

User Expertise Modelling and Adaptivity in a Speech-based E-mail System

Kristiina JOKINEN

University of Helsinki
and
University of Art and Design Helsinki
Hämeentie 135C
00560 Helsinki
kjokinen@uiah.fi

Kari KANTO

University of Art and Design Helsinki
Hämeentie 135C
00560 Helsinki
kanto@uiah.fi

Abstract

This paper describes the user expertise model in AthosMail, a mobile, speech-based e-mail system. The model encodes the system's assumptions about the user expertise, and gives recommendations on how the system should respond depending on the assumed competence levels of the user. The recommendations are realized as three types of explicitness in the system responses. The system monitors the user's competence with the help of parameters that describe e.g. the success of the user's interaction with the system. The model consists of an online and an offline version, the former taking care of the expertise level changes during the same session, the latter modelling the overall user expertise as a function of time and repeated interactions.

1 Introduction

Adaptive functionality in spoken dialogue systems is usually geared towards dealing with communication disfluencies and facilitating more natural interaction (e.g. Danieli and Gerbino, 1995; Litman and Pan, 1999; Krahmer et al, 1999; Walker et al, 2000). In the AthosMail system (Turunen et al., 2004), the focus has been on adaptivity that addresses the user's expertise levels with respect to a dialogue system's functionality, and allows adaptation to take place both online and between the sessions.

The main idea is that while novice users need guidance, it would be inefficient and annoying for experienced users to be forced to listen to the same instructions every time they use the system. For instance, already (Smith, 1993) observed that it is safer for beginners to be closely guided by the

system, while experienced users like to take the initiative which results in more efficient dialogues in terms of decreased average completion time and a decreased average number of utterances. However, being able to decide when to switch from guiding a novice to facilitating an expert requires the system to be able to keep track of the user's expertise level. Depending on the system, the migration from one end of the expertise scale to the other may take anything from one session to an extended period of time.

In some systems (e.g. Chu-Carroll, 2000), user inexperience is countered with initiative shifts towards the system, so that in the extreme case, the system leads the user from one task state to the next. This is a natural direction if the application includes tasks that can be pictured as a sequence of choices, like choosing turns from a road map when navigating towards a particular place. Examples of such a task structure include travel reservation systems, where the requested information can be given when all the relevant parameters have been collected. If, on the other hand, the task structure is flat, system initiative may not be very useful, since nothing is gained by leading the user along paths that are only one or two steps long.

Yankelovich (1996) points out that speech applications are like command line interfaces: the available commands and the limitations of the system are not readily visible, which presents an additional burden to the user trying to familiarize herself with the system. There are essentially four ways the user can learn to use a system: 1) by unaided trial and error, 2) by having a pre-use tutorial, 3) by trying to use the system and then asking for help when in trouble, or 4) by relying on advice the system gives when concluding the user

is in trouble. Kamm, Litman & Walker (1998) experimented with a pre-session tutorial for a spoken dialogue e-mail system and found it efficient in teaching the users what they can do; apparently this approach could be enhanced by adding items 3 and 4. However, users often lack enthusiasm towards tutorials and want to proceed straight to using the system.

Yankelovich (1996) regards the system prompt design at the heart of the effective interface design which helps users to produce well-formed spoken input and simultaneously to become familiar with the functionality that is available. She introduced various prompt design techniques, e.g. tapering which means that the system shortens the prompts for users as they gain experience with the system, and incremental prompts, which means that when a prompt is met with silence (or a timeout occurs in a graphical interface), the repeated prompt will be incorporated with helpful hints or instructions. The system utterances are thus adapted online to mirror the perceived user expertise.

The user model that keeps track of the perceived user expertise may be session-specific, but it could also store the information between sessions, depending on the application. A call service providing bus timetables may harmlessly assume that the user is always new to the system, but an e-mail system is personal and the user could presumably benefit from personalized adaptations. If the system stores user modelling information between sessions, there are two paths for adaptation: the adaptations take place between sessions on the basis of observations made during earlier sessions, or the system adapts online and the resulting parameters are then passed from one session to another by means of the user model information storage. A combination of the two is also possible, and this is the chosen path for AthosMail as disclosed in section 3.

User expertise has long been the subject of user modelling in the related fields of text generation, question answering and tutorial systems. For example, Paris (1988) describes methods for taking the user's expertise level into account when designing how to tailor descriptions to the novice and expert users. Although the applications are somewhat different, we expect a fair amount of further inspiration to be forthcoming from this direction also.

In this paper, we describe the AthosMail user expertise model, the Cooperativity Model, and discuss its effect on the system behaviour. The paper is organised as follows. In Section 2 we will first briefly introduce the AthosMail functionality which the user needs to familiarise herself with. Section 3 describes the user expertise model in more detail. We define the three expertise levels and the concept of DASEX (dialogue act specific explicitness), and present the parameters that are used to calculate the online, session-specific DASEX values as well as offline, between-the-sessions DASEX values. We also list some of the system responses that correspond to the system's assumptions about the user expertise. In Section 4, we report on the evaluation of the system's adaptive responses and user errors. In Section 5, we provide conclusions and future work.

2 System functionality

AthosMail is an interactive speech-based e-mail system being developed for mobile telephone use in the project DUMAS (Jokinen and Gambäck, 2004). The research goal is to investigate adaptivity in spoken dialogue systems in order to enable users to interact with the speech-based systems in a more flexible and natural way. The practical goal of AthosMail is to give an option for visually impaired users to check their email by voice commands, and for sighted users to access their email using a mobile phone.

The functionality of the test prototype is rather simple, comprising of three main functions: navigation in the mailbox, reading of messages, and deletion of messages. For ease of navigation, AthosMail makes use of automatic classification of messages by sender, subject, topic, or other relevant criteria, which is initially chosen by the system. The classification provides different "views" to the mailbox contents, and the user can move from one view to the next, e.g. from Paul's messages to Maria's messages, with commands like "next", "previous" or "first view", and so on. Within a particular view, the user may navigate from one message to another in a similar fashion, saying "next", "fourth message" or "last message", and so on. Reading messages is straightforward, the user may say "read (the message)", when the message in question has been selected, or refer to another message by saying, for example, "read the

third message". Deletion is handled in the same way, with some room for referring expressions. The user has the option of asking the system to repeat its previous utterance.

The system asks for a confirmation when the user's command entails something that has more potential consequences than just wasting time (by e.g. reading the wrong message), namely, quitting and the deletion of messages. AthosMail may also ask for clarifications, if the speech recognition is deemed unreliable, but otherwise the user has the initiative.

The purpose of the AthosMail user model is to provide flexibility and variation in the system utterances. The system monitors the user's actions in general, and especially on each possible system act. Since the user may master some part of the system functionality, while not be familiar with all commands, the system can thus provide responses tailored with respect to the user's familiarity with individual acts.

The user model produces recommendations for the dialogue manager on how the system should respond depending on the assumed competence levels of the user. The user model consists of different subcomponents, such as Message Prioritizing, Message Categorization and User Preference components (Jokinen et al, 2004). The Cooperativity Model utilizes two parameters, explicitness and dialogue control (i.e. initiative), and the combination of their values then guides utterance generation. The former is an estimate of the user's competence level, and is described in the following sections.

3 User expertise modelling in AthosMail

AthosMail uses a three-level user expertise scale to encode varied skill levels of the users. The common assumption of only two classes, experts and novices, seems too simple a model which does not take into account the fact that the user's expertise level increases gradually, and many users consider themselves neither novices nor experts but something in between. Moreover, the users may be experienced with the system selectively: they may use some commands more often than others, and thus their skill levels are not uniform across the system functionality.

A more fine-grained description of competence and expertise can also be presented. For instance,

Dreyfus and Dreyfus (1986) in their studies about whether it is possible to build systems that could behave in the way of a human expert, distinguish five levels in skill acquisition: Novice, Advanced beginner, Competent, Proficient, and Expert. In practical dialogue systems, however, it is difficult to maintain subtle user models, and it is also difficult to define such observable facts that would allow fine-grained competence levels to be distinguished in rather simple application tasks. We have thus ended up with a compromise, and designed three levels of user expertise in our model: novice, competent, and expert. These levels are reflected in the system responses, which can vary from explicit to concise utterances depending on how much extra information the system is to give to the user in one go.

As mentioned above, one of the goals of the Cooperativity model is to facilitate more natural interaction by allowing the system to adapt its utterances according to the perceived expertise level. On the other hand, we also want to validate and assess the usability of the three-level model of user expertise. While not entering into discussions about the limits of rule-based thinking (e.g. in order to model intuitive decision making of the experts according to the Dreyfus model), we want to study if the designed system responses, adapted according to the assumed user skill levels, can provide useful assistance to the user in interactive situations where she is still uncertain about how to use the system.

Since the user can always ask for help explicitly, our main goal is not to study the decrease in the user's help requests when she becomes more used to the system, but rather, to design the system responses so that they would reflect the different skill levels that the system assumes the user is on, and to get a better understanding whether the expertise levels and their reflection in the system responses is valid or not, so as to provide the best assistance for the user.

3.1 Dialogue act specific explicitness

The user expertise model utilized in AthosMail is a collection of parameters aimed at observing tell-tale signals of the user's skill level and a set of second-order parameters (dialogue act specific explicitness DASEX, and dialogue control CTL) that reflect what has been concluded from the first-

order parameters. Most first-order parameters are tuned to spot incoherence between new information and the current user model (see below). If there's evidence that the user is actually more experienced than previously thought, the user expertise model is updated to reflect this. The process can naturally proceed in 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. The second-order parameters affect the system behaviour directly. 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. The higher the value, the less experienced the user; the less experienced the user, the more explicit the manner of expression and the more additional advice is incorporated in the system utterances. The values are called DASEX, short for Dialogue Act Specific Explicitness, and their value range corresponds to the user expertise as follows: 1 = expert, 2 = competent, 3 = novice.

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.

Figure 1 provides an illustration of the relationships between the parameters. In the next section we describe them in detail.

3.1.1 Online parameter descriptions

The online component can be seen as an extension of the ideas proposed by Yankelovich (1996) and Chu-Carroll (2000). The relative weights of the parameters are those used in our user tests, partly based on those of (Krahmer et al, 1999). They will be fine-tuned according to our results.

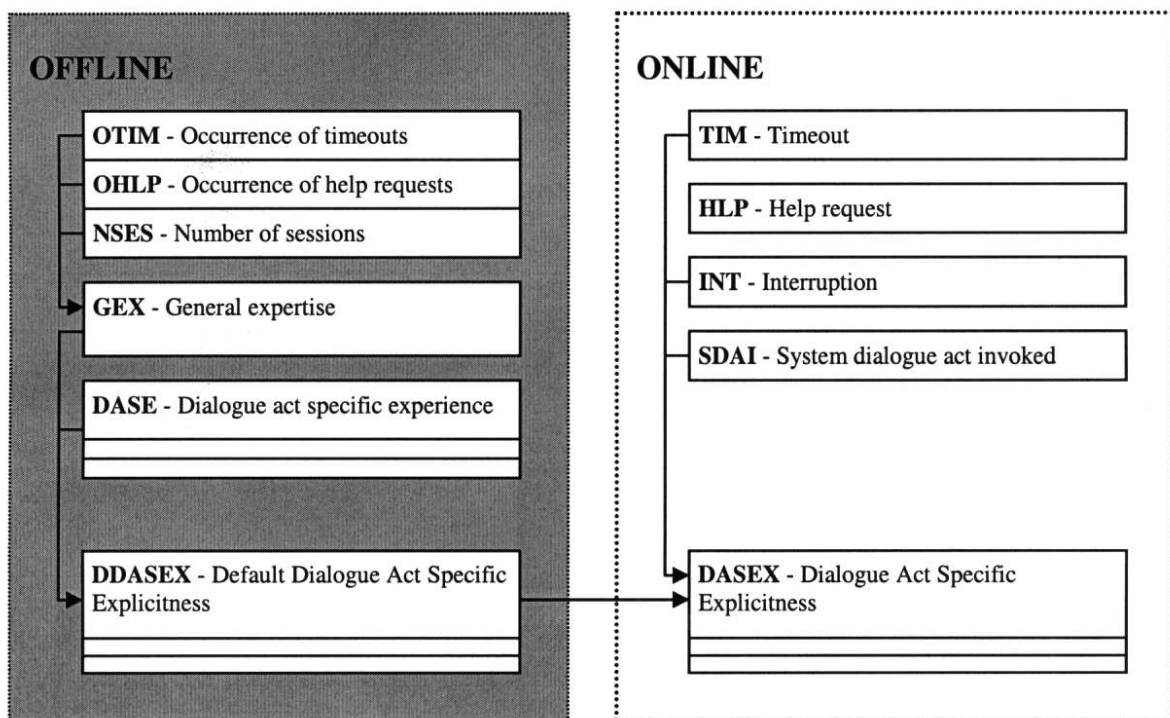


Figure 1 The functional relationships between the offline and online parameters used to calculate the DASEX values.

DASEX (dialogue act specific explicitness): The value is modified during sessions. Value: DDASEX (see offline parameters) modified by SDAI, HLP, TIM, and INT as specified in the respective parameter definitions.

SDAI (system dialogue act invoked): A set of parameters (one for each system dialogue act) that tracks whether a particular dialogue act has been invoked during the previous round. If SDAI = 'yes', then DASEX -1. This means that when a particular system dialogue move has been instantiated, its explicitness value is decreased and will therefore be presented in a less explicit form the next time it is instantiated during the same session.

HLP (the occurrence of a help request by the user): The system incorporates a separate help function; this parameter is only used to notify the offline side about the frequency of help requests.

TIM (the occurrence of a timeout on the user's turn): If TIM = 'yes', then DASEX +1. This refers to speech recognizer timeouts.

INT (occurrence of a user interruption during system turn): Can be either a barge-in or an interruption by telephone keys. If INT = 'yes', then DASEX = 1.

3.1.2 Offline parameter descriptions

DDASEX (default dialogue act specific explicitness): Every system dialogue act has its own default explicitness value invoked at the beginning of a session. Value: DASE + GEX / 2.

GEX (general expertise): General expertise. A general indicator of user expertise. Value: NSES + OHLP + OTIM / 3.

DASE (dialogue act specific experience): This value is based on the number of sessions during which the system dialogue act has been invoked. There is a separate DASE value for every system dialogue act.

number of sessions	DASE
0-2	3
3-6	2
more than 7	1

NSES (number of sessions): Based on the total number of sessions the user has used the system.

number of sessions	NSES
0-2	3
3-6	2
more than 7	1

OHLP (occurrence of help requests): This parameter tracks whether the user has requested system help during the last 1 or 3 sessions. The HLP parameter is logged by the online component.

HLP occurred during OHLP

<i>the last session</i>	3
<i>the last 3 sessions</i>	2
if not	1

OTIM (occurrence of timeouts): This parameter tracks whether a timeout has occurred during the last 1 or 3 sessions. The TIM parameter is logged by the online component.

TIM occurred during OTIM

<i>the last session</i>	3
<i>the last 3 sessions</i>	2
if not	1

3.2 DASEX-dependent surface forms

Each system utterance type has three different surface realizations corresponding to the three DASEX values. The explicitness of a system utterance can thus range between [1 = taciturn, 2 = normal, 3 = explicit]; the higher the value, the more additional information the surface realization will include (cf. Jokinen and Wilcock, 2001). The value is used for choosing between the surface realizations which are generated by the presentation components as natural language utterances. The following two examples have been translated from their original Finnish forms.

Example 1: A speech recognition error (the ASR score has been too low).

DASEX = 1: I'm sorry, I didn't understand.

DASEX = 2: I'm sorry, I didn't understand. Please speak clearly, but do not over-articulate, and speak only after the beep.

DASEX = 3: I'm sorry, I didn't understand. Please speak clearly, but do not over-articulate, and speak only after the beep. To hear examples of what you can say to the system, say 'what now'.

Example 2: Basic information about a message that the user has chosen from a listing of messages from a particular sender.

DASEX = 1: First message, about "reply: sample file".

DASEX = 2: First message, about "reply: sample file". Say 'tell me more', if you want more details.

DASEX = 3: First message, about "reply: sample file". Say 'read', if you want to hear the messages, or 'tell me more', if you want to hear a summary and the send date and length of the message.

These examples show the basic idea behind the DASEX effect on surface generation. In the first example, the novice user is given additional information about how to try and avoid ASR problems, while the expert user is only given the error message. In the second example, the expert user gets the basic information about the message only, whereas the novice user is also provided with some possible commands how to continue. A full interaction with AthosMail is given in Appendix 1.

4 Evaluation of AthosMail

Within the DUMAS project, we are in the process of conducting exhaustive user studies with the prototype AthosMail system that incorporates the user expertise model described above. We have already conducted a preliminary qualitative expert evaluation, the goal of which was to provide insights into the design of system utterances so as to appropriately reflect the three user expertise levels, and the first set of user evaluations where a set of four tasks was carried out during two consecutive days.

4.1 Adaptation and system utterances

For the expert evaluation, we interviewed 5 interactive systems experts (two women and three men). They all had earlier experience in interactive systems and interface design, but were unfamiliar with the current system and with interactive email systems in general. Each interview included three walkthroughs of the system, one for a novice, one for a competent, and one for an expert user. The experts were asked to comment on the naturalness and appropriateness of each system utterance, as well as provide any other comments that they may have on adaptation and adaptive systems.

All interviewees agreed on one major theme, namely that the system should be as friendly and reassuring as possible towards novices. Dialogue systems can be intimidating to new users, and many people are so afraid of making mistakes that they give up after the first communication failure, regardless of what caused it. Graphical user

interfaces differ from speech interfaces in this respect, because there is always something salient to observe as long as the system is running at all.

Four of the five experts agreed that in an error situation the system should always signal the user that the machine is to blame, but there are things that the user can do in case she wants to help the system in the task. The system should acknowledge its shortcomings "humbly" and make sure that the user doesn't get feelings of guilt – all problems are due to imperfect design. E.g., the responses in Example 1 were viewed as accusing the user of not being able to act in the correct way. We have since moved towards forms like "I may have misheard", where the system appears responsible for the miscommunication. This can pave the way when the user is taking the first wary steps in getting acquainted with the system.

Novice users also need error messages that do not bother the user with technical matters that concern only the designers. For instance, a novice user doesn't need information about error codes or characteristics of the speech recognizer; when ASR errors occur, the system can simply talk about not hearing correctly; a reference to a piece of equipment that does the job – namely, the speech recognizer – is unnecessary and the user should not be burdened with it.

Experienced users, on the other hand, wish to hear only the essentials. All our interviewees agreed that at the highest skill level, the system prompts should be as terse as possible, to the point of being blunt. Politeness words like "I'm sorry" are not necessary at this level, because the expert's attitude towards the system is pragmatic: they see it as a tool, know its limitations, and "rudeness" on the part of the system doesn't scare or annoy them anymore. However, it is not clear how the change in politeness when migrating from novice to expert levels actually affects the user's perception of the system; the transition should at least be gradual and not too fast. There may also be cultural differences regarding certain politeness rules.

The virtues of adaptivity are still a matter of debate. One of the experts expressed serious doubt over the usability of any kind of automatic adaptivity and maintained that the user should decide whether she wants the system to adapt at a given moment or not. In the related field of tutoring systems, Kay (2001) has argued for giving the user the control over adaptation. Whatever the

case, it is clear that badly designed adaptivity is confusing to the user, and especially a novice user may feel disoriented if faced with prompts where nothing seems to stay the same. It is essential that the system is consistent in its use of concepts, and manner of speech.

In AthosMail, the expert level (DASEX=1 for all dialogue acts) acts as the core around which the other two expertise levels are built. While the core remains essentially unchanged, further information elements are added after it. In practise, when the perceived user expertise rises, the system simply removes information elements that have become unnecessary from the end of the utterance, without touching the core. This should contribute to a feeling of consistency and dependability. On the other hand, Paris (1988) argued that the user's expertise level does not affect only the amount but the kind of information given to the user. It will prove interesting to reconcile these views in a more general kind of user expertise modeling.

4.2 Adaptation and user errors

The user evaluation of AthosMail consisted of four tasks that were performed on two consecutive days. The 26 test users, aged 20-62, thus produced four separate dialogues each and a total of 104 dialogues. They had no previous experience with speech-based dialogue systems, and to familiarize themselves to synthesized speech and speech recognizers, they had a short training session with another speech application in the beginning of the first test session. An outline of AthosMail functionality was presented to the users, and they were allowed to keep it when interacting with the system. At the end of each of the four tests, the users were asked to assess how familiar they were with the system functionality and how confident they felt about using it. Also, they were asked to assess whether the system gave too little information about its functionality, too much, or the right amount. The results are reported in (Jokinen et al, 2004). We also identified four error types, as a point of comparison for the user expertise model.

5 Conclusions

Previous studies concerning user modelling in various interactive applications have shown the

importance of the user model in making the interaction with the system more enjoyable. We have introduced the three-level user expertise model, implemented in our speech-based e-mail system, AthosMail, and argued for its effect on the behaviour of the overall system.

Future work will focus on analyzing the data collected through the evaluations of the complete AthosMail system with real users. Preliminary expert evaluation revealed that it is important to make sure the novice user is not intimidated and feels comfortable with the system, but also that the experienced users should not be forced to listen to the same advice every time they use the system. The hand-tagged error classification shows a slight downward tendency in user errors, suggesting accumulation of user experience. This will act as a point of comparison for the user expertise model assembled automatically by the system.

Another future research topic is to apply machine-learning and statistical techniques in the implementation of the user expertise model. Through the user studies we will also collect data which we plan to use in re-implementing the DASEX decision mechanism as a Bayesian network.

6 Acknowledgements

This research was carried out within the EU's Information Society Technologies project DUMAS (Dynamic Universal Mobility for Adaptive Speech Interfaces), IST-2000-29452. We thank all project participants from KTH and SICS, Sweden; UMIST, UK; ETeX Sprachsynthese AG, Germany; U. of Tampere, U. of Art and Design, Connexor Oy, and Timehouse Oy, Finland.

References

- Jennifer Chu-Carroll. 2000. MIMIC: An Adaptive Mixed Initiative Spoken Dialogue System for Information Queries. In *Procs of ANLP 6*, 2000, pp. 97-104.
- Morena Danieli and Elisabetta Gerbino. 1995. Metrics for Evaluating Dialogue Strategies in a Spoken Language System. Working Notes, AAAI Spring Symposium Series, Stanford University.
- Hubert L. Dreyfus and Stuart E. Dreyfus. 1986. *Mind over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. New York: The Free Press.

- Kristiina Jokinen and Björn Gambäck. 2004. DUMAS - Adaptation and Robust Information Processing for Mobile Speech Interfaces. *Procs of The 1st Baltic Conference "Human Language Technologies – The Baltic Perspective"*, Riga, Latvia, 115-120.
- Kristiina Jokinen, Kari Kanto, Antti Kerminen and Jyrki Rissanen. 2004. Evaluation of Adaptivity and User Expertise in a Speech-based E-mail System. *Procs of the COLING Satellite Workshop Robust and Adaptive Information Processing for Mobile Speech Interfaces*, Geneva, Switzerland.
- Kristiina Jokinen and Graham Wilcock. 2001. Adaptivity and Response Generation in a Spoken Dialogue System. In van Kuppevelt, J. and R. W. Smith (eds.) *Current and New Directions in Discourse and Dialogue*. Kluwer Academic Publishers. pp. 213-234.
- Candace Kamm, Diane Litman, and Marilyn Walker. 1998. From novice to expert: the effect of tutorials on user expertise with spoken dialogue systems. *Procs of the International Conference on Spoken Language Processing (ICSLP98)*.
- Judy Kay. 2001. Learner control. *User Modeling and User-Adapted Interaction* **11**: 111-127.
- Emiel Kraemer, Marc Swerts, Mariet Theune and Mieke Weegels. 1999. Problem Spotting in Human-Machine Interaction. In *Procs of Eurospeech '99*. Vol. 3, 1423-1426. Budapest, Hungary.
- Diane J. Litman and Shimei Pan. 2002. Designing and Evaluating an Adaptive Spoken Dialogue System. *User Modeling and User-Adapted Interaction*. Vol **12**(2/3):111-137.
- Cécile Paris. 1988. Tailoring Descriptions to a User's Level of Expertise. *Journal of Computational Linguistics*, **14** (3): 64-78.
- Ronnie W. Smith. 1993. Effective Spoken Natural Language Dialog Requires Variable Initiative Behavior: An Empirical Study. *Procs of the AAAI Fall Symposium on Human-Computer Collaboration: Reconciling Theory, Synthesizing Practice*.
- M. Turunen, E-P. Salonen, M. Hartikainen, J. Hakulinen, W.J. Black, A. Ramsay, A. Funk, A. Conroy, P. Thompson, M. Stairmand, K. Jokinen, J. Rissanen, K. Kanto, A. Kerminen, B. Gambäck, M. Cheadle, F. Olsson, M. Sahlgren. 2004. AthosMail – a multilingual Adaptive Spoken Dialogue System for E-mail Domain. *Procs of the COLING Workshop Robust and Adaptive Information Processing for Mobile Speech Interfaces*, Geneva, Switzerland.
- Marilyn A. Walker, Irene Langkilde, Jerry Wright, Allen Gorin, Diane J. Litman. 2000. Learning to Predict Problematic Situations in a Spoken Dialogue System: Experiments with How May I Help You? *Procs of NAACL'00*, Seattle, US, 210-217.
- Nicole Yankelovich. 1996. How do users know what to say? *Interactions* **3**(6): 32-43.

7 Appendix 1. Sample dialogue with a novice user

System: Welcome to AthosMail. Please dial your user code.

User: [dials the four-digit code]

S: Please dial your secret passcode.

U: [dials the passcode]

System: Please wait, I'll contact your mail server. (...) Hello Jukka, you have twelve unread messages. They have been divided into three groups. In the first group, there are the messages about Re: Comments about the draft. In the second group, the messages from Timothy. In the third group, the rest of the messages. You can select a group by saying for example "first group" or "second group". Say help if you want further instructions.

User: First group.

S: You are in the first group, with three messages about Re: Comments about the draft. The first message is from Matti Hahtomaa, the second from Jaana Jaakkola, the third from Sakari Rantala. If you want to select a message, say for example "second message" or "Eddie Everyman's message".

U: First message.

S: First message, from Matti Hahtomaa, about Re: Comments about the draft. If you want to hear the message, say "read". You can have additional information about the message by saying "tell me more".

U: Second message.

S: Second message, from Jaana Jaakkola, about Re: Comments about the draft. You can have additional information about the message by saying "tell me more".

U: Tell me more.

S: Message sent on February 3rd, 2004. Message length 24 lines.

U: Read Sakari Rantala's message.

S: Third message, from Sakari Rantala, about Re: Comments about the draft. [reads the message]
Message read. You are in the first message group.

U: Quit.

S: Are you sure you want to quit? Say yes or no.

U: Yes.

S: Bye bye.