

# Machine Learning and Knowledge Representation in the LaboUr Approach to User Modeling

Wolfgang Pohl and Achim Nick\*

GMD FIT, HCI Research Department, Sankt Augustin, Germany

**Abstract.** In early user-adaptive systems, the use of knowledge representation methods for user modeling has often been the focus of research. In recent years, however, the application of machine learning techniques to control user-adapted interaction has become popular. In this paper, we present and compare adaptive systems that use either knowledge representation or machine learning for user modeling. Based on this comparison, several dimensions are identified that can be used to distinguish both approaches, but also to characterize user modeling systems in general. The LaboUr (Learning about the User) approach to user modeling is presented which attempts to take an ideal position in the resulting multi-dimensional space by combining machine learning and knowledge representation techniques. Finally, an implementation of LaboUr ideas into the information server ELFI is sketched.

## 1 Introduction

While striving to achieve user-adapted interaction, user modeling researchers have often made use of *knowledge representation* (KR) techniques: A representation formalism is used to maintain assumptions about a user within a knowledge base. The reasoning mechanisms of the formalism are used to extend this user model. The (explicit or implicit) contents of the knowledge base can be accessed by an application and be used to support its adaptivity decisions.

KR-based user modeling originated in research on (natural-language) dialog systems (Wahlster and Kobsa, 1989). User-adapted interaction, however, has long been a concern also in the area of intelligent user interfaces. Systems developed in this area typically observe system usage and determine usage regularities to form a *usage profile* (Krogsæter et al., 1994) that is to support adaptivity decisions. This procedure can be regarded as “learning from observation”. Hence, it is no surprise that *machine learning* (ML) techniques have often been used to form usage profiles and support user-adapted interaction. Note that ML-based adaptivity involves user modeling: a usage profile provides an information source separate from other system knowledge and hence roughly fits the definition of a user model by Wahlster and Kobsa (1989).

In the next section, we will present examples of both KR-based and ML-based user modeling. We will examine them in order to identify a set of dimensions useful for clarifying the differences between both approaches and for characterizing user modeling systems in general. In Section 3 we describe LaboUr (Learning about the User), an approach to user modeling that attempts to combine the advantages of KR-based and ML-based user modeling. Finally, we will present

---

\* This work has been supported by DFG (German Science Foundation), grant No. Ko 1044/12, DFN (German Research Network) and the German Federal Ministry of Education, Science, Research, and Technology. We thank the anonymous reviewers for their very helpful remarks.

ELFI, a system that is in real-world use as a server for information on research funding. LaboUr ideas have been implemented into ELFI in order to realize user-adaptive features.

## 2 Dimensions of User Modeling Systems

Several authors have spent effort on identifying categories and dimensions for characterizing several aspects of user-adaptive systems. Rich (1983) found three characterizing dimensions for user models, while Kobsa (1989) identified categories for user model contents. We will take a look at properties of user modeling components or systems from a technical point of view.

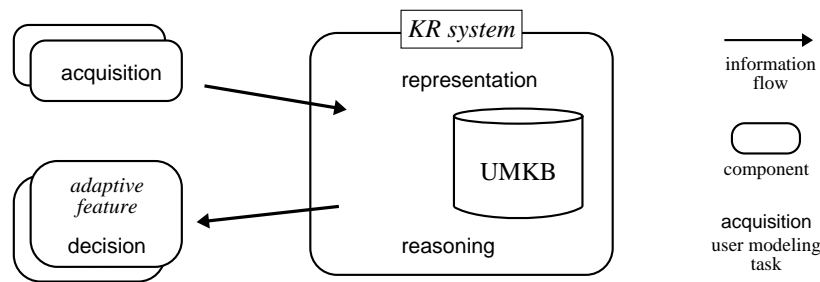
### 2.1 Knowledge Representation for User Modeling

Traditional user modeling systems often make use of knowledge representation techniques. KR formalisms offer facilities for maintaining knowledge bases (using representation formalisms) and for reasoning (using the inference procedures of representation formalisms). For user modeling, these facilities are typically employed as follows: Assumptions about individual characteristics of the user are maintained in a knowledge base, using a representation formalism. Since this knowledge base may additionally contain system knowledge about the application domain or meta-knowledge for inferring additional assumptions about the user from her current model (including stereotypes), it has been called user modeling knowledge base (UMKB, Pohl, 1998). If available, inference procedures of the representation formalism or meta-level inferences can be used to expand the user model. The use of (particularly logic-based) knowledge representation methods in user modeling systems has been analyzed in detail by Pohl (1998).

For a concrete example, we take the adaptive hypertext system KN-AHS (Kobsa et al., 1994), which makes use of the KR methods offered by the user modeling shell system BGP-MS (Kobsa and Pohl, 1995) to maintain assumptions about user knowledge. *Acquisition* of assumptions is based on how the user interacts with the hypertext hotwords and is controlled by heuristics like: "If the user requests an explanation [...] for a hotword, then he is [...] unfamiliar with this hotword" (cf. Kobsa et al., 1994, p.103). Assumptions are communicated to BGP-MS, which *represents* both assumptions and domain knowledge using a concept formalism, with one concept for each hotword. Once being added to the user model, an assumption may trigger meta-level *reasoning* that is based on concept relationships represented as domain knowledge in the UMKB. KN-AHS accesses both explicit and implicit user model contents to make its adaptivity *decisions*, e.g., about how to adapt the contents of hypertext nodes.

In this example, we have identified the following user modeling tasks: acquisition, representation, reasoning, and decision. Corresponding facilities can be found in many user modeling systems. Note that user model reasoning as described above has been called *secondary acquisition* by Pohl (1998), because additional assumptions are derived from existing ones. However, reasoning must be distinguished from acquisition, since it can also be used for other purposes like, e.g., consistency maintenance. Figure 1 illustrates the application of KR methods to user modeling. Acquisition and decision are performed outside the KR system, which is responsible for representation and reasoning.

Several other issues are typical of systems that use KR for user modeling. First, the separate acquisition components often employ procedures or rules which are triggered by one or few observations to construct an assumption about the user that is to be entered into the UMKB. Such



**Figure 1.** Using a knowledge representation system for user-adapted interaction.

an acquisition process is not incremental, i.e., it does not take observation history into account. This can lead to conflicts in the user model; the KR system needs to implement truth maintenance techniques to resolve these conflicts (Brajnik and Tasso, 1994, Paiva and Self, 1995). Second, KR-based user models mostly contain assumptions which are related to mental notions like knowledge, belief, goals, and interests and have been called *mentalistic* by Pohl (1997).

## 2.2 Machine Learning for User Modeling

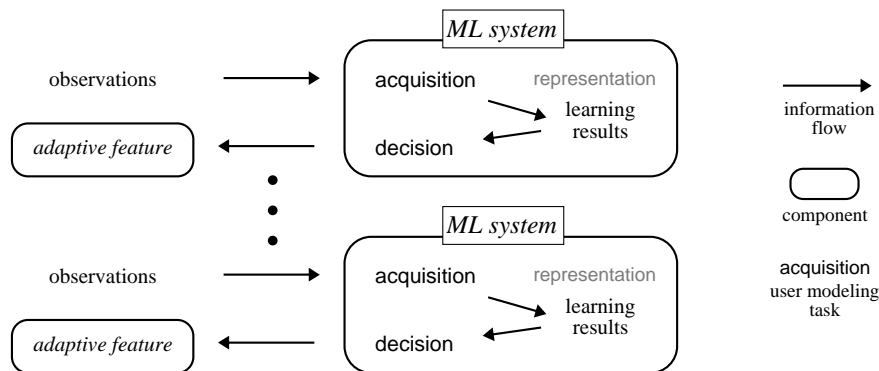
In the early Nineties, a separate strand of research on user-adapted interaction developed in the area of intelligent user interfaces, mainly apart from the user modeling community. As an early protagonist, the system Flexcel (Krogsæter et al., 1994) discovered frequently occurring parameter settings of user commands and suggested to the user to introduce key shortcuts or menu entries for later use of the same settings. Almost at the same time, “interface agents” and “personal assistants” were introduced: Both Kozierok and Maes (1993) and Mitchell et al. (1994) describe software assistants for scheduling meetings. These employ machine learning methods (memory-based learning and decision tree induction, resp.) to acquire assumptions about individual habits of arranging meetings.

More recently, a quite large number of systems using machine learning for personalized information filtering have been described in the literature, like Syskill&Webert (Pazzani and Billsus, 1997), Letizia (Lieberman, 1995), or Amalthea (Moukas, 1996). Let us have a closer look at Syskill&Webert. This system classifies Web pages (within one domain, e.g., biomedicine) as “hot” or “cold” according to user interests. It learns from initial classifications made by the user to make personalized recommendations. For each of a given set of words relevant to the domain, the system determines the two probabilities of the presence of the word in either “hot” or “cold” documents. These probabilities are the results of the learning process of Syskill&Webert. They are used by a Bayes classifier to determine the probabilities of a new document belonging to either category. If the probability for “hot” is greater than that for “cold”, Syskill&Webert makes a positive annotation, else a negative annotation to the document.

Syskill&Webert stores learning results as triplets of the following form:

$$\text{word} \quad p(\text{word present} | \text{hot}) \quad p(\text{word present} | \text{cold})$$

An instance of this scheme in the biomedicine domain is *genome .6 .3*. The stored set of triplets is called “user profile” by Pazzani and Billsus (1997). The format of profile entries is determined



**Figure 2.** Using machine learning for user-adapted interaction.

by the Bayes classifier, which needs the two probabilities for classification; different learning and decision procedures would require different formats.

In general, machine learning methods process training input and offer support for decision (mainly classification) problems based on this input. Hence, ML-based user-adaptive systems work quite differently from KR-based ones. Instead of a knowledge base, learning results are the central source of information about the user. Observations of user behavior (e.g., reactions to meeting proposals or document ratings) are used to form training examples. Learning components do *acquisition* by running their algorithms on these examples. *Representation* is implicit: Formats of learning results are specific to the learning algorithm used (decision trees, probabilities, etc.) which makes them difficult to be reused for other purposes. Due to the lack of an independent representation formalism, there is no further *reasoning* based on already acquired data. However, *decisions* are directly supported, as we have seen for Syskill&Webert. Also the meeting scheduling assistants let their learning components predict the user’s reaction to new meeting proposals and use this prediction for their individualized suggestions.

Figure 2 illustrates the use of machine learning for user-adapted interaction. “Representation” is grayed to indicate that representation is implicit in our terms. The figure also visualizes what we have discussed above: Learning results typically serve one specific decision process. For different adaptive features, different learning processes have to be installed.<sup>1</sup>

But there are not only differences in the handling of user modeling tasks. In the previous section, we observed that in systems with KR-based user modeling acquisition is non-incremental and user models are mostly mentalistic. For ML-based systems, the situation is again different. First, acquisition is incremental, i.e., it takes the history of interactions into account by processing a set of training examples, either one by one or all at once.<sup>2</sup> Hence, learning results (i.e., the user model of ML-based systems) are revised steadily; there is no need for special revision mechanisms. Second, as already stated, often the “user model” of ML-based systems is a us-

<sup>1</sup> However, it is possible that one adaptive feature can be exploited by several applications. E.g., text classification can be applied to e-mails, news articles, and Web pages.

<sup>2</sup> In the latter case, a learning method is called “non-incremental” from a technical point of view, but can be used for incremental acquisition if new observations are processed together with old ones.

age profile, i.e., it carries behavior-related information about the user. In the case of information filtering systems, however, learning results often indicate user interests in specific information content and can be regarded as (implicitly represented) mentalistic assumptions.

### 2.3 Characterizing User Modeling Systems

The discussion about the application of KR and ML methods to user modeling led us to a set of dimensions that can be used to compare both approaches but also to characterize user modeling systems in general. At first, we intended to take the mentioned user modeling tasks as dimensions, each with a binary scale of “supported” vs “not supported” (by the core user modeling system). However, this set of dimensions was not perfectly orthogonal: Reasoning about user model contents depends on the explicit representation of these contents. Therefore, we suggest a slightly different set of characterizing dimensions for user modeling systems. Some of them refer to user modeling tasks, while others cover other issues.

**input** Does the user modeling system accept *observations* of user behavior, *assumptions* (statements about the user, perhaps already formulated in the internal representation formalism), or *both*?

Note that an answer to this question does not necessarily imply a statement about acquisition support; observations might simply be stored.

**acquisition** Is acquisition truly *incremental* or *non-incremental* (i.e., are user model contents modified steadily or not)?

**representation** Is representation *explicit* (using an accessible format with a clear semantics, potentially reusable for more than one adaptivity feature) or *implicit*?

Note that explicit representation does not limit the input dimension to assumptions. E.g., a KR-based user modeling system can accept observations if these are coded in the available representation formalism.

**aspects** Information about what aspects of the user is maintained: information about user *behavior*, about *mental notions*, or about *other* user characteristics (e.g., demographic data)?

This dimension is related to representation, but not strictly: Behavior information might be represented explicitly. E.g., Mitchell et al. (1994) discuss the transformation of decision trees into an explicit, rule-like representation. Furthermore, mentalistic assumptions like interest profiles may be represented implicitly (s.a.).

**output** Does the user modeling system deliver *assumptions*, *decisions*, or *both*? I.e., can applications access a user model or a learning/decision process directly?

Note that this dimension is not strictly coupled to the input dimension; there may be user modeling processes that accept observations but are not accessible for decisions.

Table 1 shows the typical values of the above dimensions for KR-based and ML-based user modeling systems. However, many actual systems will deviate from these extremes. For example, many ML-based systems focus on user behavior, but ML-based information filtering systems implicitly represent assumptions about user interests (a mental notion).

Now we demonstrate the usefulness of these dimensions by using them to characterize a user modeling system that, at first sight, is a typical representative of neither KR-based nor ML-based user modeling. Like many other user modeling systems it uses a numerical approach to uncertainty management (see Jameson, 1996, for an overview). Such systems have also been

**Table 1.** Properties of typical KR-based and ML-based user modeling systems.

	<b>input</b>	<b>acquisition</b>	<b>representation</b>	<b>aspects</b>	<b>output</b>
<b>KR-based</b>	assumptions	non-incremental	explicit	mentalistic	assumptions
<b>ML-based</b>	observations	incremental	implicit	behavior	decisions

labeled “evidence-based” by Pohl (1998), since they explicitly maintain a degree of evidence for user model contents.

Our example is the user modeling component of the intelligent tutoring system HYDRIVE (Mislevy and Gitomer, 1996), which models a student’s competence at troubleshooting an aircraft hydraulic system. HYDRIVE employs a Bayesian Network (BN); the probability distribution of network nodes are used to *explicitly represent* variables like “Electronics Knowledge” or “Strategic Knowledge”, which are organized into different levels of abstraction. At the most concrete level, nodes represent “interpreted actions”. Such a node stores the probabilities that an observed sequence of user activities belongs to one of a fixed number of action categories. When HYDRIVE interprets an action sequence to belong to one category, it creates an interpreted action node and adds it to the BN with the probability of this category set to 1. That is, *observation* information constitutes the *input* to the user modeling system, but it has to be coded in terms of the BN formalism. *Acquisition is incremental*: While interpreted action nodes only cover a limited number of observations, they have long-term effects on the formation of assumptions through propagation of probability to higher level nodes like “Strategic Knowledge”. Since these mentalistic variables are the static part of the BN, mainly *mentalistic aspects* are represented in the user model of HYDRIVE. The probability distributions of these variables are the main source of adaptation; i.e., *assumptions* are the *output* of the user modeling system.

In several regards, HYDRIVE is typical of systems which use a BN for user modeling. In Jameson’s overview article (1996) we found that observation-related nodes are often used for input purposes, and that most systems are dealing with notions closer to mentalistic models than behavior profiles, ranging from “short-term cognitive states and events” to “personal characteristics”. In sum, by using our dimensions we found that user modeling systems with Bayesian Networks can have many typical KR-based properties. However, they share the property of incremental acquisition with typical ML-based systems; the reason is that also ML techniques implement some sort of uncertainty (using probabilities, distance measures, network weights, etc.).

### 3 The LaboUr Approach

According to the dimensions identified in the previous sections, what would an ideal user modeling system look like? First, it should be possible to report observations to the user modeling system. Thus, applications would not be forced to form assumptions about the user on their own. Second, acquisition should be incremental and not ignore interaction history, unless this is desired by an application. However, incremental acquisition processes should be complemented by heuristic acquisition, e.g., if quick results are needed and/or the number of observations is small.

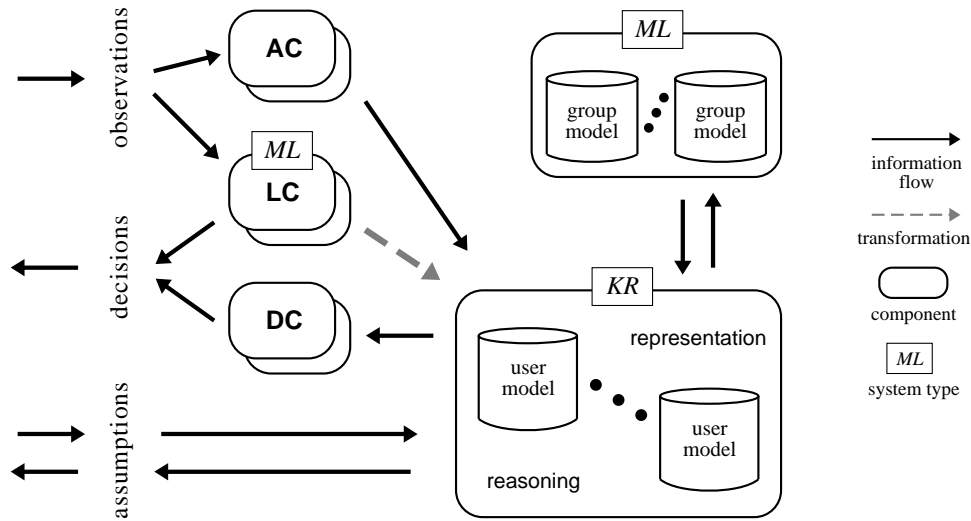


Figure 3. The LaboUr architecture.

Third, representation should be explicit, at least in cases where assumptions about the user may be useful to more than one adaptive feature and its decision process. That is, explicit representation is particularly important for a central user modeling service that is to be available to more than one application. Fourth, assumptions about the user should not be restricted to be either behavior-related or mentalistic, if both aspects are relevant to applications. Finally, the system should be able to support decisions, but ideally also allow for direct access to the user model.

We propose to integrate features of KR-based and ML-based user modeling in order to come closer to such an ideal system. A first step into this direction was made by the user model server *Doppelgänger* (Orwant, 1995), which uses learning methods to process information from several sources. Learning results are represented explicitly in a standardized format (all assumptions are stored using a symbolic notation with associated numerical confidence values), so that they can be used by all *Doppelgänger* clients. With acquisition and representation decoupled, several learning components can work on the acquisition of the same kind of data. For instance, *Doppelgänger* uses both hidden Markov models and linear prediction to acquire temporal patterns of user behavior, which are employed to predict future user activities.

Building upon these ideas, we proposed *LaboUr* (Learning about the User; Pohl, 1997), a user modeling architecture that integrates KR and ML mechanisms (see Figure 3). A *LaboUr* system accepts observations about the user, from which learning components (LCs) or acquisition components (ACs) may choose appropriate ones. LCs (which are ML-based) internally generate usage-related results that will be transformed into explicit assumptions, if possible. These assumptions are passed to a KR-based user modeling subsystem. ACs directly generate user model contents, which may be behavior-related or mentalistic, and do not support decisions. They can implement heuristic acquisition methods like those often used in systems with KR-based user modeling (cf. Section 2.1). In contrast to LCs, which typically need a significant



Figure 4. ELFI interface

number of observations to produce learning results with sufficient confidence, ACs can allow for “quick-and-dirty” acquisition from a small number of observations. This is useful to adaptive systems with short usage periods. LCs can be consulted for decision support based on learning results. In addition, there may be other decision components (DCs) that refer to user model contents. Besides supporting acquisition and decision processes, a LaboUr system can also offer direct access (input and output) to user models, due to its use of explicit representation facilities.

A LaboUr system may maintain several user models. In this case, ML techniques can further be used for group modeling, i.e., clustering user models into user group models. Then, individual user models may be complemented by suitable group information. LaboUr is an open user modeling architecture: Several sources of information about the user may contribute to the user model, which again can support several adaptive features or applications.

## 4 Applying LaboUr to ELFI

### 4.1 The ELFI Information Server

ELFI (ELectronic Funding Information) is a WWW-based brokering system for information about research funding. Users of ELFI are German researchers and research funding consultants working at universities and other research institutions. ELFI is described in detail by Nick et al. (1998). Here, we will only give a brief sketch of the system before focusing on adaptivity and implementation of LaboUr ideas in ELFI.

ELFI provides access to a database of funding programs and funding agencies. This information space is organized into hierarchies; for funding programs there are several hierarchies, e.g., of research topics and of funding types (like grant or fellowship). Based on these categories, users can actively specify their interest (e.g., in all fellowships in mathematics). The resulting interest profile is used for filtering unneeded information.

The interface of ELFI (see Figure 4) is divided into two parts. The left side presents one of the hierarchies (here: the research topic hierarchy) to be used as navigation tree. The checkmarks behind tree nodes can be used to specify an interest profile: a checked box shall indicate interest in a topic. The right side visualizes the contents of the current information subspace (specific funding programs or funding agencies). In Figure 4, funding programs on mathematics are listed, since this topic (“Mathematik”) was selected on the left side. The user can click each list item to see a detailed view of the corresponding program.



## 4.2 Adaptability and Adaptivity in ELFI

As users can modify their interest profiles, ELFI is an adaptable system. However, analysis of ELFI usage showed that profile settings are often used for temporary information filtering, so that they do not yield a long-term interest profile. Second, the profile is quite coarse, since it is relative to items of the hierarchies only. Therefore, for realizing adaptive features like recommendation of particularly relevant new research programs, we needed to add facilities for acquisition of a more stable and fine-grained and user model.

Adaptivity in ELFI is based on usage log analysis. ELFI records all user interactions with the system into a log file. Currently, we are exploiting the usage log in three different ways:

- Log files are scanned for key sequences that probably indicate user interest in a certain research topic. For example, if a user subsequently makes several selections from a list of documents related to one research topic (e.g., mathematics), then she is probably interested in that topic. This kind of heuristic sequence analysis constitutes a LaboUr acquisition component (AC).
- Navigation activity (i.e., selection of items in navigation trees) is analyzed by an LC in order to find out about frequently used items and frequent transitions between items. The results can be used to generate a personalized navigation tree: frequently used items from all navigation trees are merged, with items of one tree level sorted according to frequency of use, and less frequent items placed on a lower level. Furthermore, analysis results also indicate user interest in tree items (research topics, funding agencies, etc.).
- Finally, machine learning algorithms are employed within an LC to acquire interest information based on selection of detailed views. This is problematic, since these selections only provide positive learning examples. Learning results provide an assessment of user interests with respect to detailed view contents.

All these acquisition and learning components contribute to a comprehensive model of user interests with respect to research funding. Following the LaboUr approach, an explicitly represented user model will be constructed from the results of acquisition and learning components. This model can be used to support adaptivity decisions in ELFI, e.g. the selection of especially relevant new funding information for recommendation.

## 5 Conclusion

In this paper, we discussed and compared the use of different techniques in user modeling systems. We distinguished systems based on knowledge representation methods from systems that use machine learning techniques. Typical representatives of both approaches were examined, and a significant difference in handling user modeling tasks could be observed. Based on this discussion, we identified dimensions that can generally be used to characterize user modeling systems. Thus, we consider this paper a contribution to basic user modeling research.

Moreover, this paper presented the LaboUr approach to user modeling that tries to integrate advantages of KR-based and ML-based user modeling. We showed how LaboUr ideas are currently being implemented into the ELFI information server, thus giving a first demonstration of the applicability of the approach. In future work, we will establish more of LaboUr into ELFI, like group model formation. Furthermore, we plan to implement a LaboUr-based user modeling

service that could be utilized or even shared by the several other user-adaptive applications that are being developed in our group.

## References

- Brajnik, G., and Tasso, C. (1994). A shell for developing non-monotonic user modeling systems. *International Journal of Human-Computer Studies* 40:31–62.
- Jameson, A. (1996). Numerical uncertainty management in user and student modeling: An overview of systems and issues. *User Modeling and User-Adapted Interaction* 5(3-4):193–251.
- Kobsa, A., and Pohl, W. (1995). The user modeling shell system BGP-MS. *User Modeling and User-Adapted Interaction* 4(2):59–106.
- Kobsa, A., Müller, D., and Nill, A. (1994). KN-AHS: An adaptive hypertext client of the user modeling system BGP-MS. In *Proc. of the Fourth International Conference on User Modeling*, 99–105.
- Kobsa, A. (1989). A taxonomy of beliefs and goals for user models in dialog systems. In Kobsa, A., and Wahlster, W., eds., *User Models in Dialog Systems*. Berlin, Heidelberg: Springer. 52–68.
- Kozierok, R., and Maes, P. (1993). A learning interface agent for scheduling meetings. In Gray, W. D., Hefley, W. E., and Murray, D., eds., *Proc. of the International Workshop on Intelligent User Interfaces, Orlando FL*, 81–88. New York: ACM Press.
- Krogsæter, M., Oppermann, R., and Thomas, C. G. (1994). A user interface integrating adaptability and adaptivity. In Oppermann, R., ed., *Adaptive User Support*. Lawrence Erlbaum Associates.
- Lieberman, H. (1995). Letizia: An agent that assists web browsing. In *Proceedings of the International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers.
- Mislevy, R. J., and Gitomer, D. H. (1996). The role of probability-based inference in an intelligent tutoring system. *User Modeling and User-Adapted Interaction* 5(3-4):253–282.
- Mitchell, T., Caruana, R., Freitag, D., McDermott, J., and Zabowski, D. (1994). Experience with a learning personal assistant. *Communications of the ACM* 37(7):81–91.
- Moukas, A. G. (1996). Amalthea: Information discovery and filtering using a multi-agent evolving ecosystem. In *Proceedings of the Conference on Practical Application of Intelligent Agents and Multi-Agent Technology*.
- Nick, A., Koenemann, J., and Schalück, E. (1998). ELFI: information brokering for the domain of research funding. *Computer Networks and ISDN Systems* 30:1491–1500.
- Orwant, J. (1995). Heterogeneous learning in the Doppelgänger user modeling system. *User Modeling and User-Adapted Interaction* 4(2):107–130.
- Paiva, A., and Self, J. (1995). TAGUS – a user and learner modeling workbench. *User Modeling and User-Adapted Interaction* 4(3):197–226.
- Pazzani, M., and Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning* 27:313–331.
- Pohl, W. (1997). LaboUr – machine learning for user modeling. In Smith, M. J., Salvendy, G., and Koubek, R. J., eds., *Design of Computing Systems: Social and Ergonomic Considerations (Proceedings of the Seventh International Conference on Human-Computer Interaction)*, volume B, 27–30. Amsterdam: Elsevier Science.
- Pohl, W. (1998). *Logic-Based Representation and Reasoning for User Modeling Shell Systems*. Number 188 in Dissertationen zur künstlichen Intelligenz (DISKI). St. Augustin: infix.
- Rich, E. (1983). Users are individuals: Individualizing user models. *International Journal of Man-Machine Studies* 18:199–214.
- Wahlster, W., and Kobsa, A. (1989). User models in dialog systems. In Kobsa, A., and Wahlster, W., eds., *User Models in Dialog Systems*. Berlin, Heidelberg: Springer. 4–34.