain specific tasks from generic ones.

User Modelling

References:
[1] Turunen, M. et al. "AthosMail - a Multilingual Adaptive Spoken Dialogue System for E-mail Domain". Proc. of COLING Workshop Robust and Adaptive Information Processing for Mobile Speech Interfaces 2004.
[2] Turunen, M. & Hakulinen, J. "Jaspis - A Framework for Multilingual Adaptive Speech Applications". Proc. of ICSLP 2000.
[3] Turunen, M., Salonen, E-P., Hartikainen, M., Hakulinen, J. "Robust and Adaptive Architecture for Multilingual Spoken Dialogue Systems". Proc. of ICSLP 2004.
[4] Ramsay, A. The logical structure of English: computing semantic content, London: Pitman, 1990.