Adaptive User Modelling in AthosMail

Kristiina Jokinen, Kari Kanto and Jyrki Rissanen

University of Art and Design Helsinki
Hämeentie 135 C FIN-00560 Helsinki Finland
[kjokinen|kanto|jyrki]@uiah.fi

Abstract. In this paper we discuss the adaptive User Model component of the AthosMail system, and describe especially the Cooperativity Model which produces recommendations for the appropriate explicitness of the system utterances, depending on the user's observed competence levels. The Cooperativity Model deals with the system's dialogue control and explicitness of the given information: these two aspects affect the system's interaction capabilities and thus naturalness of the dialogue as a whole. The model consists of an offline and an online version, which use somewhat different input parameters, due to their different functionality in the system.

1 Introduction

In recent years, the notion of adaptivity has become more important when building spoken interactive systems. The systems are to be used in mobile and versatile environment, by various users with different skills, abilities, and requirements, and the system's usefulness and usability are thus increased if its functionality can be tailored according to preferences of the various users. In interface design, adaptivity is often realised in a static way as a form of personalization whereby the users personalise their interfaces according to different colour or sound choices, and provide a list of preferences for their personal profiles so as to enable preferred filtering and rating of incoming information. Online, dynamic adaptation can be realised in the system's ability to classify users into appropriate categories, e.g. on the basis of their navigation choices or a list of specific keywords, so as to produce personalised access to information sources (e.g. InfoQuest), filtering of interesting news e.g. [1, 11] and recommendations for web-browsing [13] or TV programs [4]. Recommendation systems [5] track preference information of a group by comparing the selected items of one user to similar items selected by the other users.

In the dialogue management research, the focus has been on dialogue strategies, and the system's capability to adapt its initiativeness according to the dialogue situation seems to result in more successful dialogue systems. For instance, [10] compared a user-adaptable and non-adaptable version of an information retrieval system, and concluded that the former outperformed the latter. [3] compared an adaptive dialogue system to two non-adaptive versions of the same system (one with a system initiative and the other with a mixed-initiative dialogue strategy), and found that the adaptive system performed better than the others in terms of user satisfaction, dialogue efficiency (number of utterances), and dialogue quality (ASR performance).

The user may be given an explicit option to change system properties, e.g. the system-initiative dialogue strategy to the user-initiative one, depending on the situation [10]. This increases the system's transparency, and the user's feeling of being in control of the system adds to user satisfaction. However, from the point of view of interaction, it is important that the systems also exhibit capability to adapt and adjust their behavioral patters automatically according to various types of users, various situations, and various user actions. For instance, [17] observes 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. People also tend to adapt their behaviour to the speech partner, which suggests that a system that adapts itself to the user's behaviour could make human-computer interaction more natural, too. [13] investigated adaptivity and adaptability in a spreadsheet application, and ended up proposing a system that suggests adaptations to the user: he observed that the users wanted to influence the timing and content of adaptations, but the controlling of adaptivity was too demanding a task for the user alone, so the system should provide assistance and be helpful in cases where adaptation is possible. In the small experiment conducted by [7], the users' interaction patterns were visualized using a reinforcement-based learning strategy for the action chains that have occurred in the previous dialogues. Although the corpus was not large enough for statistical conclusions, there were a few examples which can be considered as indications of the patterns that can vary across individual users, and which could be learnt by the system in its attempts to adapt and anticipate the user's behaviour.

In spoken dialogue systems, the User Model (UM) component takes care of this kind of adaptation: it records the user characteristics and allows the system to tailor its responses so that expectations about natural and enjoyable interaction could be fulfilled. From early on, the UM research focused on providing information that would be appropriate to the user's level of expertise, i.e. the new information is presented in the form that the user is most likely to understand correctly [2, 14].

The main goal of the EU-project DUMAS is to develop a prototype interactive email system, AthosMail, with components that would make the user's interaction with the system more flexible and natural. The purpose of the User Model component in AthosMail is three-fold:

1) to provide flexibility and variation in the system utterances,
2) to allow the users to interact with the system in a more natural way,
3) to allow developers to implement and test machine learning techniques.

These goals are exemplified by the User Model design which takes into account:

1) Flexible representation for encoding the system utterances and using this for utterance generation,
2) System functionality that records the user's actions and behaviour, and estimates the user's competence levels that will be further used to give recommendations on the appropriate way of responding,
3) Machine-learning module that provides views to the user's mailbox by classifying messages on the basis of their content and the user's interest.

In this paper we focus on the goals (1) and (2): the system's ability to support natural interaction according to the user's perceived competence levels. Section 2 discusses User Modelling in the AthosMail system, and Section 3 describes the adaptation in the Cooperativity Model in detail. We will conclude with a summary of the first expertise evaluation in Section 4, and views for future research in Section 5.

2 User Modelling in AthosMail

The following sub-components have been considered relevant to the AthosMail system:

1) Message Prioritization. This allows the system to sort out incoming messages so that the messages that the user most likely finds interesting and important are in the beginning of the list. The importance of a message is a function of user actions: what the user has done earlier with the same kind of messages. For instance, if the user has always deleted messages from 'Frank' without reading or listening those messages first, it is pretty obvious that the messages from 'Frank' are not very important for the user. The message priority component analyses message features such as the sender, received

group, subject, keywords and topics, and gives a score from -1 to +1 to each of these pieces of information separately. The importance of the whole message is a weighted sum of the scores of message features.

2) Goal Guessing. This allows the system to make educated guesses about the user's goals and behavioural patterns in the interaction situation. The main benefit of the suggestions is to help dialogue management to decide what to do when there is uncertainty of the user action because of bad speech recognition. The goal suggestions can also be used to help e.g. online tutoring system to give more relevant guidance to the user.

3) Cooperativity Model. This component allows the system to give recommendations on the explicitness level of the system utterances depending on the user's competence level, and the level of dialogue control exerted by the system depending on ASR success. If there seems to be a dialogue problem which originates from limited user expertise, the user will be given more explicit guidance, and if the fault is in speech recognizer or in language understanding, the system assumes a more active role and takes more initiative. The two variables involved are dialogue act specific explicitness (DASEX) and initiative (INIT). The component consists of an online and offline parts.

4) Message Categorization. This component allows the system to compare the incoming messages to the existing ones, and to cluster messages according to their topical similarity with the existing messages. The content-based categorization is based on the Random Indexing vector space methodology [15], which accumulates semantic representations of words based on co-occurrence statistics.

5) User Preferences. These are fixed properties of the user dealing with aspects like the preferred speaking style, speed, and voice, or the preferred message senders and topics.

Figure 1. The AthosMail User Model Component Architecture

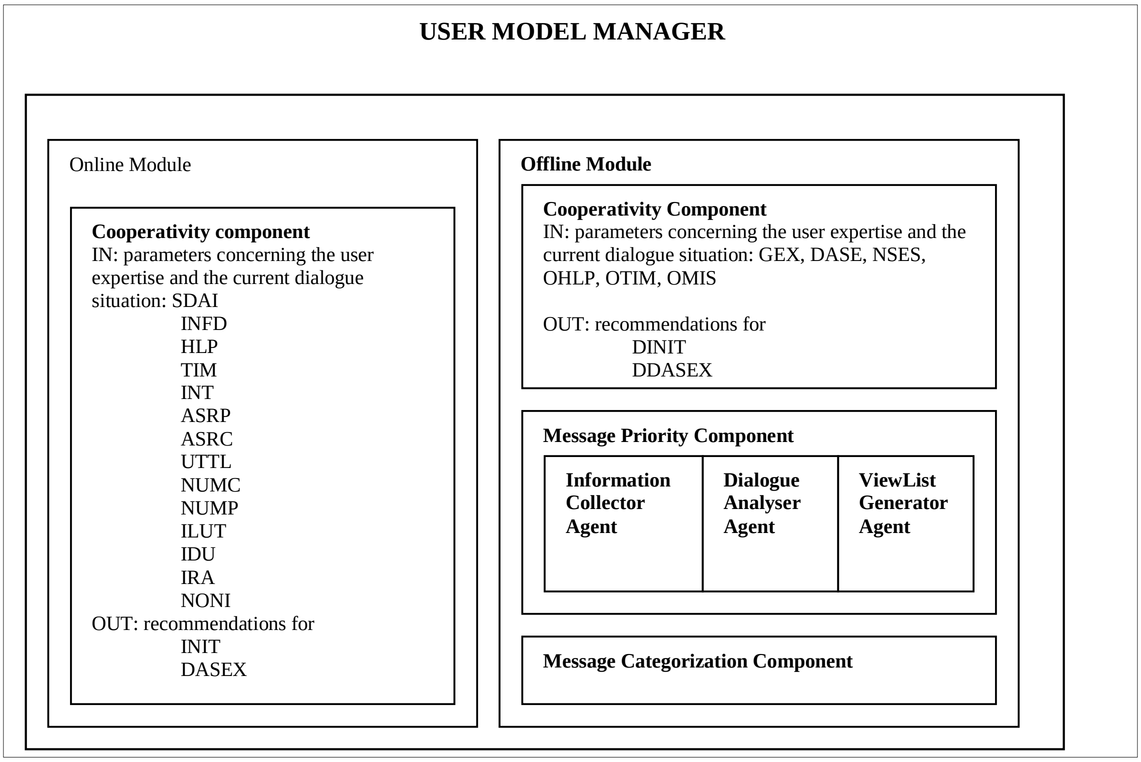

Figure 1 gives an overview of the various User Model components in AthosMail. The components produce recommendations for the Dialogue Manager Module which uses the recommendations in its own planning and generator components to decide on the next action. The recommendations concern the appropriate level of explicitness in the responses (DASEX), whether the system should assume dialogue control (INIT), and how the message list is best presented to the user when she logs in (ViewList). The recommendations are combined in the system's response planner that produces an utterance plan which can be realised in different languages using the system's presentation components. For instance, when the user is a novice, the Cooperativity component recommends the system to produce longer and more explicit utterances than when the user is familiar with the system and its functionality. The UM recommendations can also be used in the interpretation of the user utterances in order to propose expectations of the user's vocabulary and likely next actions. In the beginning of the interaction, the default user preferences are loaded into the system from the UM.

The system monitors and records the user's actions in general, but also specifically on each possible system act. Thus the system can provide help tailored with respect to the user's familiarity with individual acts. For instance, the user may need more help with commands that she does not use so often like CONFIRM or REPEAT. The users may, of course, be familiar with similar commands, but unless the users have tried the commands as part of the current system's functionality, it is controversial whether their skill with respect to the whole system can be said to be comprehensive, or whether part of it is extrapolated via adaptation from some earlier experience. The current UM model does not take into account the user's earlier experience with speech-based systems, but only their experience with AthosMail.

Explicitness in the system replies goes down when the user is more familiar with the commands, i.e. the replies become more implicit concerning the information the user needs to know at each dialogue point. However, explicit confirmation is always given whenever the user attempts to perform some of the destructive actions like cancelling the previous command or deleting a message, regardless of the user's familiarity with the system. It should be noticed that some users might want to opt out from this choice, but the choice was motivated by practical consideration of the present-day speech recognizers: to make sure that the user command was correctly interpreted, an explicit confirmation is the safest option for irreversible actions.

3 Examples of the Cooperativity Model

The User Model component is based on the existing views of what a user model should contain and on project specific requirements of how the user model can be used in the system. For instance, the Cooperativity Model is seen as a continuation of the refinement and filtering mechanism where the system's communicative goal is filtered through communicative obligations [7]. While the obligations are expressed as rules according to which the system's utterance is generated, the Cooperativity Model uses two parameters, explicitness and control, based on the system's knowledge of the current dialogue situation and user expertise.

The support for adaptive performance is motivated by the fact that the users vary in their knowledge and experience with speech-based interactive systems; they need help in learning to use a new system, and often need additional information about the various commands of the system. The two variables involved in specifying recommendations to the Dialogue Manager are dialogue act specific explicitness (DASEX) and dialogue control or initiative (INIT). The higher the explicitness value, the more explicitly the system dialogue act will be presented. The value range is: 1 = taciturn, 2 = normal, 3 = explicit. The higher the initiative value, the more the system controls the dialogue; the value range is: 1 = passive, 2 = declarative, 3 = guiding, 4 = directive. Since the effects of explicitness and initiative on the surface generation of utterances often overlap, the effect of the two variables is integrated into a unified Cooperativity Model. Table 1 gives a summary of utterance level integration.

The division of the dialogue control levels is partly based on that of [16], who established a categorization of four initiative modes: a) directive, where the computer has full control and recommends the next subgoal needed to proceed with the task, b) suggestive, where the system still holds the initiative and makes suggestions about subgoals but is willing to change the course of the dialogue according to stated user preferences, c) declarative, where the user is in control, but the system may provide relevant unsolicited information, and d) passive, where the user has complete dialogue control and the system does not recommend subgoals. The parameters that contribute to the initiative value, as well as the respective weights of the parameters, are in part based on those introduced in [18] and [9].

Table 1. The effect of Initiative and Explicitness value combinations on the surface generation of utterances.

PROMPTS			
INIT/DASEX	**(1) TACITURN**	**(2) NORMAL**	**(3) EXPLICIT**
(1) PASSIVE	options: **none** extra options: **no** prompts: **none**	options: **short** extra options: **no** prompts: **none**	INIT =+1
(2) DECLARATIVE	options: **short** extra options: **no** prompts: **none**	options: **medium** extra options: **no** prompts: **open**	options: **long** extra options: **no** prompts: **open**
(3) GUIDING	DASEX =+1	options: **medium** extra options: **no** prompts: **question**	options: **long** extra options: **yes** prompt: **question**
(4) DIRECTIVE	DASEX =+1	options: **medium** extra options: **no** prompt: **chunk by chunk**	options: **long** extra options: **yes** prompt: **chunk by chunk**

The User Model consists of two modules: the online and offline modules. They use somewhat different input parameters due to their different roles in the system functionality. The offline module keeps track of relevant dialogue history and, based on that, calculates default values for explicitness and control. The offline parameters include the number of help requests (OHLP), timeouts (OTIM), and speech recognition problems (misunderstandings, OMIS), and the module weighs them according to their frequency and recency. In addition, the module tracks the number of sessions the user has had with the system (NSES), and then calculates the user's general expertise level (GEX), default initiative and dialogue act specific explicitness (DDASEX). At the beginning of each session, the default values are passed from the offline module to the online module, and placed as the values of the active explicitness and initiative parameters. The online module modifies the default values according to the changing circumstances in the current dialogue. The online component calculates the explicitness and initiative values, based on parameters that deal with the number of help requests, timeouts, interruptions, ASR Confidence (ASRC), utterance length (UTTL), number of cancels (NUMC), etc. The dialogue events that take place during a session are recorded and relayed to the offline component at the end of the session. This way, the offline module tracks long-term developments, whereas the online module reacts to specific situations at runtime. Accordingly, the offline parameters change more slowly in order to round off coincidental fluctuation, while the online module reacts rather quickly to the user's actions, so that the user's adaptation to the system functionality can be addressed runtime immediately. The parameters and their values are described in more detail in [8].

Each system utterance type has three different surface realizations corresponding to the DASEX values. The higher the DASEX value, the more additional information the surface realization will include. The value is used for choosing between the surface realizations. The following two examples have been translated from their original Finnish forms.

Example 1: A speech recognition error message.

DASEX = 1: 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, I may have misheard what you said. 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 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.

The 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. In both examples, some possible commands are hinted at.

The effect of dialogue control or initiative (INIT) to the surface realizations is described by way of an example depicted in Table 2. Detailed parameter calculations have been substituted by verbal descriptions.

4 Expert evaluations

Prior to usability tests, we have conducted a preliminary qualitative expert evaluation, which has provided some important insights into the design of system utterances in relation to user expertise. We interviewed a group of 5 experts of interactive systems (two women and three men) who had earlier experience in interactive systems and interface design but who were unfamiliar with the current system and with interactive email systems in general. Each interview included three walkthroughs of our system.

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. Four of the five experts agreed that in an error situation the system should signal the user that the machine is to blame, but that there are things that the user can do in case she wants to help the system in the task. The system should apologize for being imperfect but also ensure that the user doesn't get feelings of guilt of not being able to act in the correct way. For instance, the responses in Example 1 were considered too much accusing the user in this way.

What novice users also need are error messages that do not bother the user with technical matters that concern only the designers. For instance, when ASR errors occur, a novice user doesn't need information about error codes or characteristics of the speech recognizer; but the system can simply talk about not hearing correctly. A reference to a piece of equipment that does the job 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 very 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 this change in politeness when migrating from novice to expert levels actually affects the user's perception of the system; at least the transition should be gradual and not too fast. There may also be cultural differences regarding politeness rules.

Usability issues are easy to overlook when investigating and developing adaptive systems. Badly designed adaptivity decreases usability as it may confuse the user; and it is thus essential that the system is consistent in its use of concepts, and manner of speech.

utterances (U = user, S = system)	*relevant parameter values, notes*
U: Read Tim's message.	`ASR output = "read John's message"`
S: There are no messages from John. You have messages from Abdul, Tim, Jaana, and Horst.	`The system misheard the sender's name. Since there are no messages from John in the current mail folder, the system suspects an ASR error and raises the INIT value by one (to two).`
U: Read Tim's message.	`ASR output = "read John's message"`
S: Sorry, I didn't understand. The senders in the current folder Inbox are Abdul, Tim, Jaana, and Horst. Whose message would you like to read?	`The system hears the same thing again, which is a further clue (although tiny) that the ASR conditions are poor. The speech recognizer also reports a low confidence value. Since a problem was suspected in the previous round as well, there is enough evidence for the system to raise the dialogue control (INIT) level again, to three. The system starts to guide the dialogue more closely.`
U: Tim's message please.	`ASR output = "John's message first"`
S: Sorry, I didn't understand. You can refer to a message by its number, its sender's names, and its relative position in the message list. Do you want to choose the message by sender or by number?	`The system can't make sense of the ASR output. INIT is raised to the highest level, and the system walks the user forwards stage by stage.`
U: Sender.	`ASR output = "sender"`
S: The senders in the current folder Inbox are Abdul, Tim, Jaana, and Horst. Please say the sender's name whose message you would like to hear.	
U: Tim.	`ASR output = "tim"`
S: A message from Tim about What where. Hi Bat, did you tape the Beckett yesterday? I missed it, darn. Tim. End of message.	`The message is played out to the user. The command is completed successfully. If no further ASR problems are detected, the INIT value starts to drop.`

Table 2. An example dialogue demonstrating the effect of control and explicitness values on utterance generation.

5 Conclusions

Previous studies concerning user modelling in various interactive applications and dialogue systems have shown the importance of UM in making the interaction with the system more efficient and more enjoyable. In this paper we have focused on the design of an adaptive User Model component and its realization in the AthosMail system. The User Model consists of several components that are meant to support adaptation in interaction, presentation, and search. We provided a more detailed view of the Cooperativity Model which integrates initiative handling and explicitness by combining various parameters and their values in order to provide recommendations to the system about appropriate ways to convey information and control the dialogue. The model is language independent, and it is being integrated into the multilingual AthosMail system.

We have conducted the first user studies to test the Cooperativity model as part of the complete AthosMail system. However, more extensive user studies are needed to evaluate the model's validity and usability in spoken language interfaces. The combined effect of dialogue control and explicitness is complex, and requires careful testing. We also plan to evaluate the system at three different sites using English, Finnish and Swedish users. User tests with the visually impaired are under preparation, too.

Acknowledgements

The research was carried out within the EU-project DUMAS (Dynamic Universal Mobility for Adaptive Speech Interfaces), IST-2000-29452 (http://www.sics.se/dumas). We want to thank all the project participants for cooperation and discussions.

References

1. Chesnais, P., Mucklo, M., Sheena J.A. 1995. The Fishwrap Personalized News System, *IEEE Second International Workshop on Community Networking Integrating Multimedia Services to the Home*. Princeton, NJ, USA.
2. Chin, D. 1989. KNOME: Modeling what the User Knows in UC. In: Kobsa, A., Wahlster, W. (eds.) *User Modeling in Dialogue Systems*. Springer-Verlag Berlin, Heidelberg 74-107
3. Chu-Carroll, J. and Nickerson, J. S. 2000. Evaluating Automatic Dialogue Strategy Adaptation for a Spoken Dialogue System. In *Proceedings of NAACL-00*.
4. Dai, W. and Cohen, R. 2003. Dynamic Personalized TV Recommendation System. *Proceedings of the UM 2003 Workshop on Personalization in Future TV*. Pittsburgh, US, 12-21.
5. Goldberg, D., Nichols, D, Oki, B.M., and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Communications of ACM*, 35(12): 51-60.
6. Jokinen, K. 1996. Goal Formulation Based on Communicative Principles. *Proceedings of the 16th International Conference on Computational Linguistics,* Copenhagen, Denmark, 598-603
7. Jokinen, K., Rissanen, J., Keränen, H., and Kanto, K. 2002. Learning Interaction Patterns for Adaptive User Interfaces. In *Adjunct Proceedings of the 7th ERCIM Workshop User Interfaces for all*, Paris, France, 53-58.
8. Jokinen, K., Kanto, K. Rissanen, J., Sahlgren, M. 2003. Prototype user modelling component. Deliverable D6.3. DUMAS IST-2000-29452.
9. Krahmer, E., Swerts, M., Theune, M., Weegels, M. 1999. Problem Spotting in Human-Machine Interaction. In *Proceedings of Eurospeech '99*. Vol. 3. Budapest, Hungary, 1423-1426.
10. Litman, D. and Pan, S. 1999. Empirically Evaluating an Adaptable Spoken Dialogue System. In *Proceedings of the 7th International Conference on User Modeling*, Banff, Canada, 55-64.
11. Malone,T., Grant,K., Turbak, F., Brobst, S. and Cohen, M. 1987. Intelligent information sharing systems. *Communications of the ACM*, (30):390-402.
12. Moukas, A., and Maes, P. 1998. Amalthaea: An evolving Multi-Agen Information Filtering and Discovery System for the WWW. *Autonomous Agents and Multi-Agent Systems*, 1(1):59-88.
13. Oppermann, R. 1994. Adaptively supported Adaptability. *International Journal of Human-Computer Studies*, 40:455-472.
14. Paris, C. 1993. *User Modelling in Text Generation*, Frances Pinter.
15. Sahlgren, M. 2003. Content-based Adaptivity in Multilingual Dialogue Systems. In *Proceedings of the 14th Nordic Conference of Computational Linguistics*, Reykjavík, Iceland.
16. Smith R. W. and Hipp D. R. 1994. *Spoken Natural Language Dialog Systems - A Practical Approach*. Oxford: Oxford University Press.
17. Smith, R. W. 1993. Effective Spoken Natural Language Dialog Requires Variable Initiative Behavior: An Empirical Study. *Proceedings of the 1993 AAAI Fall Symposium on Human-Computer Collaboration: Reconciling Theory, Synthesizing Practice*.
18. Walker, M., Langkilde, I., Wright, J., Gorin, A., and Litman, D. 2000. Learning to Predict Problematic Situations in a Spoken Dialogue System: Experiments with How May I Help You? In *Proceedings of the 1st Meeting of the NAACLs*. Seattle, Washington, USA, 210-217.