Empirically Grounded Decision-Theoretic Adaptation to Situation-Dependent Resource Limitations

Thorsten Bohnenberger, Boris Brandherm, Barbara Großmann-Hutter, Dominik Heckmann, Frank Wittig

Abstract

This article summarizes research on several interrelated general issues that can arise in the design and development of user modeling systems: the learning and subsequent adaptation of general user models on the basis of empirical data; the modeling of temporally variable properties of users, in particular time pressure and cognitive load; and the user-adaptive planning of interactions under uncertainty. The methods and results are integrated and illustrated with a prototype of a mobile assistance system for travelers in an airport.

1 Introduction

As you browse the articles and reports in this special issue, you will notice that there are many ways in which an interactive system can adapt to its users (cf. also [5]). This article explores the adaptation of an assistance system to a user's *situation-dependent resource limitations*. At the same time, it addresses several fundamental issues that arise with other types of user modeling as well.

Whether a user is able to conduct a spoken dialog with a system successfully depends in part on the appropriateness of the system's utterances with respect to the user's current situation. This dependence holds especially for mobile systems, whose users may suffer from resource limitations induced by the environment. We present for such a mobile system a multilevel approach to achieving adaptivity with respect to the user's *time pressure* and *cognitive load*.

A large international airport is just one domain that creates various situations for which adaptive assistance promises to be helpful. People may appreciate being guided to gates, check-in counters, fast food restaurants, and duty-free shops. They may want to be assisted when using facilities like check-in machines, ticket machines, and credit card phones. Whereas today people rely on posted signs and on preformulated operating instructions, an adaptive assistance system will provide support that is tailored to the current needs of an individual user.

After an introductory example of adaptive assistance and an overview of the READY[1] system, we will survey four focal research areas within the READY project: (1) empirical studies concerning users' resource limitations; (2) user modeling with dynamic Bayesian networks for making inferences about resource limitations of the user; (3) the learning of Bayesian networks from empirical data; and (4) decision-theoretic methods for planning the system's interaction with the user. Finally, we will show how results from all of these areas enable the prototype system to deal with the introductory example.

[1] **RE**source-**A**daptive **D**ialog s**Y**stem.

2 Adaptive Assistance: Example

Passenger $\mathcal{P}$'s flight is delayed. Since her business partner is going to pick her up at the destination airport, $\mathcal{P}$ wants to make a phone call to inform him about her late arrival. She finds a phone which requires the use of a credit card. Since $\mathcal{P}$ has not used this particular type of credit card phone before, she consults her mobile PDA-based airport assistance system $\mathcal{S}$. The help system instructs $\mathcal{P}$ step by step about how to operate the phone. $\mathcal{P}$ executes $\mathcal{S}$'s detailed instructions without difficulty.

Passenger $\mathcal{Q}$ arrives late at the airport. Having overslept, he rushed out of his house and forgot to turn off his electric heater. There are just a few minutes left until boarding time, but since $\mathcal{Q}$ is worried about a possible fire in his flat, he wants to call his neighbor before boarding. The only phone $\mathcal{Q}$ can find requires the use of a credit card—and $\mathcal{Q}$ has likewise not yet used this type of credit card phone. His mobile assistance system $\mathcal{S}$ adapts to his needs, presenting the information needed to operate the phone as concisely as possible: It gives several simple instructions in one turn, but at the same time it takes care not to overload $\mathcal{Q}$ with information at any one moment. $\mathcal{Q}$ makes his call quickly and later reaches the gate just in time.

Operating a credit card phone requires a sequence of actions. $\mathcal{S}$ could, for example give the following eight instructions:

1. "Get out your credit card."
2. "Lift the receiver."
3. "Dial 0."
4. "After the tone, dial 9."
5. "After the tone, enter your credit card number."
6. "Enter two digits for the month of expiration."
7. "Enter two digits for the year of expiration."
8. "After the tone, dial the desired number."

There are many ways of presenting this sequence of instructions, ranging from (a) giving just one instruction at a time to (b) giving all of them before the user $\mathcal{U}$ starts to execute the first instruction. With the latter approach, $\mathcal{U}$ is likely to forget some of the instructions before executing them, either performing the task incorrectly or requiring $\mathcal{S}$ to repeat instructions. On the one hand, the former approach can take a long time, since $\mathcal{U}$ has to confirm the execution of each instruction (e.g., by saying "OK") before $\mathcal{S}$ can give the next one. In general, some strategy lying between these two extremes will be best; but the most promising strategy should be determined on the basis of the current situation and user model.

3 Overview of READY

Figure 1 shows the components of READY and the ways in which they interact. The components are represented in a

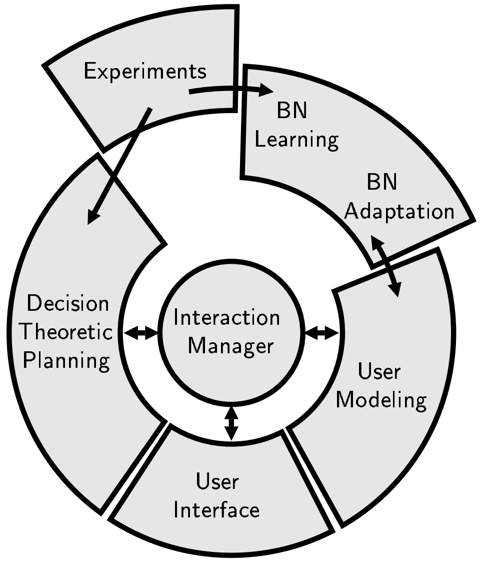

Figure 1. *Components of* READY.

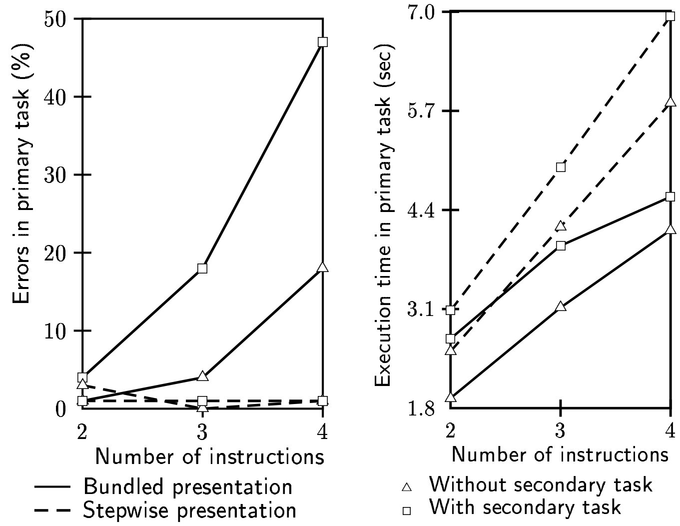

Figure 2. *Main results of Experiment 1: Percentage of errors (left) and execution times (right) as a function of the number of instructions, the presentation mode, and the presence of a secondary task.*

cake diagram. The slices within the cake represent the system's on-line components, while those sticking out represent components that are active in a preceding off-line initialization phase. In Experiments, we gather information about behaviors and actions of users through experiments. These data are used to parametrize the decision-theoretic planning process, as well as to learn initial Bayesian networks ("BNs") for the modeling of individual users.

A user interacts with READY via the User Interface, which passes the input to the Interaction Manager. The Interaction Manager performs a first rough action and response planning and provides the User Modeling component with information about how the user made his request. Input information—that is, what the user wants to do or to know—is transferred to Decision-Theoretic Planning. Communication between User Modeling and Decision-Theoretic Planning is handled by the Interaction Manager. In particular, Decision-Theoretic Planning receives from User Modeling information about the user's current state. The BNs for user modeling are adapted after each (or several) user transactions by BN Adaptation, in a semi-on-line way. After the assistance information has been put together by Decision-Theoretic Planning, it is presented to the user via the User Interface.

4 Empirical Grounding in Experiments

Empirical studies with system users usually serve the goal of evaluating an existing system. By contrast, the goal of the experiments to be summarized here was to establish a quantitative foundation for suitable modeling and adaptation for the future system. The experiments concerned two central topics, the generation of system output and the analysis of user input, respectively.

4.1 Experiment 1: Stepwise vs. Bundled Instructions

The focus of interest here is to decide when it is better to present instructions in a *stepwise* mode (i.e., one at a time) or in a *bundled* mode (i.e., all at once).

Twenty-four subjects were presented with a rather abstract user interface: At the top there was a "lamp" that occasionally turned red or green; the main part of the screen contained 6 sets of 4 "radio buttons" of the sort typically used in dialog boxes for the setting of system parameters (for details see [6]). The primary task of the subjects involved clicking on radio buttons according to the system's spoken instructions. Each sequence of instructions comprised 2, 3, or 4 instructions, which were presented in a stepwise or a bundled way. The blinking lamp was used as a secondary task that created additional cognitive load: Subjects had to press the space bar whenever the same color appeared twice in succession.

As is shown in the right-hand side of Figure 2, the bundled presentation yields an increasing speed advantage with larger numbers of instructions. But this advantage is associated with an increasing probability of errors, especially when the user is distracted by a secondary task (see the left-hand side of the figure). In addition to illustrating these general relationships, the experiment yields fine-grained quantitative data about execution times and error probabilities that can serve as a basis for user modeling and decision-theoretic planning.

4.2 Experiment 2: Symptoms of Resource Limitations in the User's Speech

This experiment was designed to lay the foundations for a system that can recognize resource limitations of a user on the basis of $\mathcal{U}$'s speech.

The 32 subjects were presented with a two-dimensional virtual airport scenario, in which they were to act as if they were walking through the airport using an assistance system (for fur-

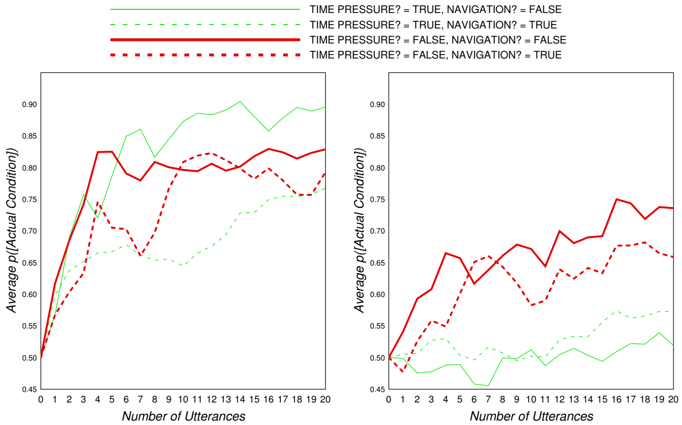

Figure 3. *Accuracy of a learned Bayesian network in recognizing whether a subject in Experiment 2 is under time pressure (left) or is navigating through the virtual airport (right) on the basis of the features of the subject's speech in a given number of utterances.*

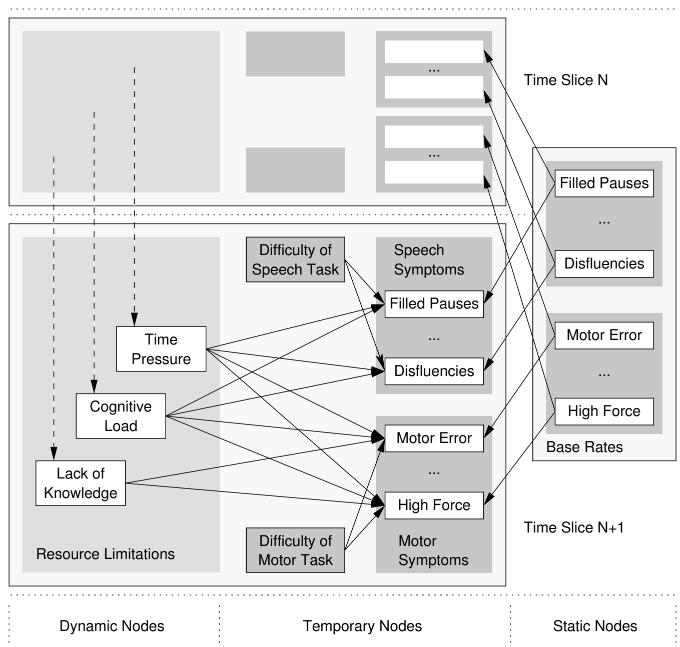

Figure 4. *Overview of a dynamic Bayesian network for the recognition of a user's resource limitations.*

ther information see [13]). In a condition with especially high cognitive load, they had to visit several locations along the way. While they were navigating through the screen (using the arrow keys on the keyboard), a picture was presented in a corner of the screen. This picture conveyed the topic of a question that the subject was to formulate (e.g., "My suitcase is broken. Where can I buy a new one?"). The subject's utterances were recorded for later analysis.

In a condition with lower cognitive load, subjects had to ask questions in a similar way, but they did not have to navigate through the airport. A further experimental variable concerned time pressure: whether subjects were instructed to complete their question as quickly as possible or to formulate it especially clearly.

A conventional analysis of the results showed that many features of the subjects' speech were in fact affected by the experimental manipulations. These features range from the articulation rate (number of syllables per second) to the presence of disfluencies such as interrupted sentences.

The data were also used for the learning of Bayesian networks (see Section 6) that could recognize resource limitations on the basis of a user's speech. As can be seen in Figure 3, in this specific experimental environment the presence of time pressure can be recognized quite well. Whether a user is navigating or not can be recognized reasonably well only when the user is not under time pressure, since time pressure leads to shorter, less revealing utterances.

5 User Modeling With Dynamic Bayesian Networks

Bayesian Networks ([14, 8]) have become increasingly popular as an inference technique for user modeling (see e.g., [4]). In a BN (see the example shown in Figure 4), directed arcs depict uncertain (often causal) relationships among variables. Tables of conditional probabilities that represent the nature and strength of these relationships are associated with the arcs.

There are mainly two properties of BNs that make them well suited for application in a user modeling context: (a) the possibility of interpreting the networks' arcs in terms of relationships among domain variables, which is helpful in both the construction and the explanation of the user model; and (b) their ability to handle uncertainty in the domain under consideration, which is typically pervasive where inferences about a user are involved.

Dynamic Bayesian networks (DBNs) overcome the static nature of standard BNs. The DBN shown in Figure 4 illustrates how a DBN includes a sequence of *time slices* and various types of nodes. DBNs are especially suitable when user modeling concerns temporally variable properties of users.

READY is to adapt to a user's resource limitations of time pressure and cognitive load largely on the basis of two types of evidence in $\mathcal{U}$'s behavior: (a) features of $\mathcal{U}$'s speech and (b) properties of $\mathcal{U}$'s *manual input behavior*, such as button presses and taps on the screen with a stylus. When new evidence arrives, a new time slice is added to the DBN.

The temporal variability of time pressure and cognitive load is taken into account in the DBN in that each dynamic node that represents one of these properties in a given time slice has only a probabilistic relationship with the corresponding instance in the next time slice (these relationships are shown in Figure 4 as dashed arcs). By contrast, the *base rates* that represent stable properties of $\mathcal{U}$'s behavior, such as a general tendency to use High Force when tapping the screen, are realized as static nodes that have the same impact on each time slice. The base rates can be saved and reused by $\mathcal{S}$ for this particular user.

A BN for the interpretation of speech symptoms was learned on the basis of Experiment 2 (Section 4.2), while the BN for the interpretation of features of manual input behavior was constructed on the basis of a literature study ([11]). The combination of these two BNs allows $\mathcal{S}$ to make inferences about $\mathcal{U}$ on the basis of (simulated) multimodal input.

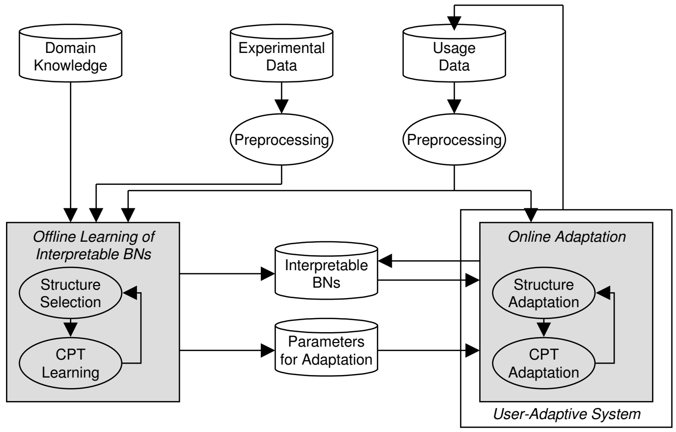

Figure 5. *Overview of BN learning procedures. (Cylinders represent repositories of data and BNs, ellipses stand for algorithmic procedures, and boxes symbolize larger conceptual entities. The flow of information between these components is depicted by directed arcs.).*

Manual input behavior is mapped onto relatively abstract *motor symptoms* (see [11] for details). For example, "tapping very hard on the screen" is mapped onto High Force. In connection with motor symptoms, it is also important to take $\mathcal{U}$'s prior knowledge into account. Otherwise, $\mathcal{U}$'s lack of knowledge about how to operate the assistance system might be misinterpreted in terms of time pressure or cognitive load.

After the instantiation of the relevant evidence nodes, the DBN is evaluated, and new estimates of $\mathcal{U}$'s time pressure and cognitive load can be sent to the decision-theoretic planning module.

6 Learning Bayesian Networks

Two well-known problems are those of how to initialize the two parts of BN user model—i.e., which initial conditional probabilities and which initial structure to use. Often the second problem is relatively manageable: It may be fairly easy to identify an adequate structure on the basis of theoretical considerations regarding the causal relations among entities of interest in the domain. Although some previous work in user modeling has applied standard BN learning techniques to these problems (see e.g., [9, 12]), to our knowledge no previous work has addressed the specific issues that arise when these methods are applied in a user modeling context. In the READY project, we have developed tools for an integrated approach to learning and adapting BNs for user-adaptive systems, as is shown in Figure 5. This methodology is flexible with regard to several *dimensions* that will be addressed in the rest of this section (although the current READY prototype does not make use of all aspects of the methodology).

Off-line learning and on-line adaptation The first two dimensions (shown as gray boxes in Figure 5) are *off-line learning* and *on-line adaptation*. During the off-line phase, a general user model is learned on the basis of data concerning previous system users or data acquired by experiments such as those described in Section 4. This model is in turn used as a starting point for the interaction with a particular new user: The initial general model is adapted to the individual current user, and it can be saved after the interaction for future application to this user.

The off-line learning procedure yields not only a general model but also parametric information concerning adaptation to individual users. The general idea is that some parts of the learned general user model should be adapted to an individual user faster than others. For example, Experiment 1 (Section 4.1) showed that subjects performed similarly with regard to their error rates but differed rather widely in terms of their average execution times. Accordingly, it makes sense to adapt the part of the model that is related to execution time to the individual user faster than the part related to errors. A formal exposition of this idea and a comparison of alternative methods of adaptation to individual users can be found in [7].

Experimental data and usage data Two further dimensions concern the nature of the data that are available. We distinguish between experimental data and usage data (see the upper part of Figure 5). Experimental data are collected in controlled environments (cf. the experiments presented in Section 4). Usage data are collected during real interactions between users and the system. Usage data often include more missing data, and rare situations may not be represented, while experimental data in general do not represent the real usage situations very accurately. Often, a combination of the two types of data is available. Because of our off-line/on-line approach we can handle this combination, for example, by (a) learning a general model of the relationships between resource limitations and speech symptoms on the basis of experimental data and then (b) adapting the model using usage data concerning speech symptoms of the individual user.

Learning the conditional probabilities and structure of a BN Since a BN comprises two components, the learning and adaptation tasks are also two-dimensional: (a) learning the conditional probability tables and (b) learning the BN's structure. In a user modeling context, we often have to deal with sparse data, but in most cases we have additional domain knowledge available. Our approach involves introducing such domain knowledge into the learning procedures so as to improve the accuracy of the learned BNs, especially when the data are sparse (see the upper left-hand part of Figure 5). For the learning of the conditional probabilities, we developed a new method in which the user model's designer can label the BN's arcs with plus (+) or minus (−) signs, respectively, before learning takes place. These signs are intended to reflect monotonic relationships between variables; for example, the hypothesis that distraction increases the likelihood of user errors. This additional knowledge guides the learning algorithm through the search space by in effect penalizing solutions that violate (some of) the specified *qualitative constraints* (see [15] for details). For exploiting prior knowledge about the structure of BNs, various approaches are available, including straightforward ones such as specifying in advance that the learned structure must contain particular arcs.

Degree of interpretability Many researchers have argued that the interpretability and transparency of user models are key factors that determine the acceptance of user-adaptive systems (see, e.g., [5]). In particular, a system that is able to explain its recommendations or decisions may be better accepted by users. This goal of interpretability is one motivation for allowing the incorporation of prior knowledge in the learning of BNs: Even if prior knowledge does not make a BN more accurate, it may be worth incorporating because of the way in which it increases interpretability.

7 Decision-Theoretic Planning

The dynamic Bayesian networks described in the previous two sections give us a powerful means for making inferences about $\mathcal{U}$'s current situation. But how can $\mathcal{S}$ use this information to adapt to $\mathcal{U}$'s needs? For example, how can $\mathcal{S}$ present the instructions for operating the credit card phone in the way which is most appropriate to the situation at hand? Or in terms of decision theory: how can $\mathcal{S}$ present the instructions in such a way that they promise the maximum expected utility for $\mathcal{U}$? To achieve this, $\mathcal{S}$ needs to plan the dialog with the user several steps ahead. In effect, $\mathcal{S}$ should consider the likely outcomes of all possible instruction strategies, in order to be able to choose the most promising one.

7.1 Modeling of Dialogs With Markov Decision Processes

A decision-theoretic approach has already been applied to the learning of dialog strategies in spoken dialog systems (see, e.g., [10]). The underlying representation—a Markov decision process (MDP, see e.g., [1, 3])—can also be used for planning in stochastic environments. We have already applied decision-theoretic planning, for example, to generate sequences of location-aware and situation-dependent navigation recommendations (see [2]). In the context of an adaptive assistance dialog (as well as in the navigation scenario), decision-theoretic planning allows $\mathcal{S}$ to consider uncertainties about how successfully the user will execute instructions. Markov decision processes are particularly well suited to the handling of trade-offs between competing goals—in our example, the trade-off between completing instructions without error versus completing them as quickly as possible.

We make the following basic assumption, which is supported by the empirical data of Experiment 1 (Section 4.1): the probability that $\mathcal{U}$ executes an instruction incorrectly depends in part on how long $\mathcal{U}$ is required to remember the instruction before executing it. Moreover, some steps are inherently more error-prone than others, and some require a longer time for execution than others (e.g., dialing a single digit vs. entering a complete credit card number).

Within the framework of Markov decision processes, the possible courses of the dialog between $\mathcal{S}$ and $\mathcal{U}$ are modeled in terms of states connected by stochastic transitions. Each state is defined by a set of features; for example, one feature is the number of instructions that $\mathcal{U}$ needs to keep in working memory at a given point in the dialog (see [6] for details). Each transition has a certain cost—the times for the system to give an instruction or for the user to execute an instruction—and a probability for the successful completion of the action in question. Finally, each goal state of the MDP is associated with a reward, which is to be weighed against the costs of the actions required to reach the goal state.

Through the application of a standard algorithm such as value iteration (see, e.g., [3]) to an MDP, a *policy* can be derived: a mapping from (dialog) states to actions (utterances). On the basis of the policy, $\mathcal{S}$ can always determine the optimal utterance to make in the dialog state in which it finds itself—as long as the dialog proceeds in a way that is consistent with the underlying MDP.

7.2 The Influence of the User Model

When the user modeling component has inferred a high degree of time pressure in $\mathcal{U}$, what impact should this inference have on the dialog planning process? One way of modeling time pressure is to assign a high cost to each second of time that is required by an action performed by $\mathcal{S}$ or $\mathcal{U}$. Where $\mathcal{S}$ is planning sequences of instructions, increasing the cost of time causes $\mathcal{S}$ to derive policies that involve relatively large bundles of instructions. This is the general result that one would expect intuitively, since such policies tend to lead to faster task completion.

When the user modeling component has inferred a high degree of cognitive load in $\mathcal{U}$, the probabilities within the MDP of correct executions of actions by $\mathcal{U}$ in the various states should be lowered. (For example, the data of Experiment 1 made it possible to estimate the relevant transition probabilities for two different levels of cognitive load.) As one would expect, the policies that result in this case tend to be more careful, involving smaller bundles of instructions or even stepwise presentation.

8 Example Dialog

8.1 The User Interface

The example dialog will be easier to read after a brief introduction to the user interface of the READY prototype. The user interface is realized on a workstation display as a simulation of a handheld device. So that we can simulate the use of a broad range of types of input and output, the prototype does not include components such as real speech recognizers and synthesizers; instead, all input and output is specified with the typical modalities of graphical workstations (e.g., menus and graphical and textual displays).

In the upper left-hand corner of the Graphics / Text widget in Figure 6, the small map shows the content of the simulated PDA screen. The manual input to the PDA is simulated with a description of the content of the input and the way in which it is entered. For example, in the upper right-hand corner of the Graphics / Text widget, it has been specified that $\mathcal{U}$ was presented with a Map, in which she Clicked on the Phone. Below, it has been specified (among other things), that $\mathcal{U}$ at first clicked next to the target, though quite Close to it.

Speech input is realized analogously (see the Speech widget on the right in Figure 6). In our example, the utterance of $\mathcal{U}$ is specified as having the content I'd like to use this... and an Articulation rate of 4-8 syll/sec.

Figure 6. *Screen shot of the user interface of the* READY *prototype.*

The manual and the speech input are two parts of a multimodal utterance expressing $\mathcal{U}$'s desire to use a particular phone. $\mathcal{S}$'s multimodal output is displayed in an analogous fashion via the interface.

The two small panels shown at the top of Figure 6 display aspects of the system's internal processing.

8.2 Assisting Passenger $\mathcal{Q}$

We now return to the example of passenger $\mathcal{Q}$ that was introduced in Section 2. $\mathcal{Q}$ is in a hurry when he requests help from the airport assistance system. His articulation rate is high, and on his first attempt he does not click accurately on the phone icon displayed on his PDA's display, clicking slightly next to it instead. Instantiation of the nodes corresponding to these two symptoms in the dynamic Bayesian network causes $\mathcal{S}$ to infer a relatively high level of time pressure. The updated user model is now used to parameterize the decision-theoretic planning process. A dialog policy is computed which determines the $\mathcal{S}$'s next four instructions. $\mathcal{S}$ starts with a single instruction:

- $\mathcal{S}$: "Get out your credit card."
- $\mathcal{S}$ waits for feedback (e.g., $\mathcal{Q}$ might say something like "OK, I've got my credit card ready now . . . ").

$\mathcal{S}$ then bundles the next two instructions:

- $\mathcal{S}$: "Lift the receiver."
- $\mathcal{S}$: "Dial 0."
- $\mathcal{S}$ waits for feedback.

In principle, $\mathcal{S}$ could have given the first three instructions in one bundle. Many factors contributed to the decision to give the first one separately; one of them is the relatively long duration of the action of getting out the credit card, which decreases the likelihood that any subsequent instructions within the same bundle would be remembered accurately.

Although $\mathcal{Q}$'s articulation rate has meanwhile decreased to a moderate level, $\mathcal{S}$'s estimate of $\mathcal{U}$'s time pressure is still relatively high, since the DBNs incorporate the assumption that the degree of time pressure is unlikely to change suddenly at any given moment. There is still no evidence for an unusual level of cognitive load.

$\mathcal{S}$ now bundles the next two instructions:

- $\mathcal{S}$: "After the tone, dial 9."
- $\mathcal{S}$: "After the tone, enter your credit card number."
- $\mathcal{S}$ waits for feedback.

These instructions would not be bundled if the actions were to be performed in the reverse order: After the complex action of entering the credit card number, the probability of remembering any other instruction would be relatively low.

$\mathcal{S}$ gives the last three instructions stepwise:

- $\mathcal{S}$: "Enter two digits for the month of expiration."

- $\mathcal{S}$ waits for feedback.
- $\mathcal{S}$: "Enter two digits for the year of expiration."
- $\mathcal{S}$ waits for feedback.
- $\mathcal{S}$: "After the tone, dial the desired number."
- $\mathcal{S}$ waits for feedback.

This choice of stepwise presentation is influenced in part by the working memory demands of the actions involved. For example, entering information about the expiration date is classified as being more demanding than lifting the receiver).

9 Relation to Other User Modeling Research

To a greater extent than most other user modeling research, the work summarized here represents basic research concerning general issues that can arise in the development of many different types of user modeling system, such as: How can general user models be learned on the basis of empirical data, and how can they subsequently be adapted to individual users and different situations? How can diverse sources of evidence be exploited in the modeling of temporally variable properties of users? How can resource limitations of time pressure and cognitive load be conceptualized in a way that is simple enough for on-line user modeling yet realistic enough to give rise to appropriate system adaptations? How can a system that offers recommendations plan several steps ahead, taking into account uncertainty about how the user will respond to its recommendations?

The ultimate value of this research will be determined in part through efforts to adapt its results for use in more application-oriented systems (for example, at our neighboring German Research Center for Artificial Intelligence).

Acknowledgments

The research summarized in this paper was supported by the "Deutsche Forschungsgemeinschaft" (DFG) in its Collaborative Research Center 378 on Resource-Adaptive Cognitive Processes, Project READY. The two experiments summarized in Section 4 were designed and conducted in collaboration with Ralf Rummer of the Department of Psychology of Saarland University, in the context of the master's theses of Leonie March and Christian Müller, respectively. We thank Anthony Jameson for his support.

References

[1] Jim Blythe. Decision-theoretic planning. *AI Magazine*, 20(2):37–54, 1999.

[2] Thorsten Bohnenberger and Anthony Jameson. When policies are better than plans: Decision-theoretic planning of recommendation sequences. In James Lester, editor, *IUI 2001: International Conference on Intelligent User Interfaces*, pages 21–24. ACM, New York, 2001.

[3] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[4] Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. The Lumière project: Bayesian user modeling for inferring the goals and needs of software users. In Gregory F. Cooper and Serafin Moral, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Fourteenth Conference*, pages 256–265. Morgan Kaufmann, San Francisco, 1998.

[5] Anthony Jameson. Adaptive interfaces and agents. In Julie A. Jacko and Andrew Sears, editors, *Handbook of Human-Computer Interaction in Interactive Systems*. Erlbaum, Mahwah, NJ, 2002. In press.

[6] Anthony Jameson, Barbara Großmann-Hutter, Leonie March, Ralf Rummer, Thorsten Bohnenberger, and Frank Wittig. When actions have consequences: Empirically based decision making for intelligent user interfaces. *Knowledge-Based Systems*, 14:75–92, 2001.

[7] Anthony Jameson and Frank Wittig. Leveraging data about users in general in the learning of individual user models. In Bernhard Nebel, editor, *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 1185–1192. Morgan Kaufmann, San Francisco, CA, 2001.

[8] Finn V. Jensen. *Bayesian Networks and Decision Graphs*. Springer, New York, 2001.

[9] Tessa Lau and Eric Horvitz. Patterns of search: Analyzing and modeling Web query dynamics. In Judy Kay, editor, *UM99, User Modeling: Proceedings of the Seventh International Conference*, pages 119–128. Springer Wien New York, Vienna, 1999.

[10] Esther Levin, Roberto Pieraccini, and Wieland Eckert. A stochastic model of human-machine interaction for learning dialogue strategies. *IEEE Transactions on Speech and Audio Processing*, 8:11–23, 2000.

[11] Kristin Lindmark. Interpreting symptoms of cognitive load and time pressure in manual input. Master's thesis, Department of Computer Science, Saarland University, Germany, 2000.

[12] Michael Mayo and Antonija Mitrovic. Optimising ITS behaviour with Bayesian networks and decision theory. *International Journal of Artificial Intelligence in Education*, 12, 2001.

[13] Christian Müller, Barbara Großmann-Hutter, Anthony Jameson, Ralf Rummer, and Frank Wittig. Recognizing time pressure and cognitive load on the basis of speech: An experimental study. In Mathias Bauer, Piotr Gmytrasiewicz, and Julita Vassileva, editors, *UM2001, User Modeling: Proceedings of the Eighth International Conference*. Springer, Berlin, 2001.

[14] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

Figure 7. *Thorsten Bohnenberger.*

Figure 8. *Boris Brandherm.*

[15] Frank Wittig and Anthony Jameson. Exploiting qualitative knowledge in the learning of conditional probabilities of Bayesian networks. In Craig Boutilier and Moisés Goldszmidt, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference*, pages 644–652. Morgan Kaufmann, San Francisco, 2000.

10 Authors' Vitae

The authors all hold master's degrees in computer science, and they are currently (or recently have been) affiliated with the READY project. Their research focuses on the following areas, respectively: Thorsten Bohnenberger: decision-theoretic planning for intelligent user interfaces; Boris Brandherm: user modeling and dynamic Bayesian networks; Barbara Grossmann-Hutter: experimental research with users as a basis for intelligent user interfaces; Dominik Heckmann: ubiquitous user modeling for situated interaction; Frank Wittig: machine learning techniques for Bayesian networks in the context of user modeling.

Figure 9. *Barbara Großmann-Hutter.*

Figure 10. *Dominik Heckmann.*

Figure 11. *Frank Wittig.*