

Adaptivity in Speech-based Multilingual E-mail Client

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

University of Tampere, Department of Computer
Sciences, TAUCHI, SPI - group
33014 University of Tampere
dumas@cs.uta.fi

Jyrki Rissanen, Kari Kanto, Kristiina Jokinen

University of Art and Design Helsinki,
Hämeentie 135 C, FIN-00560 Helsinki,
Finland
dumas@uia.fi

ABSTRACT

In speech interfaces users must be aware what can be done with the system – in other words, the system must provide information to help the users to know what to say. We have addressed this challenge by using adaptive techniques that support the learning and use of speech applications. We describe how adaptivity can be supported on architectural level, how user modeling can help to make the interface more adaptive, how integrated tutoring teaches the users to use speech applications and how context adaptive universal commands support cross domain learning. Specific issues concerning e-mail domain are discussed and examples from a working speech-based e-mail application are given.

Author Keywords

Voice I/O, adaptation.

ACM Classification Keywords

H 5.2, User-centered design, Voice I/O.

INTRODUCTION

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. In general, adaptive techniques help different users by utilizing customized interaction methods and techniques. An example of the benefits of interaction level adaptivity is reported by Litman and Pan [4]. In this paper we present several adaptive features of a multilingual speech based e-mail client called AthosMail. These features make the speech interface easier to learn and more effective and pleasant to use.

E-mail domain is particularly suitable area for adaptive speech applications since dialogues tend to be more open-ended than structured. This is because there is no single static task that the user tries to solve. This makes speech based e-mail application very different from, e.g. timetable systems that are commonly modeled as forms. The lack of fixed goal means that the interaction model for this kind of system is user-initiated. However, users must somehow be aware of what can be done with the system. To address this issue we have implemented different adaptive features that help the users in learning how such systems can be used. These features include integrated tutoring, output generation that takes user expertise into account and context adaptive universal commands.

The domain of the application is one of the major issues in interface design, especially in speech interfaces. For example, a speech-based e-mail client must be adaptive to the contents of the mailbox and individual messages. In a nutshell, speech-based interaction in e-mail domain is about navigation between messages and reading of relevant ones. To make the navigation tasks easier we have implemented components that categorize the messages into semantically meaningful groups, thus adapting the structure of the mailbox suitable for speech user interfaces. Furthermore, the interaction is adapted to the mailbox structure, and the e-mail messages are presented to the user taking into account the characteristics of the user and the messages.

Next we present the functionality of the AthosMail system. After that, we introduce the adaptive system architecture, the user modeling components and different adaptive features of the system (integrated tutoring, adaptation to the domain information). Finally, conclusions are presented.

FUNCTIONALITY OF THE ATHOSMAIL SYSTEM

AthosMail is a telephone based e-mail reading client developed in the EU-funded DUMAS project (Dynamic Universal Mobility for Adaptive Speech Interfaces, IST-2000-29452). AthosMail is based on the existing Mailman application [9] and developed further to add adaptive features to the system [11]. AthosMail consists of two systems: the offline e-mail processing system called the MailServer and the spoken dialogue system, AthosMail that the user interacts with. The user can give spoken inputs

such as “Do I have any messages from Kalle”. The functionality of AthosMail covers the basic e-mail reading functions such as reading and deleting of messages. In addition the user can mark messages as interesting; this is later used in user modeling. The interaction model in AthosMail is user initiated with some mixed-initiative features. Mixed-initiative features are used for certain critical functions, namely deleting and quitting, in these cases the system confirms the action before executing it.

ADAPTIVE ARCHITECTURE

The AthosMail application is constructed on top of the Jaspis architecture [8]. 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.

Jaspis-based applications store all their information in the shared Information Storage. 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-level adaptation, as presented next.

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.

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 integrated tutoring, are included to the system in this way.

An example of the system-level adaptation

The general system level adaptation mechanism is applied to various tasks. For example, the presentation agents (used for producing outputs) 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. The details of the AthosMail system architecture are presented in [10] and its adaptive dialogue management approach is presented in [6].

MODELLING OF USER EXPERTISE

The purpose of AthosMail user modeling is to create the basis for the flexibility and variation of system outputs, so as to allow the users interact in a more natural way. The user expertise model utilized in AthosMail features first-order parameters aimed at observing telltale signals of the user's skill level, and a set of second-order parameter (DASEX, dialogue act specific explicitness) that reflects what has been concluded from the first-order parameters [3]. The first-order parameters are tuned to spot incoherence between new information and the current user model. If there's evidence that the user is actually more experienced than previously thought, the user expertise model is updated accordingly. The process can proceed to the other direction as well, if the user model has been too fast in concluding that the user has advanced to a higher level of expertise.

There is a separate experience value for each system function, which enables the system to behave appropriately even if the user is very experienced in using one function but has never used another. For inexperienced users, the system utterances are more explicit and contain more additional advice regarding the functionality the utterance is related to. DASEX values reflect the perceived user expertise and set the corresponding level of explicitness. The value range is: 1 = expert, 2 = competent, 3 = novice. An example of the effect of the DASEX variation follows:

U: “Select the third group”

S (DASEX 3): “Group three contains messages from Kristiina Jokinen. Subject of first message is: D6.1. Subject of second, tele-meeting on Tuesday 21/10. Third, Deliverables due!! You can choose a message saying for example first message or second message. If you want further instructions say help”

S (DASEX 1): “Group three, messages from Kristiina Jokinen. Subject of first message is: D6.1. Subject of second, tele-meeting on Tuesday 21/10. Third, Deliverables due!!”

Example 1: DASEX variation (levels 3 and 1).

The model comprises an online component and an offline component. The former is responsible for observing runtime events and calculating DASEX recommendations on the fly, whereas the latter makes long-time observations and, based on these, calculates default DASEX values to be used at the beginning of the next session. The offline component is, so to speak, rather conservative; it operates on statistical event distributions instead of individual parameter values and tends to round off the extremes, trying to catch the overall learning curve behind the local variations. The components work separately. In the beginning of a new session, the current offline model of the

user's skill level is copied onto the online component and used as the basis for producing the DASEX recommendations, while at the end of each session, the offline component calculates the new default level on the basis of the occurred events. The user expertise is utilized as a part of the evaluation process, as presented in the previous section.

INTEGRATED TUTORING

In addition to monitoring the user expertise and adapting the speech outputs to the user we have developed an alternative approach – integrated tutoring [1]. In this approach the user interface (e.g. speech outputs) stays the same and the guidance is given by using a set of tutoring agents that teach how to use the system.

The tutor explicitly tells the user what to say and monitors that the user does so. The tutoring takes part at the same time when the user is using the system so that the tutor interrupts the dialogue and gives advice to the user and explains what the system is doing. The tutoring components were simply added as an additional set of components to existing system, thus utilizing the flexibility and adaptivity of the architecture [1].

S: "Please wait, connecting to your mail server... Hi, Test user. No new messages. 17 old messages. You have 3 groups. Group one, ..."

T: "Hi, I'm your tutor. I'll teach you how to use the system. Next choose one of the available groups. You can do this by saying, for example, 'third group'. So, use the group number you wish."

U: "Third group"

T: "Good. Now you are in the third group. Next AthosMail will list messages in the group."

S: <the system selects the wanted group>

Example 2: Tutoring feature.

UNIVERSAL COMMANDS

In addition to the guidance given by the system, the user is able to request more help. The system includes two context adaptive features that can be accessed via universal commands that are used across domains. These features are motivated by previous research [5]. "Tell me more!" command can be used to get more detailed information about the last system utterance. "What next?"-command can be used to get context sensitive help. The "What next?" feature examines the dialogue history and adapts the help message regarding to the state of the dialogue.

S: "Message number one from Adam Funk, subject is "AthosMail review"

U: "Tell me more"

S: "Message number one from Adam Funk, subject is "AthosMail review, urgent". Message was sent on Wednesday at 14.24. It has 6 lines. Summary of message: "AthosMail review. Hi, I got this from you."

U: "What next"

S: "You can make the system read the message by saying "read" or you can select another message by saying for example "second message"

Example 3: Context adaptive universal commands.

ADAPTATION TO THE DOMAIN

The domain affects the design of a speech interface considerably, the overall contents of the mailbox being the most influential feature in e-mail systems. For example, the organization of messages affects the whole interaction. Furthermore, speech technology sets limits on what can be done. In the following we describe how adaptive techniques are used to address these questions.

Categorization of messages

By categorizing messages into semantically meaningful groups it is possible to give the user a quick overview of the mailbox contents. The categorization also reduces the length of system outputs, by splitting long message lists to smaller units. It is well known that long spoken lists are hard to understand [7]. The user can navigate between message groups by referring them with the ordinal number, by using referring expressions 'next', 'previous' and 'last', as presented in example 3.

The categorization of messages is done regarding to various criteria, such as sender, subject, date and importance. The categorization algorithm divides messages into manageable amount of groups, putting similar messages to the same group. The algorithm favours sets of groups that have three to nine groups and three to seven messages within each group. However, it is possible that there are only a few messages in the users' mailbox and only one group is formed. In this case the interface is adapted so that the system does not talk about groups. E.g. in the first prompt of example 2 the messages would be listed instead of groups and the interaction with tutor would proceed without teaching how to select groups. When needed, new groups are formed dynamically (see example 4).

Modelling of message importance

The Message Priority component analyses how important a message is for a user and it suggests possible message orderings for folder based on the importance of the messages in the folders. The importance of a message is a function of user actions: what the user has done earlier with the same kind of messages.

The user commands are divided into three different categories depending on their impact on the ranking of the messages: actions that are indicative of the user's increased interest towards the message content (positive actions), and those that indicate the opposite (negative actions). Furthermore, a third class, that of neutral actions is also possible. An example of a positive action could be when the user moves a message to some folder, and an example of a negative action could be when the user deletes a message without never reading or listening it.

The Message Priority component analyses message features such as the sender, receiver group, subject, keywords (if available) and topics (if available), and gives a score from -1 to +1 to each of these pieces of information separately. E.g. if the message is from 'Mary' and the user has moved

the message to a folder, this is a positive action and therefore the score of sender 'Mary' will be raised. The importance of each message is a weighted average of the scores of message features (sender, subject, keywords etc.).

The use of sender names

It is a natural way to query for relevant messages by using sender names in a speech based e-mail system. The limitations of speech recognition, makes it impossible to use a static list of all possible names. Instead, the most suitable option is to extract the sender names from the mailbox each time new messages arrive and add them dynamically to the speech recognition grammars of each individual user. Furthermore, when Finnish is used the names must be inflected to proper word form.

When the user logs into the system the pre-processed mailbox (containing the extracted and inflected list of sender names) is fetched from the MailServer and added to the speech recognition grammars. This is done with agents that are selected dynamically depending on the language and the used recognizer. These agents decide which words and in which form they are added. The following example shows how the sender names can be used in AthosMail. As demonstrated, the grouping of messages is adapted dynamically to the actual use of sender names.

S: "Three groups, group one contains the messages with subject "regarding: Your question", group two messages with subject "Change suggestions" group three all the rest"

U: "List senders"

S: "Five senders. Listing senders. Adam Funk, Markku Turunen, ..."

U: "What messages do I have from Adam Funk?"

S: "I created a new group. Fourth group contains three messages from Adam Funk. Subject of first message is ..."

Example 4: The use of sender names.

Reading of messages

The reading of messages is crucial factor in e-mail domain. Messages are processed in several ways to make speech outputs intelligible and pleasant. When new messages are fetched they are converted into structural form. E-mail specific text elements, such as emoticons and web- and e-mail addresses, are detected and read in an adaptive way, taking into account the characteristics of the technology, user and message. Messages containing multiple languages are spoken so that synthesizers are changed between chapters in a message regarding to the recognized language.

CONCLUSIONS

We have presented several adaptation techniques for speech user interfaces. Examples from a working application were presented. The domain of an application has a major influence on how the speech interface should be designed and implemented. In the domains where user initiated interaction model is preferable it is beneficial to make the interface adaptive for different users and user groups. We believe that no single technique itself solves all problems of

speech user interfaces. Instead, by making alternative solutions available and adapting the interface dynamically, the users can use the method they see best for them. We have presented how AthosMail, a speech-based e-mail client, adapts the user interface regarding to user expertise and the contents of the mailbox in order to make the interaction more fluent and effective. We have conducted user tests with AthosMail that show promising results in means of system acceptability and user satisfaction [2].

REFERENCES

1. Hakulinen, J., Turunen, M., & Salonen, E-P., "Agents for Integrated Tutoring in Spoken Dialogue Systems", *Proc. of the Eurospeech 2003*, 2003.
2. Hartikainen, M., Salonen, E-P. & Turunen, M., "Subjective Evaluation of Spoken Dialogue Systems Using SERVQUAL Method", *Proc. of ICSLP 2004*, 2004. (to appear)
3. Jokinen, K. & Kanto, K., "User Expertise Modelling and Adaptivity in a Speech-based E-mail System". *Proc. of ACL-2004*, 2004. (to appear)
4. Litman, D., Pan, S. Designing and Evaluating an Adaptive Spoken Dialogue System. *User Modeling and User-Adapted Interaction, Vol. 12, No. 2/3*, 2002.
5. Rosenfeld, R., Olsen, D., Rudnicky, A. Universal speech interfaces. *ACM Interactions, Vol. 8, No. 6, ACM Nov./Dec.*, 2001.
6. Salonen, E-P., Hartikainen, M., Turunen, M., Hakulinen J. & Funk, J.A., "Flexible Dialogue Management Using Distributed and Dynamic Dialogue Control", *Proc. of ICSLP 2004*, 2004. (to appear)
7. Suhm, B., "Towards Best Practices for Speech User Interfaces", *Proc. of the Eurospeech 2003*, 2003.
8. Turunen, M. & Hakulinen, J., "Jaspis - A Framework for Multilingual Adaptive Speech Applications", *Proc. of ICSLP 2000*, 2000.
9. Turunen, M. & Hakulinen, J., "Mailman - a Multilingual Speech-only E-mail Client based on an Adaptive Speech Application Framework", *Proc. of Workshop on Multi-Lingual Speech Communication (MSC 2000)*, 2000.
10. Turunen, M., Salonen, E-P., Hartikainen, M. & Hakulinen, J. "Robust and Adaptive Architecture for Multilingual Spoken Dialogue Systems. *Proc. of ICSLP 2004*, 2004. (to appear)
11. Turunen, M., Salonen, E-P., Hartikainen, M., Hakulinen, J., Black, W.J., Ramsay, A. Funk, A., Conroy, A., Thompson, P., Stairmand, M., Jokinen, K., Rissanen, J., Kanto, K., Kerminen, A., Gambäck, B., Cheadle, M., Olsson, F., Sahlgren, M. 2004. "AthosMail – a multilingual Adaptive Spoken Dialogue System for E-mail Domain" *Proc. of the COLING Workshop Robust and Adaptive Information Processing for Mobile Speech Interfaces*, 2004. (to appear).