Transparent User Modeling for a Mobile Personal Assistant

Mathias Bauer
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3
66123 Saarbrücken
bauer@dfki.de

Abstract

User models (UM)—the explicit representation of all relevant aspects of a user's preferences, beliefs, etc. —form the basis of virtually all adaptive systems. While early approaches relied on explicit interviews to fill in the details of such a UM, more recent systems unobtrusively observe their users and automatically infer the contents of the UM. This reduces the workload for the user, but it also makes her lose control over the system's knowledge about her and the way this knowledge is used. This paper discusses the necessity for *transparent* user modeling as well as various techniques for achieving it.

1 Introduction

A *user model* (UM)—the explicit representation of all relevant aspects of a user's preferences, goals, plans, intentions, beliefs etc.—forms the basis of most *adaptive* systems. Unlike *adaptable* systems these do not rely on the user customizing the system behavior to her particular needs, but rather try to infer what kind of customization (or "personalization") is most appropriate for the current user and her task. (Throughout this paper we will use the female form to refer to the user.) Possible adaptations include modifications of the user interface (e.g. hiding or highlighting of certain widgets), changing the style of information presentation, or re-ordering documents according to relevance for the user, to name but a few.

Early adaptive systems often filled their UMs by interviewing the user about her preferences ("explicit user modeling"). Nowadays, however, the user's regular interaction is typically observed in an unobtrusive way and the system automatically infers relevant parameters using e.g. machine-learning methods ("implicit user modeling"). (See [Fischer et al., 1991] for a discussion of the implications of these two paradigms and [Webb et al., 2001] for an overview of the use of machine learning for user modeling.)

While this lessens the user's burden to tell the system to do the right thing, she also loses control over the system's behavior, the data collected, and the inferences drawn from them. In order to increase the user's acceptance of the system behavior and her trust in the correctness of the UM, it is an indispensable prerequisite to allow the user to inspect, correct, and modify "her" UM. The bandwidth of solution approaches ranges from simple listings of UM entries to fully-fledged explanation components.

Making the user model "transparent" in this way poses a number of technical/design challenges and principle questions that are currently being addressed in a number of research projects some of which will be discussed below.

- How much *insight* does the user need and desire, and how can this insight be achieved? Can we design a language to communicate with the user that provides an appropriate abstraction from low-level observations without sacrificing precision and correctness of the information presented?
- How much *control* does the user need and want? Experiences from previous projects indicate that in certain situations many users prefer not to actively interfere with the system's user modeling process.
- How can the user be enabled to *exert* effective control? This includes not only providing means to alter particular UM entries, but also to anticipate the consequences of doing so.

The rest of this paper is organized as follows. Section 2 introduces the notion of *transparent user modeling* and discusses its necessity both from the user's perspective and from a legal point of view. Section 3 reviews a number of approaches to achieve transparency in adaptive systems before Section 4 provides a more abstract account of how to improve communication between user and system. Section 5 presents its application and its concrete implementation in the ongoing project SPECTER. Finally, Section 6 summarizes and concludes with some more general remarks.

2 The Need for Transparency

In order to regain (the option to) control over the way the system deals with personal data, the system has to provide the user with means to

- effectively *inspect* the current contents of her UM; this includes a clear identification of the information sources and inference processes involved (e.g. was a particular fact explicitly entered by the user or indirectly inferred using some reasoning mechanisms?);
- easily *correct* or *modify* the contents of the UM whenever she feels incorrectly assessed or deliberately chooses to alter its contents to influence the system behavior (optimally, this includes enabling the user to estimate in advance the consequences of actively interfering with the user model);

- actively *survey*, *direct*, or *bias* processes drawing inferences from the UM contents, thus producing additional entries explicitly represented in the UM or deriving "intermediary" facts for internal use by the system.

We call a system that implements all of these aspects a *transparent user-modeling system* and the class of UMs so generated *transparent user models*. (Judy Kay coined the notion of "scrutable" UMs to describe a very similar concept, see e.g. [Kay et al., 2001]). Currently we are not aware of any system that fully achieves all of these goals and so will be somewhat lenient in the use of the term "transparent", applying it also to systems approximating at least some of these desiderata. A discussion of some relevant work is the topic of Section 3.

There exist at least two different kinds of motivation for making the user-modeling process transparent to the user.

2.1 User-Dependent Motivations

There is sufficient evidence that the user's acceptance of the system behavior is strongly influenced by her understanding of (and control over) the processes going on—especially those involving her own personal data. This lesson already learnt in early expert systems like MYCIN [Shortliffe, 1976]— equipped with an *explanation component*—similarly applies to today's systems such as personalized product recommenders (see [Swearingen & Sinha, 2001] for an empirical evaluation of acceptance criteria for such systems).

Judy Kay provides arguments for the necessity of what she calls *scrutable* user models that allow the user to inspect and modify their contents whenever she feels inappropriately assessed by the system in applications such as adaptive hypertext [Czarkowski & Kay, 2003] or the general-purpose user model server Personis [Kay et al., 2001]. (Similarly, a study on preferences regarding Web browser settings revealed a tendency towards manually controlling the acceptance of cookies [Endler and Bayanifar, 1998]).

While this seems to speak clearly for the necessity to enable the user to exert maximum control over the system behavior—particularly in contexts where her personal data are exploited for the system's decision making—recent studies raise doubts about their general feasibility. Of about 6.400 repeated users of the adaptive Website for the conference User Modeling 2001 [UM, 2001], less than 7% cared to even inspect their user model and only about 3% actively modified it.

One particular component of this Website, an adaptive recommender system for conference presentations—the so-called *adaptive hotlist*—was thoroughly evaluated w.r.t. controllability [Jameson and Schwarzkopf, 2002]. The results indicate clear differences among the test subjects caused by either personal control needs and preferences as well as situational factors such as the speed of the Internet connection.

Similarly, a recent evaluation of a prototype for a commercial adaptive product recommender system for cars at DFKI resulted in less than 25% of all users modifying their respective user models, even though the consultation dialog almost inevitably ended on a page containing an easily editable visualization of the system's hypothesis of the user's preferences.

Obviously, not everybody always wants to have total control as this imposes additional responsibilities and workload on the part of the user and might be in conflict with her personality in general. This is in accordance with the observations reported about in [Heuwinkel, 2003] and [Adams & Sasse, 2001] where the aspect of *trust* in various social contexts is discussed. Trust—and thus, system acceptance by the user—is shown to depend on the particular type of situation and a shared understanding among all partners involved about how to deal with personal information.

2.2 Legal Aspects

Obviously, systems dealing with personal data, such as adaptive systems, touch a number of privacy issues. Independent from a user's personal attitude towards potential violations of her privacy, there exist numerous laws prescribing what *has* to done in any case. The basis for all these prescriptions is the user's fundamental right to know what information the system maintains about them and what it is used for. The 1995 European Privacy Directive [EU, 1995] postulates the user to be made available:

- her personal data for inspection and correction,
- information on who receives this data,
- the purposes for which the data were collected,
- the right to access to and correct the data, and
- "an intelligible form of the data undergoing processing and of any available information as to their source."

These postulates closely correspond to the requirements for a transparent user model as discussed at the beginning of this section (see also [Kay et al., 2001]).

To cut a long story short, no developer of a system dealing with personal information—among them all kinds of adaptive systems—has an excuse for ignoring transparency issues, no matter what special breed of users they expect or how warily they plan to take care of the personal data.

3 Previous Attempts to Achieve Transparency

The question of how to provide the user some insight into her UM has already played a central role in the well-known Doppelganger system [Orwant, 1994] in which a variety of UM visualization methods were applied ranging from regular email messages containing summaries of the current UM contents to graphical displays of Markov models representing the user's observed and predicted patterns of behavior.

The plan-recognition system presented in [Bauer, 1996] takes into account individual patterns of behavior to assess the likelihood of a plan hypothesis using a mixture of machine learning and a numerical uncertainty formalism to represent both qualitative and quantitative aspects of the user's observed behavior. In order to explain the reasons for selecting or discarding a plan hypothesis, a numerical sensitivity analysis identifies the most important recent observation and verbalizes the corresponding classification of the current episode with the aid of a decision tree learned from previous observations. So the system reveals just enough information about the UM to make the user understand the system's behavior. Full transparency, however, is not achieved as the user does not have access to the full UM and cannot easily correct its contents.

In contrast to this, the abovementioned website [UM, 2001] at any time provides a simple, editable visualization of the system's assessment of the user's interest in various document and event categories. The probability values computed using Bayesian networks are discretized into 5 levels from "low" to "high" which reduces the perceived complexity of the UM (and the computations performed to create it) while preserving the essentials of its contents for inspection and modification by the user.

In [Czarkowski and Kay, 2003], an adaptive hypermedia system explains adaptations performed for a particular user by simply visualizing those attribute/value pairs in her UM that triggered the adaptation. While this seems to work well as long as all entries of the UM are provided by the user (explicit modeling), the system is likely to encounter problems as soon as new UM entries result from complex inferences or immediate observations of the user behavior. In this case, the system also has to provide insight into how it arrived at these conclusions.

Among numerous other attempts to make the UM transparent to the user, the approach taken by the MIAU project is worth mentioning [Baldes et al., 2003]. In a group decision-making environment, the various group members are represented by virtual characters which—using "their" respective user's individual UM—negotiate with the other agents to arrive at a solution that is acceptable to everybody. Each character verbalizes relevant aspects of a UM and thus complements a simple interface with which the user can immediately edit various aspects of her UM, thus reacting on other users' suggestions.

4 The Communication Problem

One of the central problems in human-computer interaction consists in finding a common language that allows both user and system to express their respective view of the world in a way that is comprehensible to the other. While this is a very general observation, it notably applies to the attempt to make a UM transparent to the user. In [Bauer et al., 2001] a knowledge-level account of this *communication problem* is given which we repeat here in a slightly simplified version. The application discussed there deals with a programming by demonstration (PBD) system in which a user simply marks relevant portions of an HTML document in her browser and gives some additional hints. Using this input, the system is expected to learn a general procedure—a "wrapper"—to extract relevant information from Web documents with a particular structure and appearance.

Figure 1 depicts the three major categories of knowledge a user and a system can have in this situation. *Structural* knowledge deals with the internal structure of the domain objects to be dealt with (in this case, HTML documents). The *procedural* category comprises knowledge about how to create and run a wrapper. *Semantic* knowledge refers to the interpretation of the visual rendering of the HTML code in the browser—what is the meaning of presenting parts of a document using particular stylistic means or the spatial proximity of two objects etc.—and the text and graphics themselves (meaning of words, symbols, etc.)

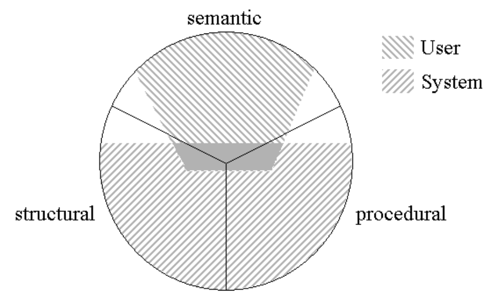

Figure 1: Shared knowledge of user and system.

User and system are obviously equipped with very different knowledge sets. While the system can easily deal with HTML code and "knows" how to generate and execute a wrapper—even if this knowledge is far from being complete, see Figure 1—, the user's strengths are clearly on the semantic side which is largely inaccessible to the system (just as are the structural and procedural subtleties to the majority of users).

The small grey-shaded region in Figure 1 depicts the overlap of concepts that are known to both user and system and thus form the basis for a dialog. Improving communication between both partners requires enlarging this region. Of the two possible ways—teaching the user technical details or making the system understand some of the semantics involved—only the latter was actually implemented. (After all, PBD is about teaching the system, not the user.)

In order to do so, the system was equipped with some spatial reasoning capabilities and heuristics to relate spatial phenomena to semantic interpretations. As a consequence, the basis for communication between user and system was considerably enlarged as a whole class of spatial concepts—which are used in a very intuitive way by most users—could now be used to describe target concepts and explain processing steps. This way of improving communication between user and system also formed the basis for addressing similar problems within the project SPECTER to be described next.

5 Transparency in SPECTER

The project SPECTER aims at developing a context- and affect-aware mobile personal assistant in particular for instrumented environments (see Kröner et al., 2004). In order to optimally support its user, the system permanently observes her actions, her physiological reactions to certain events, and various aspects of the environment (e.g. the presence of certain persons or objects, the current location etc.) using a variety of sensors. Using this rich input, the system applies a number of inference mechanisms such as plan recognition as well as machine-learning techniques in order to create both a *personal journal* (PJ) and a user model (UM). While the latter contains the system's assumptions about the user's long-term preferences, goals etc., the former represents a kind of episodic memory which serves as an information repository to both the user and the system.

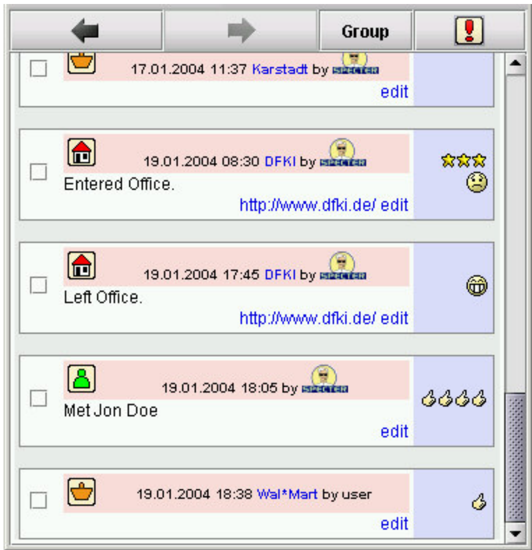

Figure 2: Excerpt of a Personal Journal.

One particular aspect of user support consists in facilitating decision-making or triggering certain services in particular kinds of situations. (E.g. check the bank account whenever the user is likely to go shopping and spend more than she is carrying in cash.)

Remark: From the point of view of context-aware systems research (e.g. [Chen and Kotz, 2000]), this means that SPECTER manifests both active and passive context awareness—*active* in that it exploits information about the current situation to modify the system behaviour; *passive* in that it creates a Personal Journal that serves as a kind of episodic memory to the user. The particular privacy requirements for such a system are discussed e.g. in [Jiand and Landay, 2002].

SPECTER obviously deals with a huge amount of very personal information and uses it to automatically—albeit under the user's control—perform actions in the real world. As a consequence it is of vital interest to the user to be informed about the data maintained by SPECTER and their actual usage in a given situation. This situation is aggravated by the fact that both the various sensors may deliver imperfect readings and the inference mechanisms applied are far from being perfect, thus inducing some uncertainty in their results.

5.1 Interaction with the Personal Journal

In order to present its entries to the user, the personal journal (PJ) is equipped with an interface based on a browser/viewer metaphor. The browser provides the basic machinery for administrating all entries, navigating through the PJ, and fulfilling (retrieval) requests by other components or the user. The user indirectly interacts with the PJ using a variety of *viewers*, special-purpose visualization components that provide different presentation and interaction facilities depending on the current task, context, or the types of entries.

Using these viewers, the user can

- set explicit reminder points, indicating that an entry needs closer examination at some later point in time;
- group a set of entries, thus indicating an inherent relationship (to be further elaborated) among them;

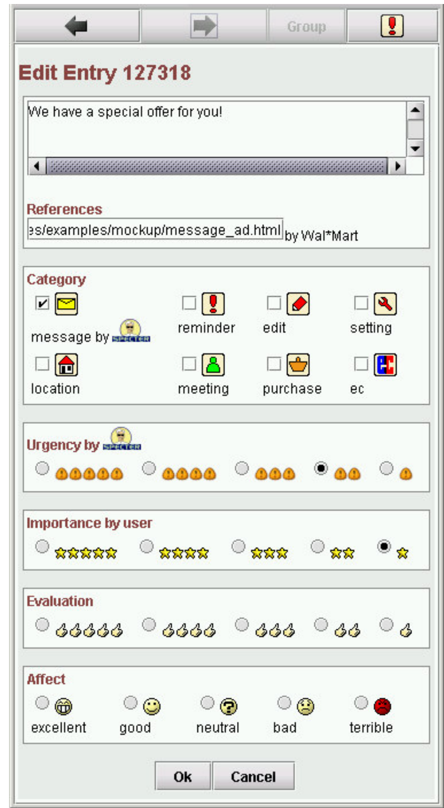

Figure 3: The PJ Annotation Editor.

- annotate an entry with free text, links to other entries or Web pages, category assignments, or ratings.

Figure 2 depicts an excerpt of a PJ with an entry representing a seemingly agreeable encounter with some person (Jon Doe). This rating was inferred by SPECTER (indicated by the icon with the project logo) on the basis of physiological sensor readings and can be manually overridden by the user. (Note that declaring the source of an entry is an important prerequisite for transparency of the system behavior.)

The last entry depicted refers to a shopping episode in a department store that—according to the user's own assessment—was rated rather negatively (only one thumb up). Figure 3 shows part of the annotation editor that allows the user to annotate and rate an entry w.r.t. a number of categories. In the case depicted, the user rates an entry containing a message from the department store as completely unimportant (one star). Note that the event category (message) and the urgency level were automatically filled in by SPECTER.

Remark: In addition to controling and overriding system-made entries, this functionality can also be used to provide *additional* input to the system in cases when e.g. particular types of sensors are not available. For example, even without wearing physiological sensors, a user can still add information about her affective state in a certain situation that can serve as valuable input to the system, e.g. for the identification of classes of unpleasant events.

5.2 Collaborative Learning of the User Model

While the interaction with the PJ affected the basic entries resulting either immediately from sensor readings or from a more complex inference process, the user from time to time also needs to control and influence these processes themselves.

Assume the user wants to teach SPECTER to react in a particular way in certain situations. An example would be the automatic execution of a bank account status check whenever the user is likely to use her EC card to pay for her shopping. (For non-European readers: An EC card is like a credit card except that the funds are transferred to the recipient directly from the purchaser's bank account; so trying to pay with insufficient funds will result in a slightly embarrassing situation.) Using this particular means of payment may depend on the total sum of all purchases, the type of store (not all stores accept EC payment), her current mood or a number of other factors. So SPECTER will need a way to classify (shopping) events in order to predict the use of an EC card and trigger the bank account check.

To create such a model of this class of events, the following steps have to be carried out:

- identify examples for this type of situations;
- select appropriate features to characterize these data;
- apply a machine-learning algorithm to create a classification model and check its accuracy.

In SPECTER this is a collaborative process between user and system in which the communication problem as discussed in Section 4 occurs.

In a first step, SPECTER's machine-learning component needs a number of *training examples*—previous shopping episodes stored in the personal journal that can be used to distinguish EC payments from "non-EC payments".

Assume the user marked a PJ entry representing an unpleasant shopping experience as mentioned above by setting a reminder point. Later, when the user decides to work on the problem of creating a classifier for this type of situation, the system displays the entry marked with the reminder point and asks the user to indicate what is special about it. The user indicates the use of the EC card ("MeansOfPayment = EC card" in the PJ) whereupon SPECTER looks for previous entries of the same category (shopping) with identical and differing values for MeansOfPayment and classifies these examples according to the value specified ("positive" for EC card, "negative" for all other values). This subset of historical PJ entries will serve as training data for the subsequent process.

Once the training data have been identified, SPECTER applies a machine-learning algorithm to create an appropriate classifier. One useful learning technique in this context is decision tree learning, which yields a relatively comprehensible type of model. Even though users would rarely be willing or able to define a decision tree of acceptable accuracy entirely by hand, critiquing a decision tree proposed by the system may be a reasonably easy—and perhaps even enlightening—activity, if the user interface is well designed.

When the system presents a learned tree, the user can critique it in several ways, including:

- eliminating irrelevant attributes,
- selecting paths from the tree,
- and modifying split decisions.

The question of what interface designs are best suited for this type of critiquing requires further exploration and user testing. The simple interface depicted in Figure 4 allows the user to critique the current decision tree without urging her to deal with its internal structure or the way it is constructed from the data. (Note that there is an advanced interface that allows the user to inspect the decision tree either in a graphical way or decomposed into an equivalent set of decision rules.)

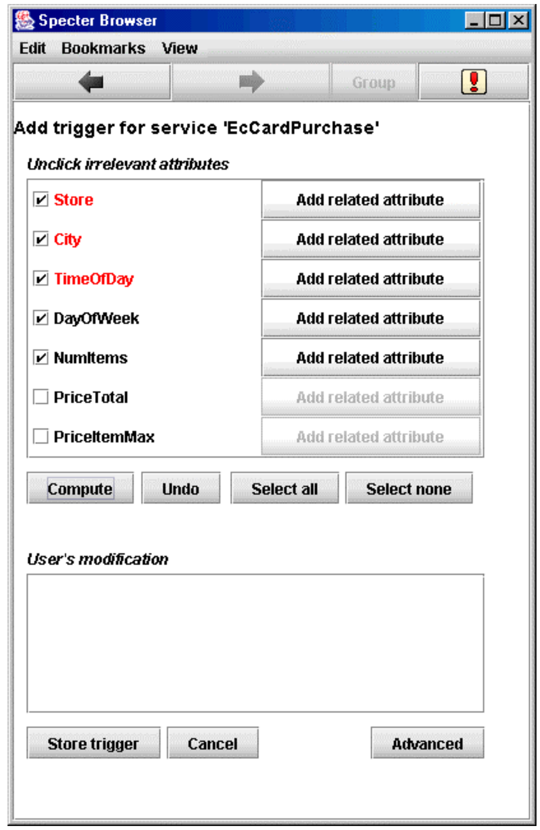

Figure 4: The interface for critiquing a decision tree (simple version).

Here all the user needs to know is the set of attributes used to describe the training data and discriminate between positive and negative cases. If the user, for example, feels that the "DayOfWeek" should not play a role in this process—mainly because she herself does not base her decision on whether or not to use her EC card on this criterion—she can either simply remove this attribute, thus disallowing its further use, or replace it by another, semantically related attribute capturing other, possibly more relevant time-related aspects of her shopping behavior.

Searching for candidates to replace a seemingly inappropriate attribute is done in a collaborative way. SPECTER is equipped with a domain ontology capturing important concepts and their (semantic) interrelationships e.g. within the shopping domain. In an interactive, *ontology-guided exploration* process, the system suggests a set of candidate attributes derived from semantically related ontology concepts, thus enabling the user to explore the semantic neighborhood of those attributes used in the original tree.

As in the PBD example described in Section 4, extending the conceptual basis shared by user and system serves the purpose of improving the communication of both partners.

As mentioned there, this is achieved by providing additional semantic knowledge to the system that can be naturally interpreted by the user. In the SPECTER scenario, *semantic* knowledge refers to background knowledge in a variety of domain, including a good deal of commonsense knowledge. *Structural* and *procedural* knowledge refer to the ability to identify (statistical) correlations among the sample data taken from the PJ and translating these into patterns that can be used for classification purposes, respectively.

Unlike in the PBD example, we also try to improve the user's such as to make her understand at least the basic principles underlying e.g. a concrete decision tree as described above. To this end, we are currently investigating various visualization techniques and hope to be able to present first results at the workshop.

6 Conclusion

In this paper we tried to argue for and explain the necessity of making the behavior of complex systems transparent to their users and sketched the way this is done in the project SPECTER. This kind of transparency is particularly important for adaptive systems that make use of very sensitive, personal information. We tried to make clear that there are a number of good reasons to allow the user some insight into the system behavior and the way it deals with the user's data, ranging from very personal fears to legal regulations. A good communication between user and system forms the basis for any kind of transparency. The knowledge-level account of potential problems hindering such an effective communication (as discussed in Section 4) provides general guidelines for overcoming difficulties arising from an insufficient set of shared concepts.

Acknowledgements

This research was supported by the German Ministry of Education and Research (BMB+F) under grant 524-40001-01 IW C03 (project SPECTER). Thanks to all project members.

References

[Adams and Sasse, 2001] Adams, A. & Sasse, M.A. (2001). *Privacy in Multimedia Communications: Protecting Users, Not Just Data*. In A. Blandford, J. Vanderdonckt & P. Gray (Eds.), *People and Computers XV - Interaction without frontiers. Joint Proceedings of HCI2001 and ICM2001* (pp. 49-64).

[Baldes *et al.*, 2003] Baldes, S., Bauer, M, Dengler, D., Kudenko, D., Paul, G., Rist, T. & Schmitt, C. (2003). *MIAU—Supporting Group Decisions in E-Commerce Applications*. In J.A. Jacko (Ed.), *Proceedings of the 10th International Conference on Human-Computer Interaction (HCII03)* (pp. 193-202).

[Bauer, 1996] Bauer, M. (1996). *Justification of Plan Recognition Results*. In W. Wahlster (Ed.), *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI96)* (pp. 647-651).

[Bauer *et al.*, 2001] Bauer, M., Dengler, D. & Paul, G. (2001). *Trainable Information Agents for the Web*. In H. Lieberman (Ed.), *Your Wish Is My Command* (pp. 87-114). Morgan Kaufmann.

[Chen and Kotz, 2000] Chen, G. and Kotz, D. (2000) *A Survey of Context-Aware Mobile Computing Research*. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.

[Czarkowski and Kay, 2003] Czarkowski, M. & Kay, J. (2003). How to give the user a sense of control over the personalization of AH? In *Proceedings of the Fourth Workshop on Adaptive Hypertext and Hypermedia* (pp. 121-132).

[Endler-Bayanifar, 1998] Endler-Bayanifar, H.-G. (1998) Online poll regarding cookie preferences at www.umfrage.com (does not exist anymore).

[European Union, 1995] European Union. (1995). The European Parliament and the Council of the European Union, European Community Directive on Data Protection at www.doc.gov/ecommerce/ eudir.htm.

[Fischer *et al.*, 1991] Fischer, G., Lemke, A., Mastaglio, T. & Morch, A. (1991). *The Role of Critiquing in Cooperative Problem Solving*. ACM Transactions on Information Systems, 9, 123 – 151.

[Heuwinkel, 2003] Heuwinkel, K. (2003). *Subjective Aspects of Personalization: The Impact of Trust and Information*. In P. Isaias and N. Karmakar (Eds.), *Proceedings of IADIS WWW/Internet 2003 Conference* (pp. 211-218).

[Jameson and Schwarzkopf, 2002] Jameson, A. & Schwarzkopf, E. (2002). *Pros and Cons of Controllability: An Empirical Study*. In P. de Bra (Ed.), *Adaptive Hypermedia and Adaptive Web-Based Systems: Proceedings of AH 2002* (pp. 193-202) Berlin: Springer.

[Jian and Landay, 2002] Jiang, X. and Landay, J. (2002) *Modeling PrivacyControl in Context-Aware Systems*. In *IEEE Pervasive Computing* 1 (3), 59-63.

[Kay *et al.*, 2001] Kay, J., Kummerfeld, B. & Lauder, P. (2001). *Foundations for personalised documents: a scrutable user model server*. In *Proceedings of ADCS'2001, Australian Document Computing Symposium* (pp. 43-50).

[Kröner *et al.*, 2004] Kröner, A., Baldes, S., Jameson, A. & Bauer, M. (2004). *Using an Extended Episodic Memory Within a Mobile Companion*. In K. Mase, Y. Sumi & S. Fels (Eds.), *Proceedings of the "Pervasive2004" Workshop on Memory and Sharing of Experiences*. to appear.

[Orwant, 1994] Orwant, J. (1994). *Apprising the User of User Models: Doppelganger's Interface*. In B. Goodman, A. Kobsa, & D. Litman (Eds.), *Proceedings of the 4th International Conference on User Modeling (UM94)* (pp. 151-156).

[Shortliffe, 1976] Shortliffe, E.H. (1976). *Computer-Based Medical Consultation: MYCIN*. Elsevier.

[Swearingen and Sinha, 2001] Swearingen, K. & Sinha, R. (2001). *Beyond algorithms: An HCI perspective on recommender system*. In ACM SIGIR Workshop on Recommender Systems, New Orleans.

[UM, 2001] UM. (2001). *Official Website of the 8th International Conference on User Modeling (UM2001)* at www.dfki.de/um2001.

[Webb *et al.*, 2001] Webb, G., Pazzani, M. & Billsus, D.. (2001). *Machine Learning for User Modeling*. User Modeling and User-Adapted Interaction, 11, 19-29.