User Modeling: Recent Work, Prospects and Hazards[1]

Alfred Kobsa[2]
WG Knowledge-Based Information Systems
Department of Information Science
University of Konstanz
D-78434 Konstanz
GERMANY

Abstract

User modeling has made considerable progress during its existence now of more than a decade. In this paper, a survey of recent developments will be presented, which concentrates on the modeling of a user's knowledge, plans, and preferences in a domain, on the exploitation of new sources of information about the user, on issues of representation, inference and revision, on user modeling shell systems and servers, and on the verification of the practical utility of user models. Research trends and research deficiencies in these areas will be outlined, and potential risks described.

1. Introduction

User modeling has made considerable progress during its existence now of more than a decade. Particularly in the last few years, the need for software systems to automatically adapt to their current users has been recognized in many application areas. Consequently, research on user modeling (which originated in the field of natural-language dialog systems) has spread into many disciplines which are concerned with the development of computer systems that are to be used by heterogeneous user populations. These fields include Human-Computer Interaction, Intelligent Interfaces, Adaptive Interfaces, Cognitive Engineering, Intelligent Information Retrieval, Intelligent Tutoring, Active and Passive Help Systems, Guidance Systems, Hypertext Systems, and Expert Systems, to name just the most prominent application areas.

Several introductory surveys on user modeling have already appeared in the last few years (see, e.g., (Kass & Finin, 1988; Kobsa & Wahlster, 1988; Kobsa & Wahlster,

---

1) Appeared in M. Schneider-Hufschmidt, T. Kühme and U. Malinowski, eds. (1993): Adaptive User Interfaces: Principles and Practice. North-Holland, Amsterdam, 1993. Copyright © North-Holland. The described research has been supported by the German Society for the Advancement of Scientific Research through Grant No. Ko 1044/4-2, and by the University of Konstanz through Grant No. AFF 17/92.

2) The paper has been written while the author was a visiting researcher at the Department of Computer Science, Columbia University, New York, NY.

1989; Norcio & Stanley, 1989; Kok, 1991)). This paper therefore concentrates on more recent research trends and describes interesting developments as well as research deficiencies. Examples include the currently very popular stereotype approach, a technique that has been frequently employed for the modeling of users' domain knowledge. Another very active area at the moment is plan recognition, albeit mostly from a research perspective and not so much from an application perspective. The interest in the modeling of user preferences has also sharply risen, particularly as far as the users' information preferences are concerned. New techniques and information sources for acquiring assumptions about the user are being explored. These techniques link user modeling research more closely to modern user interfaces than did previously employed techniques that relied strongly on a (fictional) natural-language interaction. The increased demand for expressive representation systems for user models, powerful general inference mechanisms, and the need for assumption revision in user models led to a transfer of AI techniques into user modeling research. Integrated representation, reasoning and revision tools (so-called 'user modeling shell systems') are being developed that should facilitate the deployment of user modeling components in application systems. Finally, the empirical validation of the utility of developed adaptive systems has strongly gained in importance. The first results suggest that users will indeed profit from systems that adapt to their needs.

2. User knowledge

The stereotype approach to user modeling (which was proposed by (Rich 1979ab; Rich, 1989) and later refined, e.g., by (Chin, 1989)) has proven very useful for application areas in which a quick but not necessarily completely accurate assessment of the user's background knowledge is required. In this approach, the developer of a user modeling component in an application system has to fulfill three tasks:

- *User subgroup identification*: The user model developer must identify subgroups within the expected user population whose members are very likely to possess certain homogeneous application-relevant characteristics. The user population of a statistical database with national hospitalization data for cancer patients could, for instance, be divided into medical professionals, scientists and statisticians, and on a more specific level, doctors, nurses, medical statisticians, actuaries, hospital managers and policy makers.

- *Identification of key characteristics*: The user model developer should identify a small number of key characteristics which allow one to identify the members of a user subgroup (the presence or absence of these characteristics should be recognizable by a computer system). In the example of the hospitalization database, such characteristics could be the kind of questions users ask, the terminology they employ, the kind of statistical information they seek, the level of help they request, etc.

- *Representation in (hierarchically ordered) stereotypes*: The application-relevant characteristics of the identified user groups must be formalized in an appropriate representation system. The collection of all represented characteristics of a user subgroup is called a stereotype for this subgroup. If the contents of one stereotype form a subset of the contents of another stereotype, stereotype hierarchies may be constructed in which the contents of superordinate stereotypes become inherited by the subordinate stereotypes and hence need only be represented once. In Fig. 1, stereotypes were developed for the user subgroups in the hospitalization domain (they contain, for instance, assumptions about the technical terms that are familiar to these subgroups). The stereotypes are ordered in a hierarchy. The universal characteristics of all system users are contained in the topmost stereotype only, and become inherited by its subordinate stereotypes. The stereotypes named 'medical professionals', 'scientists' and 'statisticians' contain a representation of the characteristics of the respective subgroups. Their contents become again inherited by their subordinate stereotypes.

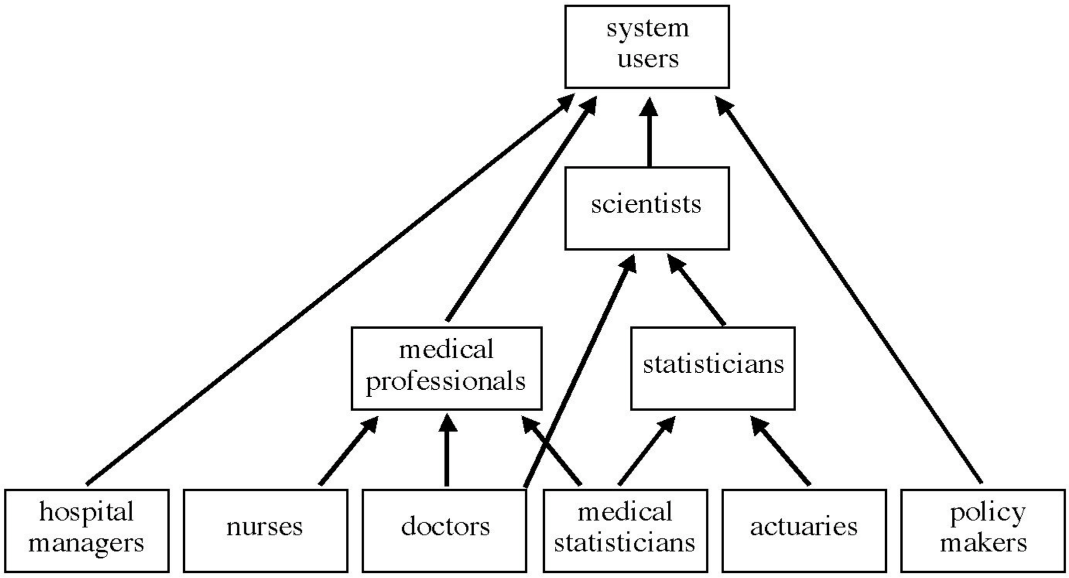

Fig. 1: Example stereotype hierarchy in a medical domain

Stereotypes for user groups with different background knowledge have been specifically employed for user-adapted information and advice on operating systems (Chin, 1989; Nessen, 1989; Boyle & Encarnacion, 1994). Most authors postulate a linear hierarchy of stereotypes corresponding to beginners, intermediates and experts. However, it is doubtful whether such a simple classification of users can be empirically justified. (Sutcliffe & Old, 1987), for instance, found that the modeling of users' familiarity with operating system commands requires the identification of coherent knowledge clusters in the user, rather than levels of user expertise.

Since stereotypical assumptions are particularly prone to inaccuracy, considerably more attention is paid to allowing users to inspect and possibly also modify their system's model of them (see e.g. (Kay, 1990; Orwant, 1991; De Rosis et al., 1992; Boyle & Encarnacion, 1994)). (Kobsa, 1990b) questions, however, whether this will also be possible for more general user models which will contain a far larger number of mostly trivial and low-level assumptions. Moreover, these assumptions are hard to translate from the formal representation language employed (e.g., modal logic) into a language that is comprehensible to the user.

3. User plans

Much emphasis has been recently put on the recognition of users' plans, where a plan is a sequence of user actions that achieve a certain goal. Plan recognition systems observe the user's input actions and try to determine all possible user plans to which the observed actions can be complemented. This set of candidate plans for the unknown user plan can be further narrowed down as soon as new user actions occur.

While the basic algorithm is fairly simple and straightforward, serious combinatorial problems arise when it is practically employed, due to the following reasons:

a) it is often unclear when the user commences a new plan;

b) actions and short action sequences may often be part of more than one plan;
c) users may interrupt or suspend the execution of their current plans (for various reasons, such as when issuing the ‘date’ command or when replying to an email message which they just received);
d) there is often more than one action sequence for achieving a (sub-)goal (i.e., there can be variations of user plans).

Two kinds of techniques are mainly employed for the recognition of users’ plans:

- *Plan libraries*: In this approach, all possible user plans are already pre-stored in a so-called plan library (possibly, these plans contain open variables which have still to be instantiated). The observed user action sequence is compared with these pre-stored plans, and all plans are selected whose beginnings match the observed user input. In this approach, it is difficult to take the possibility of plan variations into account. All permissible deviations from a plan would have to be stored as separate plans.

- *Plan construction*: In this approach, the system possesses a library of all possible user actions, together with the effects and the preconditions of these actions. The observed user action sequence is completed by all possible user action sequences which fulfill the requirement that the effects of preceding actions meet the preconditions of subsequent actions.

The generality of the plan library approach is restricted by the requirement that all possible user plans must already be specified beforehand. However, this need not be a problem in small domains in which users can only pursue a limited number of goals. The plan construction approach is problematic from a complexity point of view. Often the number of possible extensions to observed user input that can be constructed from action operators is very large, particularly since the length of the user’s plan is unknown. Recent work therefore investigates focusing heuristics that favor the further analysis and expansion of certain plan candidates over others, based on domain knowledge. (Appelt & Pollack, 1992) present a framework for choosing among competing plan ascriptions based on weighted abduction. (Raskutti & Zukerman, 1991) address the same problem by measuring the coherence and information content of the ascribed plan using Bayesian probability theory. (Mayfield, 1992) investigates when to terminate the plan-goal explanation chain based on concepts of explanation completeness and the system’s expectations and curiosity.

Other authors deal with the problem that the user’s actions may not necessarily be correct. One should therefore be hesitant to exclude plan candidates that do not match the user’s input if the mismatch is only small. In this vein, (Eller & Carberry, 1992) provide a unified framework for recognizing plans from slightly ill-formed and well-formed dialogs using meta-rules to relax constraints when necessary.

4. User preferences

Until recently, little attention has been paid to the modeling of user preferences. One of the few exceptions is the HAM-ANS system which modeled users’ preferences with respect to hotel rooms they could book through the system (this work is summarized in (Morik, 1985)). (Morik & Rollinger, 1985) modeled a real estate agent that made assumptions about the preferences of clients who looked for apartments to rent.

Recently, however, considerably more work is being spent on the modeling of user preferences, particularly their information preferences. In the area of intelligent information retrieval, (Brajnik et al., 1987) envisages a comprehensive natural-language access system to bibliographic databases that models the user’s information needs. (Botman et. al, 1989) developed an access system to relational databases that draws assumptions about the user's interests based on the requested information, and volunteers additional information in which the user might be interested as well.

In the area of information filtering, (Jennings and Higuchi, 1993) developed a personalized electronic news server that is based on a connectionist network architecture. In the learning phase, the network is supplied with a sample of electronic news articles that the user finds interesting. Nodes and link weights adapt to the frequency of information-bearing words in these articles and the probability of their co-occurrence, respectively. After the learning phase, the network is used for determining which news articles should be presented to the user. The adapted network can be regarded as containing the user's information preferences with respect to news articles, represented by the concepts that are likely to occur in such articles. In a similar vein, (Kass & Stadnyk, 1992) propose to monitor users' access to technical component databases in order to determine who is interested in certain parts, and who should therefore receive company-wide distributed email messages with engineering change notices.

In the area of adaptive hypertext systems, (Kaplan et al., 1993) developed the HYPERFLEX system that recommends information to users based on their specific informational needs and preferences. The system is based on associative matrices that record the hypertext frames which users with certain goals usually inspect, and bases its advice to new users on this past history.

5. User model acquisition

Important changes can be observed concerning the sources of information that user modeling components exploit for drawing assumptions about the user. While early research focused strongly on natural-language input, a much wider variety of knowledge sources have been employed recently. These include, among others,

- the observation of users' direct-manipulative interaction with software systems such as
  - -- text editing in a WYSIWYG editor (Hirschmann, 1990; Krause et al., 1993),
  - -- navigation in a hypertext system (Kaplan et al., 1993),
  - -- stretchtext activation in a hypertext system (Boyle & Encarnacion, 1994),
  - -- usage of a command-based interface (Benyon, 1993),
- the analysis of the information which a user retrieves from a database (Botman et al., 1989; Kass & Stadnyk, 1992),
- the analysis of electronic news in which the user is interested (Jennings & Higuchi, 1993),
- a number of other information sources, particularly UNIX resource files (Orwant, 1991).

Although natural language is seemingly a very rich information source for the acquisition of assumptions about a dialog partner (see (Kobsa, 1983) for many examples), this recent liberation from the burden of natural-language analysis seems very important for the practical applicability of user modeling techniques. It can be anticipated that the first commercial systems will most likely not rely on natural-language input.

Other novel research trends in the acquisition of user models can be observed on the interaction level. Acquisition so far was mostly either explicit (i.e., the system asks the user for information about him/her) or implicit (i.e., the system performs user modeling based exclusively on its normal interaction with him/her). Work by (Wu, 1991) and (Shifroni & Shanon, 1992) is concerned with mixed-mode acquisition of assumptions about the user, and particularly with decision strategies on when the system should initiate an explicit acquisition phase.

6. Representation, Inference and Revision

Issues of formal representation and reasoning did thus far not play an important role in the field of user modeling. With the exception of KL-ONE-like languages (which were used for instance by (Kobsa, 1985; Paris, 1989; Kobsa, 1990a; Kass, 1991)), formal representations have hardly been employed. This situation changed recently due to increased representational and inferential demands that are imposed on user models. Currently investigated representations include the following:

- *PROLOG* (Finin, 1989; Eydner & Vergara 1993), which offers a comparatively rich representation language with "built-in" backward reasoning and the possibility of a smooth migration from knowledge representation to programming;

- *Predicate logic* (Appelt & Pollack, 1992; Kobsa, 1992; Fink & Herrmann, 1993), which offers more expressiveness than PROLOG, as is needed in many application domains;

- *Languages with second-order predicates* (van Arragon, 1991) and *modal logi*c (Kobsa, 1992), which allow one to represent assumptions about (nested) beliefs and goals of different agents in the same representation language;

- *Connectionist networks*, which have been particularly employed for classification tasks (Bodendorf et al., 1990; Jennings & Higuchi, 1993).

Entries in user models are mainly assumptions and not confirmed facts, and possibly must later be revised when new information about the user becomes available. A problem arises if such assumptions were already used for inferring additional assumptions about the user. In this case, not only the original assumptions but also all inferences based on them must be retracted, unless they are supported by other assumptions. Ongoing research concerns the development of reason maintenance systems (Huang et al., 1991; Brajnik & Tasso, 1992; Eydner & Vergara, 1993) that record the dependencies among assumptions. When conflicts are detected among the assumptions about the user, these systems determine which assumptions should be retracted to resolve the conflict.

7. User modeling shell systems and servers

User modeling components in software systems are often expensive to develop. Thus far, system developers essentially have to start from scratch each time. Research on user modeling shell systems aims therefore at the development of integrated representation, reasoning, and revision tools that form an "empty" user modeling mechanism. When filled with application-dependent user modeling knowledge, these shell systems would fulfill essential functions of a user modeling component in an application system. Parallels to this line of research can be found, e.g., in the field of expert systems in which the experience gained from individual expert systems lead to the development of expert system shells. In return, the availability of these shells considerably stimulated the development of application systems.

In the next few subsections, the 4 major user modeling shell systems that have been developed to date will be surveyed. Orwant's (Orwant, 1991) "personalization server" will then be briefly described which is the first instance of a user model server, i.e. a user modeling system that is not part of an application system but rather an independent process that communicates with all applications that the user is interacting with, and supplies them with assumptions about the user.

7.1 GUMS

The GUMS system (Finin, 1989), which is based on Prolog, is aimed at providing a set of services for the maintenance of assumptions about the user's beliefs. GUMS does not draw assumptions itself. Instead, it accepts and stores new facts about the user which are provided by the application system, verifies the consistency of a new fact with the currently held assumptions by trying to deduce the negated fact from the current assumptions, informs the application system about recognized inconsistencies, and answers queries of the application concerning its current assumptions about the user.

The shell system allows for the definition of a stereotype hierarchy (however, the hierarchy is constrained to a tree, and only a single stereotype may apply to the user at a time). The initial stereotype pertaining to the user must be selected either by the user model designer or the application program. Each stereotype includes a number of definite facts about the user; if one of these facts is contradicted with new information about the user, the stereotype is abandoned. It is then replaced by the most specific direct or indirect superordinate stereotype that does not contain the contradictory fact (stereotypes in parallel paths are not considered).

Two types of inference rules are supported by GUMS, namely definite rules and default rules. These are processed in a backward chaining mode when GUMS answers queries of the application system concerning the deducibility of a given fact from the current assumptions about the user, and when GUMS determines whether a new fact about the user is consistent with the current assumptions by trying to deduce the negation of this fact from its current assumptions. In both cases, negation by failure is possible if a predicate that has been asked for has been declared as closed, i.e., as defined by the current system knowledge pertaining to it. The use of default rules requires GUMS to continue to search for a goal until one is found that is not based on defaults, or until all solutions have been checked. The result of a query to the knowledge base is never recorded. Thus, no truth maintenance is needed since no assumptions exist in the knowledge base that were deduced from other assumptions.

7.2 UM

UM (Kay, 1990) is intended to become a toolkit for user modeling. It is still in an experimental stage at the moment and its architecture is considerably simpler than those of all other user modeling shell systems. The system offers two principal mechanisms for the development of user modeling components, namely

- *structured text files* organized in a directory hierarchy, each of which collects details of various aspects of the user's knowledge in one subdomain, and
- a *rule interpreter* which executes rule-like programs that amend the user model.

Each entry in a text file of the user model contains the following information:

- a knowledge "component" (in Kay's terms) with which the user is assumed to be familiar or not familiar,
- a list of information which supports this assumption (six different information sources with different degrees of reliability are distinguished), and
- a list of information which negates this assumption.

Individual user models that can be constructed using UM are thus thematically organized (the hierarchy reflects no inheritance relationship). There exists no representation system within the user model; instead, the interpretation of the entries is left over to the programs that retrieve assumptions about the user. UM also does not contain any stereotype hierarchy, although a stereotype may be one of the above-mentioned knowledge sources which support or negate an entry in the user model. Any kind of retrieval or updating of information in the user model has to be performed by application programs that may be written in the format executable by UM's rule interpreter.

An interesting aspect of UM's user models is that these are regarded as belonging to the respective users who can access them and, if they wish, modify them. The fact that users may possibly "sabotage" their own models is tolerated. This approach has interesting consequences as to the control of users over the models of them which the system constructs (see section 9).

7.3 UMT

UMT (Brajnik & Tasso, 1992) allows the user model developer to define user stereotypes that contain the characteristics of user subgroups in the form of attribute-value pairs. Stereotypes can be ordered in arbitrary hierarchies, which support the inheritance of stereotype contents. Each stereotype possesses an activation condition which specifies when the stereotype can be applied to the current user. UMT also puts a rule interpreter at the disposition of the user model developer which allows for the definition of user modeling inference rules. Possible contradictions between assumed user characteristics also have to be explicitly defined using rules.

Like GUMS, UMT does not draw assumptions itself, but accepts and stores new assertions about the user which are provided by the application system (depending on the reliability of these assertions, they can be regarded as invariable *premises* or as (later still retractable) *assumptions*. Stereotypes that become activated due to the new assertions about the user add still more assumptions (namely the feature-value pairs describing the characteristic of the respective user subgroups). Some of these assumptions may possibly be contradictory. UMT then applies all inference rules (including the contradiction detection rules) to the set of premisses and assumptions, and records the inferential dependencies.

A reason maintenance component in the system then determines all possible user models, which are all consistent sets of assertions containing the premises, a subset of the assumptions, and all inferences derived from these premises and assumptions. The Current User Model will be selected from the set of possible user models using preference criteria (such as that assumptions which were reported to the system by the application have a higher weight than assumptions from stereotypes). If inconsistencies with new information from the application is later detected (or if the application disconfirms advice from UMT concerning the user), the assumptions on which the offending assertion was based can be easily detected (since inferential dependencies become recorded), and the set of possible user models will be revised and re-evaluated to find the new Current User Model.

7.4 BGP-MS

BGP-MS (Kobsa, 1990a; Kobsa, 1992), which is currently under development, offers a "partitioned" user model which allows for the representation of more than one type of assumption about the user, notably the user's beliefs, goals and abilities. Assumptions about the user may include the user's presumed assumptions about other agents, like e.g. the user's beliefs about the system's goals (arbitrary nesting is possible). Emphasis is being put on distinguishing whether assumptions are privately held or shared between the user and the system.

The stereotypes which BGP-MS provides can also be ordered in an arbitrary hierarchy, with inheritance, stereotype activation rules and independent retraction rules being supported (an optional set of predefined rules is available for the definition of activation and retraction conditions). Global parameters allow the user model developer to fine-tune the operation of the stereotype mechanism, for instance by determining its precision, its recall, and the frequency of user reclassification.

Assumptions about the user can be communicated to BGP-MS in a first-order language with modal operators (if possible, they will become translated into representationally weaker but computationally more efficient internal representations, like SB-ONE (Kobsa, 1991) and KN-PART (Fink & Herrmann, 1993)). The deduction mechanism OTTER (McCune, 1990) is available for drawing inferences from initial assumptions. Strong emphasis is put on assisting the user model developer in the definition of the user-modeling knowledge pertaining to the application domain. Graphics-based interfaces will be available for the definition of the stereotype hierarchies, their contents, and the activation and retraction rules.

Like all other shell systems, BGP-MS also requires that assumptions about the user must be made by the application system. In addition, though, it offers a library of domain-independent assumptions about the user that can be drawn when certain communication acts occur in the interaction between the user and the system (Kobsa, 1983; Kutter, 1993). The application must then only notify BGP-MS about the communication act that took place, and the resulting domain-independent inferences will be made by the shell system and become entered into the user model. For instance, if the system communicates to the user that it declines his or her goal *p* and informs BGP-MS that this communication act took place, then BGP-MS will draw the assumption that from now on it is mutually believed that the user wants *p*, and that the system does not want *p* to be the case. These communication acts are independent of the interaction language or medium that is employed by the user and the system.

7.5 User model servers

A complementary yet partly competing proposal to user modeling shell systems are user model servers. User modeling shell systems, when filled with application-dependent user modeling knowledge, become part of the application, i.e. they receive information about the user from this application only and supply only this application with assumptions about the user. In contrast, user model servers would be centralized user modeling components for more than one application (possibly for all applications with which the user interacts). It seems that the capabilities of user model servers would then be restricted to domain-independent (i.e., fairly shallow) user modeling, unless they incorporate all the domain-dependent user modeling knowledge which would be necessary if each of these applications would employ a user modeling shell system.

To date, the idea of user model servers has not been worked out very thoroughly. The only exception is (Orwant, 1991) who envisages a central “personalization server”, which constantly gathers information about computer users from their everyday workplace activities. Independently, client systems that want to adapt to users make requests to this server for information about these users. A success or failure feedback report of the client systems is not yet foreseen, but could probably be added.

8. Utility of user models

Except for Rich’s (Rich, 1979a) empirical evaluation of her library recommendation system that bases its advice concerning books to read on a model of its current user, hardly any verification of the utility of user models was performed during the eighties. Fortunately, the concern for empirical questions has considerably increased in the last few years. Several investigations into the usefulness of user models in interactive software systems have been carried out recently, including the following:

- *Context-sensitive help in a WYSIWYG text editor* (Hirschmann, 1990; Krause et al. 1993): it was shown that the adaptation of help menus in a WYSIWYG text editor to the user's current error improved significantly the task performance, and decreased the redundancy and the error rate.

- *Navigational help in a hypertext system* (Kaplan et al., 1993): it was shown that navigational aid based on experiences with previous users and information about the current user's goal can significantly reduce the time searching for information.

- *Adaptive hypertext* (Boyle & Encarnacion, 1994): it was shown that the comprehension and search speed were significantly improved in a hypertext system that adapted its text to the user's level of expertise.

- *Personal news filter* (Jennings & Higuchi, 1993): it was shown that the precision and recall of an electronic news filter that was based on a connectionist user model was satisfactory for many users.

These positive results in several application areas for user modeling are contrasted by possibly negative findings in the area of preference modeling. (Allen, 1990), who had subjects predict other people's interests in newspaper articles, claims that the outcome of his experiments casts doubt on whether information preferences can be predicted at all by people, let alone by computer systems.

The question of whether user models and adaptive interactive computer systems offer sufficient enough advantages for their users such that the costs for their development are outweighed by their benefits can only be empirically determined. Further empirical work to this end is thus urgently needed. The first results for various types of applications were certainly promising, but at the moment one should not over-generalize them. Possibly results will be available in the future concerning the classes of applications which are amenable to user modeling, and application classes for which adaptivity is not so beneficial.

9. Guidelines to prevent potential abuses

Thus far, user modeling has been almost exclusively used for making computer systems more cooperative. Of course, user models could also support a system in pursuing non-cooperative interests. A small example is HAM-ANS (Jameson & Wahlster, 1982) which, in its non-cooperative mode, will deliberately misguide the user in order to "positively" influence his/her decisions concerning particular hotel rooms to book.

While user modeling is certainly an important contribution to making computers accessible to casual users (particularly to the general public), it is also subject to potential misuse like many other new technologies. Certain guidelines should therefore be observed both in research and in practical applications, in order to resist these dangers already in an early stage (see (Kobsa, 1990b) for a more detailed discussion):

- Users should be made aware of the fact that a computer system contains a user modeling component. In many applications, the presence of such a component may not be obvious, particularly not to casual users. The user's awareness of the user modeling abilities of a system is however a necessary prerequisite in order that he or she can decide whether or not to consent to being modeled by the system.

- Users should be instructed that computer systems might make errors. User modeling components draw mostly *assumptions* about the user, which may not necessarily be correct. User modeling therefore inherently involves the risk of misunderstandings (as is the case in normal communication between people). It is therefore only prudent to encourage the user to guard against possible misconceptions of the system, as he or she is anyway used to do when communicating with other people.

- Users should be instructed that a computer system might pursue non-cooperative interests. User models may not only be employed by a computer system for supporting the users' goals in a cooperative way, but also for influencing users according to the interests, e.g., of some company. If one does not regard a computer system as an objective expert, but rather as a representative of a concern, then one can rely on the strategies which one normally uses to protect oneself against the influence of other people.

- Users should have the possibility to inspect and, if necessary, also to change their user models. It can be anticipated, however, that in many applications user models will often comprise an enormous amount of fairly trivial assumptions, which are moreover difficult to translate into ordinary English. It will therefore often be difficult to render user models comprehensible to their owners.

- If technically possible, users should be able to "switch off" a user modeling component if he or she does not consent to being modeled. It is by far not clear, however, whether it is possible to separate the user modeling component from the remaining system in such a way that the user can in fact relinquish it. In many application domains, user modeling is not only a prerequisite for a system's cooperativeness, but even for its ability to conduct a coherent intelligent interaction at all.

- Long-term user characteristics should be modeled with caution since their misuse is probably more serious than the misuse of transient user characteristics. Moreover, long-term user characteristics are often regarded as more "personal" than short-term user characteristics.

- Goal and plan recognition should not be used for screening the user as to whether or not his/her goals in accessing a computer system are admissible.

- Results in user modeling research should be made accessible to the general public, which will eventually be affected by them.

References

R. B. Allen (1990): User Models: Theory, Method, and Practice. International Journal of Man-Machine Studies, 32:511-543.

D. E. Appelt and M. E. Pollack (1992): Weighted Abduction for Plan Ascription. User Modeling and User-Adapted Interaction, 2:1-25.

D. Benyon (1993): Adaptive Systems: A Solution to Usability Problems. User Modeling and User-Adapted Interaction 3:65-87.

F. Bodendorf, H.-G. Lindner, and H. Linß (1990): Benutzermodellierung mit Hilfe Neuronaler Netze: Ein Prototyp fur OS/2-Nutzer. Arbeitspapier 4/1990, Abteilung Wirtschaftsinformatik, Universität Erlangen-Nürnberg, Erlangen, Germany.

A. M. Botman, A. J. Kok, and L. Siklossy (1989): Methods for User Modelling in an Intelligent Interface for Data Retrieval. Unpublished Manuscript, Department of Mathematics and Computer Science, Free University, Amsterdam, Netherlands.

C. D. B. Boyle and Antonio O. Encarnacion (1994): An Adaptive Hypertext Reading System. To appear in User Modeling and User-Adapted Interaction.

G. Brajnik, G. Guida, and C. Tasso (1987): User Modeling in Intelligent Information Retrieval. Information Processing & Management, 23:305-320.

G. Brajnik and C. Tasso (1992): A Flexible Tool for Developing User Modeling Applications with Nonmonotonic Reasoning Capabilities. Proceedings of the Third International Workshop on User Modeling, Dagstuhl, Germany, 42-63.

D. N. Chin (1989): KNOME: Modeling what the User Knows in UC. In A. Kobsa and W. Wahlster (eds.): User Models in Dialog Systems, pp. 74-107. Springer, Berlin, Heidelberg.

F. De Rosis, S. Pizzutilo, A. Russo, D. C. Berry, and F. J. Nicolau Molina (1992): Modeling the User Knowledge by Belief Networks. User Modeling and User-Adapted Interaction, 2:367-388.

R. Eller and S. Carberry (1992): A Meta-Rule Approach to Flexible Plan Recognition in Dialogue. User Modeling and User-Adapted Interaction, 2:7-53.

G. Eydner and H. Vergara, 1993: PROTUM: A User Modeling Truth Maintenance System Based on Prolog and KN-PART (in German). Master thesis, Dept. of Information Science, University of Konstanz, Germany.

T. W. Finin. GUMS (1989): A General User Modeling Shell. In A. Kobsa and W. Wahlster (eds.): User Models in Dialog Systems, pp. 411-430. Springer, Berlin, Heidelberg.

J. Fink and M. Herrmann (1993): KN-PART - Ein Verwaltungssystem zur Benutzermodellierung mit prädikatenlogischer Wissensrepräsentation. WIS Memo Nr. 5, WG Knowledge-Based Information Systems, Dept. of Information Science, University of Konstanz, Germany.

A. Hirschmann (1990): Das Hilfesystem MATHILDE. PhD Thesis, Philosophische Fakultät, Universität Regensburg, Regensburg, Germany.

X. Huang, G. I. McCalla, J. E. Greer, and E. Neufeld (1991): Revising Deductive Knowledge and Stereotypical Knowledge in a Student Model. User Modeling and User-Adapted Interaction, 1:87-115.

A. Jameson and W. Wahlster (1982): User Modelling in Anaphora Generation: Ellipsis and Definite Description. In Proc. of the 5th European Conference on Artificial Intelligence, pp. 222-227, Orsay, France.

A. Jennings and H. Higuchi (1993). A User Model Neural Network for a Personal News Service. User Modeling and User-Adapted Interaction 3:1-25.

C. Kaplan, J. Fenwick and J. Chen (1993): Adaptive Hypertext Navigation Based on User Goals and Context. User Modeling and User-Adapted Interaction 3:193-220.

R. Kass and T. Finin (1988): Modeling the User in Natural Language Systems. Computational Linguistics 14(3):5-22.

R. Kass (1991): Building a User Model Implicitly from a Cooperative Advisory Dialog. User Modeling and User-Adapted Interaction, 1:203-258.

R. Kass and I. Stadnyk (1992): Using User Models to Improve Organizational Information. In Proceedings of the 3rd International Workshop on User Modeling, Dagstuhl, Germany.

J. Kay (1990): um: A Toolkit for User Modelling. In Proc. of the Second International Workshop on User Modeling, pp. 1-11, Honolulu, HI.

A. Kobsa (1983): Präsuppositionsanalyse zum Aufbau von Dialogpartnermodellen. Conceptus, 17(40/41):165-179.

A. Kobsa (1985): Benutzermodellierung in Dialogsystemen. Springer, Berlin, Heidelberg.

A. Kobsa and W. Wahlster, eds. (1988): Computational Linguistics 14(3): Special Issue on User Modeling. MIT Press, Cambridge, MA.

A. Kobsa and W. Wahlster (1989): User Models in Dialog Systems. Springer, Berlin, Heidelberg.

A. Kobsa (1990a): Modeling the User's Conceptual Knowledge in BGP-MS, a User Modeling Shell System. Computational Intelligence, 6:193-208.

A. Kobsa (1990b): User Modeling in Dialog Systems: Potentials and Hazards. AI & Society: The Journal of Human and Machine Intelligence, 4:214-231.

A. Kobsa (1991): Utilizing Knowledge: The Components of the SB-ONE Knowledge Representation Workbench. In: J. Sowa, ed.: Principles of Semantic Networks: Explorations in the Representation of Knowledge. San Mateo, CA: Morgan Kaufmann.

A. Kobsa (1992): Towards Inferences in BGP-MS: Combining Modal Logic and Partition Hierarchies for User Modeling (Preliminary report). In Proceedings of the 3rd International Workshop on User Modeling, Dagstuhl, Germany, pp. 35-41.

A. Kok (1991): A Review and Synthesis of User Modeling in Intelligent Systems. The Knowledge Engineering Review, 6:21-47.

J. Krause, A. Hirschmann and E. Mittermaier (1993): The Intelligent Help System COMFOHELP: Towards a Solution of the Practicability Problem for User Modeling and Adaptive Systems. User Modeling and User-Adapted Interaction 3:249-282.

O. Kutter (1993): Aufbau eines Benutzermodells durch Analyse von Kommunikationsakten. Projektarbeitsbericht, AG Wissensbasierte Informationssysteme, Informationswissenschaft, Universität Konstanz.

J. Mayfield (1992): Controlling Inference in Plan Recognition. User Modeling and User-Adapted Interaction, 2:83-115.

W. W. McCune (1990): OTTER 2.0 Users Guide. Technical Report ANL-90/9, Argonne National Laboratory, Mathematics and Computer Science Division, Argonne, IL.

K. Morik (1985): User Modelling, Dialog Structures, and Dialog Strategy in HAM-ANS. In Proc. of the 2nd EACL, pp. 268-273, Geneva, Switzerland.

K. Morik & C.-R. Rollinger: The Real Estate Agent: Modeling Users by Uncertain Reasoning. AI Magazine 6, 44-52

E. Nessen (1989): SC-UM: User Modeling in the Sinix Consultant. Applied Artificial Intelligence, 3:33-44.

A. F. Norcio and J. Stanley (1989): Adaptive Human-Computer Interfaces: A Literature Survey and Perspective. IEEE Transactions on Systems, Man, and Cybernetics, 19: 399-408.

J. L. Orwant (1991): The Doppelganger User Modelling System. In Proc. of the IJCAI Workshop W4: Agent Modelling for Intelligent Interaction, pp. 164-168, Sydney, Australia.

C. L. Paris (1989): The Use of Explicit User Models in a Generation System for Tailoring Answers to the User's Level of Expertise. In A. Kobsa and W. Wahlster (eds.): User Models in Dialog Systems, pp. 200-232. Springer, Berlin, Heidelberg.

B. Raskutti and I. Zukerman (1991): Generation and Selection of Likely Interpretations During Plan Recognition in Task-Oriented Consultation Systems. User Modeling and User-Adapted Interaction, 1:323-353.

E. Rich (1979a): Building and Exploiting User Model. PhD thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

E. Rich (1979b): User Modeling via Stereotypes. Cognitive Science, 3:329-354.

E. Rich (1989): Stereotypes and User Modeling. In A. Kobsa and W. Wahlster (eds.): User Models in Dialog Systems, pp. 35-51. Springer, Berlin, Heidelberg.

E. Shifroni and B. Shanon (1992): Interactive User Modeling: An Integrative Explicit-Implicit Approach. User Modeling and User-Adapted Interaction 2:287-330.

G. Sutcliffe and A. C. Old (1987): Do Users Know They Have User Models? Some Experiences in the Practice of User Modelling. In H.-J. Bullinger and B. Shakel (eds.): Human-Computer Interaction: INTERACT'87, pp. 35-41. North-Holland, Amsterdam.

P. van Arragon (1991): Modeling Default Reasoning Using Defaults. User Modeling and User-Adapted Interaction, 1:259-288.

D. Wu (1991): Active Acquisition of User Models: Implications for Decision-Theoretic Dialog Planning and Plan Recognition. User Modeling and User-Adapted Interaction, 1:149-172.