



## Generic User Modeling Systems

ALFRED KOBSA

*Department of Information and Computer Science, University of California, Irvine,  
CA 92697-3425, U.S.A., E-mail: kobsa@uci.edu*

(Received: 13 February 2000; in final form 20 July 2000)

**Abstract.** The paper reviews the development of generic user modeling systems over the past twenty years. It describes their purposes, their services within user-adaptive systems, and the different design requirements for research prototypes and commercially deployed servers. It discusses the architectures that have been explored so far, namely shell systems that form part of the application, central server systems that communicate with several applications, and possible future user modeling agents that physically follow the user. Several implemented research prototypes and commercial systems are briefly described.

**Key words:** user models, tool systems, user model shells, user model servers, user model agents

### 1. Early History of User Modeling Shell Systems

User modeling is usually traced back to the works of Allen, Cohen and Perrault (e.g., Perrault et al., 1978; Cohen and Perrault, 1979; Allen, 1979) and Elaine Rich (Rich, 1979a, 1979b). For a ten-year period following this seminal research, numerous application systems were developed that collected different types of information about, and exhibited different kinds of adaptation to, their current users. Many of these early user-adaptive applications have been reviewed in Morik (1982), Kobsa and Wahlster (1989) and McTear (1993).

In this early work, the user modeling was performed by the application system, and often no clear distinction could be made between system components that served user modeling purposes and components that performed other tasks. From the mid-eighties onwards, such a separation was increasingly made (e.g., Kobsa, 1985; Sleeman, 1985; Kass, 1988; Allgayer et al., 1989), but no efforts are reported on rendering the user modeling component reusable for the development of future user-adaptive systems.

In 1986, Tim Finin published his ‘General User Modeling System’ GUMS (Finin and Drager, 1986; Finin, 1989). This software allowed programmers of user-adaptive applications the definition of simple stereotype hierarchies, and for each stereotype, of Prolog facts describing stereotype members and rules prescribing the system’s reasoning about them. At runtime, GUMS accepts and stores new facts about the user which are provided by the application system, verifies the consistency of a new fact with currently held assumptions, informs the application about recognized inconsistencies, and answers queries of the application concerning the

currently held assumptions about the user. Albeit GUMS was never used together with an application system, it set the framework for the basic functionality of future ‘general’ (i.e., application-independent) user modeling systems, namely the provision of selected user modeling services at runtime that can be configured during development time. When filled by the developer with application-specific user modeling knowledge, these systems would serve as separate user modeling components in the respective applications.

Kobsa (1990) seems to be the first author who used the term ‘user modeling shell system’ for such kinds of software tools. The term ‘shell system’, or ‘shell’ for short, was thereby borrowed from the field of Expert Systems. There, van Melle (1982) and Buchanan and Shortliffe (1984) had condensed the experiences made with the medical expert system MYCIN (Shortliffe, 1976) into EMYCIN (‘Essential’ MYCIN), an ‘empty’ expert system that had to be filled with domain-specific rules for deployment as a ‘real’ expert system. Commercial expert system shells like Knowledge Craft (Knowledge Craft, 1988), KEE (Intellicorp) and ART (Clayton, 1985) thereafter became very popular in the late seventies and early eighties. The general aims that underlay the drift to user modeling shell systems, namely software decomposition and abstraction to support modifiability and reusability, is of course much older than expert system shells.

## 2. Academic Developments

### 2.1. EXAMPLE SYSTEMS

In the early nineties, several research groups in different countries independently started condensing basic structures and processes into user modeling shells that they believed were important for user-adaptive application systems. Major shell systems developed during this time include the following ones:

**UMT** (Brajnik and Tasso, 1994) allows the user model developer the definition of hierarchically ordered user stereotypes, and of rules for user model inferences as well as contradiction detection. Information about the user that is received from the application can be classified as invariable premises or (later still retractable) assumptions. After the firing of all applicable inference rules and the activation of all applicable stereotypes, contradictions between assumptions are sought and various resolution strategies applied (‘truth maintenance’).

**BGP-MS** (Kobsa and Pohl, 1995; Pohl, 1998) allows assumptions about the user and stereotypical assumptions about user groups to be represented in a first-order predicate logic. A subset of these assumptions is stored in a terminological logic. Inferences across different assumption types (i.e., types of modals) could be defined in a first-order modal logic. The system can be used as a network server with multi-user and multi-application capabilities.

**DOPPELGÄNGER** (Orwant, 1995) is also a user modeling server that accepts information about the user from hardware and software sensors. Techniques for generalizing and extrapolating data from the sensors (such as beta distributions, linear prediction, Markov models, and unsupervised clustering for stereotype formation) are put at the disposal of user model developers. Users can inspect and edit their user models.

**TAGUS** (Paiva and Self, 1995) represents assumptions about the user in first-order formulas, with meta-operators expressing the assumption types. The system allows for the definition of a stereotype hierarchy and contains an inference mechanism, a truth maintenance system, and a diagnostic subsystem that includes a library of misconceptions. It also supports the ‘simulation of the user’ through forward-directed inferences on the basis of the user model, and the diagnosis of unexpected user behavior.

**um** (Kay, 1995) is a toolkit<sup>1</sup> for user modeling that represents assumptions about the user’s knowledge, beliefs, preferences, and other user characteristics in attribute-value pairs. Each piece of information is accompanied by a list of evidence for its truth and its falsehood. The source of each piece of evidence, its type (observation, stereotype activation, rule invocation, user input, told to the user) and a time stamp is also recorded.

Other shell systems include Huang et al. (1991), Vergara (1994), Kono et al. (1994), Blank (1996) and Machado et al. (1999).

## 2.2. CHARACTERISTICS

The decisions as to what important structures and processes should go into user modeling shell systems were mostly based on intuition and/or experience of the shell developers through prior work on user-adaptive systems. Efforts to put these decisions on more empirical grounds were seemingly only made in Kleiber (1994) and Pohl (1998, 1999). Even these authors however merely identified individual user-adaptive application systems in the literature that would have profited from the functionality of their own developed shell system, rather than performing a comprehensive review of prior user-adaptive systems, and determining current and predicting future system needs.

In an attempt to extend the *de facto* definition of user modeling shells introduced by GUMS and to avoid characterizing user modeling shell systems via internal structures and processes, Kobsa (1995) listed the following frequently-found *services* of such systems:

<sup>1</sup> From the point of view of the application system, um was more a library of user modeling functions than an independent user modeling component. It therefore is not a user modeling shell in a strict sense.

- the representation of assumptions about one or more types of user characteristics in models of individual users (e.g. assumptions about their knowledge, misconceptions, goals, plans, preferences, tasks, and abilities);
- the representation of relevant common characteristics of users pertaining to specific user subgroups of the application system (the so-called stereotypes);
- the classification of users as belonging to one or more of these subgroups, and the integration of the typical characteristics of these subgroups into the current individual user model;
- the recording of users' behavior, particularly their past interaction with the system;
- the formation of assumptions about the user based on the interaction history;
- the generalization of the interaction histories of many users into stereotypes;
- the drawing of additional assumptions about the current user based on initial ones;
- consistency maintenance in the user model;
- the provision of the current assumptions about the user, as well as justifications for these assumptions;
- the evaluation of the entries in the current user model, and the comparison with given standards.

This characterization of user modeling shell systems is observational only, and it is again not backed up by a comprehensive analysis of what user modeling services are actually demanded from current and future user-adaptive systems.

Several requirements for user modeling shell systems were regarded as important, including the following ones.

#### *Generality, including domain independence*

Shell systems were required to be usable in as many application and content domains as possible, and within these domains for as many user modeling tasks as possible. They were therefore expected to provide as many services as possible. 'Concessions' in this regard were only made for shell systems in student-adaptive tutoring systems (Huang et al., 1991; Kono et al., 1994; Paiva and Self, 1995; Machado et al., 1999), which were expected to be usable for teaching different subject matters, but not for additional application domains besides educational ones.

#### *Expressiveness*

Shell systems were expected to be able to express as many types of assumptions about the user as possible at the same time. This not only included the different types of propositional attitudes mentioned above, but also all sorts of reflexive assumptions concerning the user and the system (see Kobsa, 1989; Taylor et al., 1996), plus uncertainty and vagueness in these assumptions.

### *Strong Inferential Capabilities*

Shell systems were expected to perform all sorts of reasoning that are traditionally distinguished in artificial intelligence and formal logic, such as reasoning in a first-order predicate logic, complex modal reasoning (e.g., reasoning about types of modalities), reasoning with uncertainty, plausible reasoning when full information is not available, and conflict resolution when contradictory assumptions are detected.

The rationale for assigning so much importance to these requirements rather than others lies in the affinity of user modeling research of those days to artificial intelligence, natural-language dialog (Kobsa and Wahlster, 1989; Zukerman and Litman, 2001), and intelligent tutoring (Kass, 1989; Sleeman and Brown, 1982). User modeling shells were expected to support the complex assumptions and complex reasoning about the user that had been identified in these domains, and additionally to be usable in a wide range of other domains as well. When in the mid-nineties user-adaptive application systems shifted towards different domains with less demanding user modeling requirements (like user-adaptive learning environments (Brusilovsky et al., 1998; Brusilovsky, 2001) and user-tailored web sites (Kobsa et al., 2001)), such complex user modeling and reasoning capabilities became redundant. Moreover, commercial applications necessitate additional services and requirements that were largely lacking in the research-oriented shells of these days (see Section 3.2).

Another observation is that virtually all of these user modeling shell systems adhered to a ‘mentalistic’ paradigm (Pohl, 1997). They modeled ‘propositional attitudes’ (Russel, 1954) of the user like his or her knowledge, beliefs, goals, preferences and interests. User behavior was only used as an information source for drawing assumptions about users’ propositional attitudes, and not regarded as a phenomenon that should be analyzed and modeled *per se*. DOPPELGÄNGER (Orwant, 1995), and more recently LaboUr (Pohl, 1997), were the only user modeling shells that provided means for detecting patterns in user behavior.

The ‘academic’ user modeling shells of the early nineties did not enjoy much distribution, even not in the research community. It seems that BGP-MS was the only user modeling shell that was used outside of the institution at which it was originally developed, and even for this system there exist only few reports on extensive external usage. However, some of the ideas that were first explored in these prototypical systems (particularly the stereotype approach and the client-server architecture) have since made it into commercial user modeling software.

### **3. The Beginning of a Commercial Boom**

In the late 1990s, the value of web personalization was increasingly recognized in the area of electronic commerce (Hof et al., 1998; Allen et al., 1998; Cooperstein et al., 1999; Hagen et al., 1999). Web personalization allows product offerings, sales promotions, product news, ad banners, etc. to be targeted to each individual user,

taking the user's navigation data, purchase history and other previous interactions with the electronic merchant into account. From a more general perspective, personalization allows the relationship with customers on the Internet to migrate from anonymous mass marketing and sales to 'one-to-one' marketing (Peppers and Rogers, 1993, 1997).

User modeling and user modeling shells can play an important role in this endeavor (Fink and Kobsa, 2000). Around 50 different tool systems for web personalization with very different capabilities are currently being advertised (which does however not necessarily mean that they are also available).

### 3.1. EXAMPLE SYSTEMS

Major current tool systems for web personalization include the following ones:

**Group Lens** (Net Perceptions, 2000) employs various collaborative filtering algorithms (Breese et al., 1998; Herlocker et al., 1999) for predicting users' interests. Predictions are based on ratings explicitly provided by users (e.g., in on-line forms), implicit ratings derived from navigational data (e.g., products that the online customer viewed and products that have been put into the shopping cart), and data from transaction history (e.g., products purchased in the past).

**LikeMinds** (Andromedia, 2000) is similar to Group Lens. Major differences include a more modular architecture, better load distribution, ODBC support, and slightly different input types (namely purchase data, navigational data, explicitly stated user preferences, and pre-defined product similarities).

**Personalization Server** (ATG, 2000) allows for the definition of rules that assign individual users to one or more user groups based on their demographic data (e.g., gender and age), information about the user's system usage, and information about the user's software, hardware and network environments. Rules can also be defined for inferring individual assumptions about the user from his or her navigation behavior, and for personalizing the content of web pages. The operation of Personalization Server thus follows very much the 'stereotype approach' from classical user modeling research (Rich, 1979b; 1989).

**Frontmind** (Manna) provides a rule-based development, management, and simulation environment for personalized information and personalized services on the web. It distinguishes itself from other rule-based products like Personalization Server by having Bayesian networks for modeling users' behavior integrated into its personalization framework.

**Learn Sesame** (Open Sesame, 2000; Bowne, 2000) allows for the definition of a domain model consisting of objects, object attributes, and event types. It accepts information about the user from an application, categorizes this information based on the domain model, and tries to detect recurrent patterns, correlations and similarities through incremental clustering. Interesting observations are then reported back to the application.

Other recent commercial user modeling systems are discussed in Fink and Kobsa (2000). In addition, there exist a number of comprehensive commercial e-commerce environments with built-in user modeling components that however cannot be separated from the rest of the system (e.g., in Broadvision's 'One-To-One' (Broadvision, 2000) and Microsoft's 'Site Server' (Microsoft, 2000)).

### 3.2. CHARACTERISTICS

A central characteristic of most current commercial systems is their client-server architecture. Here, user modeling systems are not functionally integrated into the application but communicate with the application through inter-process communication and can serve more than one user/client applications at the same time. This has generally not been the case for the academic user modeling shell systems mentioned in Section 2, even though Kobsa et al. (1994), Orwant (1995), Pohl and Höhle (1997) and Machado et al. (1999) represent important steps in this direction.

A client-server based architecture provides a number of advantages compared to embedded user modeling components (see Fink and Kobsa (2000) and Billsus and Pazzani (2000) for a more comprehensive discussion):

- Information about the user is maintained in a central or virtually integrated repository and put at the disposal of more than one application at the same time.
- User information acquired by one application can be employed by other applications, and vice versa.
- Information about users is stored in a non-redundant manner. The consistency and coherence of information gathered by different applications can be achieved more easily.
- Information about user groups, either available *a priori* as stereotypes (e.g., Rich, 1979a; 1979b; 1981; 1989) or dynamically calculated as user group models (e.g. Orwant, 1995; Paliouras et al., 1999), can be maintained with low redundancy.
- Methods and tools for system security, identification, authentication, access control and encryption can be applied for protecting user models in user modeling servers (Schreck, 2000).
- Complementary user information that is dispersed across the enterprise (e.g., demographic data from client databases, past purchase data from transactional systems, user segmentations from marketing research) can be integrated more easily with the information in the user model repository.

In addition, many more general advantages of centralized systems design (e.g., centralized user modeling servers relieve clients from user modeling tasks and can take advantage of powerful hardware resources), as well as disadvantages (e.g., necessity of a network connection, potential central point of failure), also apply.

We will henceforth use the term ‘user modeling server’ to refer to these commercial user modeling systems. Since the term ‘shell’ has meanwhile become outdated, we will use the term ‘(generic) user modeling system’ to refer to any generic system which offers user modeling services at runtime that can be configured at the time of development.

Commercial user modeling servers must support services that to some extent are different from those that were expected from academic user modeling shells. Examples of such new user modeling services include:

*Comparisons of different users’ selective actions.* In certain application areas, users’ choices cannot very well be reconstructed by step-wise reasoning processes, but only by reference to vaguer concepts like users’ taste, personality and lifestyle (examples include the selection of music, books, clothes, and restaurants). In such domains it was found useful to match users’ selective actions (buying items, bookmarking them, putting them into a shopping cart, rating them highly) with those of other users, and to predict users’ future selective actions based on those of the most similar other users. Many current commercial user modeling servers therefore support the comparison of different users’ action patterns, using ‘collaborative’ (aka clique-based) filtering algorithms (Breese et al., 1998; Herlocker et al., 1999).

*Import of external user-related information.* Many businesses already own customer and marketing data, and usually want to integrate these into user modeling systems when starting with personalized e-commerce. To access external data, ODBC interfaces or native support for a wide variety of databases are a must. Due to legacy business processes and software, external user-related information often continues to be updated in parallel to the e-commerce application, and therefore needs to be continually integrated at reasonable costs and without impairing the response time.

*Privacy support.* Company privacy policies, industry privacy norms and conventions, national and international privacy legislation, and privacy-supporting software tools and service providers are slowly emerging. While this is not yet the case to date, user modeling servers should ideally support any company privacy policy that complies with these constraints and be able to take advantage of all major privacy software and services that will then be available on the market.

Current commercial user modeling servers are very much behavior-oriented. Observed user actions or action patterns often lead directly to adaptations, without an explicit representation of the user characteristics (interests, knowledge, plans, etc.) that probably underlie this behavior and justify these adaptations. Making these assumptions explicit would allow the user modeling system to employ them for purposes other than only those for which they were recorded, as was the case for the classical user modeling shells.



Current user modeling servers also rate rather poorly on the dimensions of generality, expressiveness and inferential capabilities, all of which were regarded as important for the academic user modeling shells. In many cases, they are quite domain-dependent, their user model representation is very much intertwined with processing considerations, and they can only be used for limited personalization purposes. However, these classical quality characteristics are not regarded as important for commercial user modeling servers anymore. For them, the following dimensions seem to be considerably more significant, along which in return the classical user modeling shells rate very poorly.<sup>2</sup>

*Quick adaptation.* In order to bond users with web shops, adaptations should already take place for first-time visitors during their usually relatively short initial interaction. Several commercial user modeling systems can therefore select between more than one modeling and personalization methods with different degrees of complexity, depending on the amount of data that is already available about the user.

*Extensibility.* Current user modeling servers support a number of user model acquisition and personalization methods, but companies may want to integrate their own methods or third-party tools. Application Programmer Interfaces (APIs) and interfaces that allow for the (possibly bi-directional) exchange of user information between user-modeling tools are therefore required.

*Load balancing.* Under real-world conditions, user model servers will experience dramatic changes in their average load. Noticeable response delays or even denials of requests should only occur in emergency situations. User modeling servers should be able to react to increases in load through load distribution (ideally with CORBA-based components that can be distributed across a network of computers) and possibly through less thorough (and thereby less time-consuming) user model analyses.

*Failover strategies.* Centralized architectures need to provide fallback mechanisms in case of a breakdown.

*Transactional Consistency.* Parallel read/write on the user model and abnormal process termination can lead to inconsistencies that must be avoided by carefully selected transaction management strategies (Fink, 1999).

#### **4. The Future of Generic User Modeling Systems**

It goes without saying that predictions concerning the future of user modeling systems are fairly speculative, due to the rapidly changing nature of computing and computing devices. Since personalization has already been demonstrated to benefit

<sup>2</sup> See Zukerman and Litman (2001) for a similar paradigm shift in the area of natural-language systems.

both the users and the providers of personalized services and since personalization is therefore going to stay, it is practically certain that generic tool systems that allow for the easy development and maintenance of personalized systems will be needed in the future as well.

The exact form which user modeling systems of the future will take on is however likely to be strongly influenced by many characteristics of system usage that are difficult to predict. Here are a few considerations concerning likely future avenues:

*Mobile user models.* Computing is increasingly becoming mobile, but in the near future the reliability of mobile networks (and possibly also their bandwidth) will still fall short of the demands imposed by the client-server architecture for user modeling systems, which requires permanent connectivity. Hence for ubiquitous computing<sup>3</sup> and ubiquitous information<sup>4</sup> scenarios, mobile user models (which we may cautiously dub ‘user model agents’) seem to be worth considering. These user model agents may reside on the server side and be replicated at the beginning of each interaction. Or, they may be true mobile agents and stay with the user all the time, either on his or her computing device or on a gadget that the user always wears (like a chip that can be read from a distance and is integrated in a plastic card, a wristwatch, or jewelry (Fink et al., 1997)).

*User models for smart appliances.* Personalization has so far been almost exclusively confined to computing systems. Recently, however, appliances are being offered that feature limited but very useful personalization. Examples include car radios with a chip card that both contains a security code and also stores the drivers’ preferences concerning pre-set stations, volume and tone, and traffic news. Electronic car keys exist that adjust the driver seat, the mirrors and the GPS system to the driver’s individual preferences when plugged into the ignition lock. While these are proprietary solutions with proprietary minuscule user models, it is likely that we will see far more examples of personalizable appliances in the future. Since people will not want to carry a small user model gadget for each and every personalized appliance, standardized solutions and hence the need for generic tool systems will soon arise.

*Multiple-purpose usage.* Information about the characteristics of individual users may not only be interesting for personalization purposes. Other possible applications include organizational directory services, skill inventory systems, and organizational or global expert-finding applications. Considerations concerning a central user model server versus a local user model agent can also be made with respect to these kinds of applications (Yimam and Kobsa, 2001), and it seems that in principle generic user model systems can be developed that

---

<sup>3</sup> The user carries a mobile information device wherever he or she goes.

<sup>4</sup> The user is able to conjure up his or her personal information environment anywhere, say, at every wall and desktop.

can support all these different usage purposes for people-related information. Basing the user model representation of such systems on standards that were developed for directory services (like LDAP, cf. Fink, 2001) would also help transfer user modeling systems from the area of proprietary developments into that of industry standards.

As a consequence of this plethora of different possible applications of information about users, it is unlikely that in the near future there will be a single or a small number of universal ‘user modeling pearl systems’ suitable for a large number of user modeling tasks, as has to some extent been the tacit hope in the late eighties and early nineties. Instead one can expect to find a wide variety of generic user modeling systems, each of which is going to support only a few of the very different future manifestations of personalization and other applications of information about the user. Privacy requirements, the need to include user information from legacy systems, and the need to exchange user information across different systems will however hopefully enforce some standardization, at least on the communication level.

### Acknowledgements

I would like to thank Peter Brusilovsky, Josef Fink, Judy Kay, Wolfgang Pohl and Ingrid Zukerman for their valuable comments on an earlier version of this paper.

### References

- Allen, C., Yaeckel, K. D. and Yaeckel, B.: 1998, *Internet World Guide to One-To-One Web Marketing*, John Wiley and Sons, New York.
- Allen, J. F.: 1979, A Plan-based approach to speech act recognition. *Technical Report 131/79*, Dept. of Computer Science, University of Toronto, Canada.
- Allgayer, J., Harbusch, K., Kobsa, A., Reddig, C., Reithinger, N. and Schmauks, D.: 1989, XTRA: A Natural-Language access system to expert systems. *International Journal of Man-Machine Studies* **31**, 161–195.
- Andromedia: 2000, LikeMinds. Andromedia, <http://www.andromedia.com/products/likeminds/index.html>
- ATG: 2000, Dynamo Product Suite, Art Technology Group, <http://www.atg.com/products/highlights>
- Billsus, D. and Pazzani, M. J.: 2000, User modeling for adaptive news access. *User Modeling and User-Adapted Interaction* **10**(2–3), Special Issue on Deployed User Modeling, 147–180.
- Blank, K.: 1996, *Benutzermodellierung für adaptive interaktive Systeme: Architektur, Methoden, Werkzeuge und Anwendungen*. Sankt Augustin, Germany: infix.
- Bowne: 2000, Bowne and Co. <http://www.bowne.com>
- Brajnik, G. and Tasso, C.: 1994, A shell for developing non-monotonic user modeling systems. *International Journal of Human-Computer Studies* **40**, 31–62.
- Breese, J., Heckerman, D. and Kadie, C.: 1998, Empirical analysis of predictive algorithms for collaborative filtering. *Proc. of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, San Francisco, pp. 43–52.

- Broadvision: 2000, Broadvision. [www.broadvision.com](http://www.broadvision.com)
- Brusilovsky: 2001, Adaptive hypermedia. *User Modeling and User-Adapted Interaction* **11**(1–2), 87–110 (this issue).
- Brusilovsky, P., Kobsa, A. and Vassileva, J. (eds.): *Adaptive Hypertext and Hypermedia*. Kluwer Academic Publishers, Dordrecht, Netherlands.
- Buchanan, B. G. and Shortliffe, E. H.: 1984, *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, MA.
- Clayton, B. D.: 1985, ART Programming Tutorial, Version 1.0. Inference Corporation, Los Angeles, CA.
- Cohen, P. R. and Perrault, C. R.: 1979, Elements of a Plan-based theory of speech acts. *Cognitive Science* **3**, 177–212.
- Cooperstein, D., Delhagen, K., Aber, A. and Levin, K.: 1999, *Making Net Shoppers Loyal*. Forrester Research, Cambridge, MA.
- Finin, T. W.: 1989, GUMS: A general user modeling shell. In: A. Kobsa and W. Wahlster (eds.), *User Models in Dialog Systems*. Springer-Verlag, Berlin, Heidelberg, pp. 411–430.
- Finin, T. W. and Drager, D.: 1986, A general user modeling system. *Proc. of the 6th Canadian Conference on Artificial Intelligence*, Montreal, Canada, pp. 24–29.
- Fink, J.: 1999, Transactional consistency in user modeling systems. In: J. Kay (ed.), *UM99 User Modeling: Proceedings of the Seventh International Conference*. Springer-Verlag, Wien New York. pp. 191–200.
- Fink, J.: 2001, *User Modeling Servers – Requirements, Design, and Implementation*. Ph.D. Thesis, Dept. of Mathematics and Computer Science, University of Essen, Germany (forthcoming).
- Fink, J. and Kobsa, A.: 2000, A review and analysis of commercial user modeling servers for personalization on the World Wide Web. *User Modeling and User-Adapted Interaction* **10**(2–3), Special Issue on Deployed User Modeling, 209–249.
- Fink, J., Kobsa, A. and Jaceniak, I.: 1997, Individualisierung von Benutzerschnittstellen mit Hilfe von Datenchips für Personalisierungsinformation. *GMD-Spiegel* **1/1997**, 16–17. <http://www.ics.uci.edu/~kobsa/papers/1997-GMD-kobsa.ps>
- Hagen, P., Manning, H. and Souza, R.: 1999, *Smart Personalization*. Forrester Research, Cambridge, MA.
- Herlocker, J., Konstan, J., Borchers, A. and Riedl, J.: 1999, An algorithmic framework for performing collaborative filtering. In: M. Hearts, F. Gey and R. Tong (eds.), *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, pp. 230–237. <http://www.cs.umn.edu/Research/GroupLens/Algs.pdf>
- Hof, R. Green, H. and Himmelstein, L.: 1998, Now it's YOUR WEB. *Business Week* (October 5), 68–75.
- Huang, X., McCalla, G. I., Greer, J. E. and Neufeld, E.: 1991, Revising deductive knowledge and stereotypical knowledge in a student model. *User Modeling and User-Adapted Interaction* **1**(1), 87–115.
- Intellicorp: Intellicorp. <http://www.intellicorp.com>
- Kass, R.: 1988, *Acquiring a Model of the User's Beliefs from a Cooperative Advisory Dialog*. Ph.D. Thesis, Dept. of Information and Computer Science, University of Pennsylvania, Philadelphia, PA.
- Kass, R.: 1989, Student modeling in intelligent tutoring systems – implications for user modeling. In: A. Kobsa and W. Wahlster (eds.), *User Models in Dialog Systems*. Springer-Verlag, Berlin, Heidelberg, pp. 386–410.

- Kay, J.: 1995, The um Toolkit for reusable, long term user models. *User Modeling and User-Adapted Interaction* 4(3), 149–196.
- Kleiber, U.: 1994, Erklärung in interaktiven Systemen und Unterstützungsmöglichkeiten durch das System BGP-MS. *WIS Memo 6*, WG Knowledge-based Information Systems, Department of Information Science, University of Konstanz, Germany.
- Knowledge Craft: 1988, Knowledge Craft, 3.2 Edn. Carnegie Group, Inc., Pittsburgh, PA.
- Kobsa, A.: 1985, *Benutzermodellierung in Dialogsystemen*. Springer Verlag, Berlin, Heidelberg.
- Kobsa, A.: 1989, A taxonomy of beliefs and goals for user models in dialog systems. In: A. Kobsa, and W. Wahlster (eds.), *User Models in Dialog Systems*. Springer-Verlag, Berlin, Heidelberg, pp. 52–68.
- Kobsa, A.: 1990, Modeling the user's conceptual knowledge in BGP-MS, a user modeling shell system. *Computational Intelligence* 6, 193–208.
- Kobsa, A.: 1995, Editorial. *Using Modeling and User-Adapted Interaction* 4(2), Special Issue on User Modeling Shell Systems, iii–v.
- Kobsa, A., Koenemann, J. and Pohl, W.: 2001, Personalized Hypermedia Presentation Techniques for Improving Customer Relationships. *The Knowledge Engineering Review*, forthcoming.
- Kobsa, A., Müller, D. and Nill, A.: 1994, KN-AHS: An adaptive hypertext client of the user modeling system BGP-MS. *Proceedings of the Fourth International Conference on User Modeling*, Hyannis, MA, 99–105. Reprinted in M. Maybury and W. Wahlster (eds.), (1998). *Intelligent User Interfaces*. Morgan Kaufman, San Mateo, CA, pp. 372–378.
- Kobsa, A. and Pohl, W.: 1995, The BGP-MS user modeling system. *User Modeling and User-Adapted Interaction* 4(2), 59–106.
- Kobsa A. and Wahlster, W. (eds.): 1989, *User Models in Dialog Systems*. Springer Verlag, Heidelberg, Berlin.
- Kono, Y., Okeda, M. and Mizoguchio, R.: 1994, THEMIS: A nonmonotonic inductive student modeling system. *Journal of Artificial Intelligence in Education* 5(3), 371–413.
- Machado, I., Martins, A. and Paiva, A.: 1999, One for all and all in one: A learner modelling server in a multi-agent platform. In: J. Kay (ed.), *UM99 User Modeling: Proceedings of the Seventh International Conference*, Springer-Verlag, Wien, New York.
- Manna: Manna. <http://www.mannainc.com>
- McTear, M. (ed.): 1993, *Artificial Intelligence Review* 7(3). Special issue on user modeling.
- Microsoft: 2000, Product and Technology Catalog. [www.microsoft.com/products](http://www.microsoft.com/products)
- Morik, K.: 1982, *Überzeugungssysteme der Künstlichen Intelligenz: Validierung vor dem Hintergrund linguistischer Theorien über implizite Äußerungen*. Niemeyer, Tübingen.
- Net Perceptions: 2000, Net Perceptions. <http://www.netperceptions.com>
- Open Sesame: 2000, Open Sesame. Bowne and Co., <http://www.opensesame.com>
- Orwant, J.: 1995, Heterogenous 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.
- Paliouras G., Karkaletsis, V., Papatheodorou, C. and Spyropoulos, C.: 1999, Exploiting learning techniques for the acquisition of user stereotypes and communities. In: J. Kay (ed.), *UM99 User Modeling: Proceedings of the Seventh International Conference*. Springer-Verlag, Wien, New York. pp. 169–178.
- Peppers, D. and Rogers, M.: 1993, *The One to One Future: Building Relationships One Customer at a Time*. Currency Doubleday, New York, N.Y.

- Peppers, D. and Rogers, M.: 1997, *Enterprise One to One: Tools for Competing in the Interactive Age*. Currency Doubleday, New York, N.Y.
- Perrault, C. R., Allen, J. F. and Cohen, P. R.: 1978, Speech acts as a basis for understanding dialogue coherence. *Report 78-5*, Department of Computer Science, University of Toronto, Canada.
- Pohl, W.: 1997, LaboUr – Machine learning for user modeling. In: M. J. Smith, G. Salvendy and R. J. Koubek (eds.), *Design of Computing Systems: Social and Ergonomic Considerations (Proceedings of the Seventh International Conference on Human-Computer Interaction)*. Elsevier, Amsterdam. pp. 27–30.
- Pohl, W.: 1998, *Logic-Based Representation and Reasoning for User Modeling Shell Systems*. Sankt Augustin, Germany: infix.
- Pohl, W.: 1999, Logic-based representation and reasoning for user modeling shell systems. *User Modeling and User-Adapted Interaction* **9**(3), 217–282.
- Pohl, W. and Höhle, J.: 1997, Mechanisms for flexible representation and use of knowledge in user modeling shell systems. In: A. Jameson, C. Paris and C. Tasso (eds.), *User Modeling: Proceedings of the Sixth International Conference*. Springer-Verlag, Wien, New York. pp. 403–414.
- Rich, E.: 1979a, *Building and Exploiting User Models*. Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.
- Rich, E.: 1979b, User modeling via stereotypes. *Cognitive Science* **3**, 329–354.
- Rich, E.: 1981, Users are individuals: Individualizing user models. *International Journal of Man-Machine Sciences* **18**, 199–214.
- Rich, E.: 1989, Stereotypes and user modeling. In: A. Kobsa, and W. Wahlster (eds.), *User Models in Dialog Systems*. Springer, Berlin, Heidelberg, pp. 35–51.
- Russel, B.: 1954, *Human Knowledge: Its Scope and Limits*. George Allen & Unwin, London, U.K.
- Schreck, J.: 2000, *Security and Privacy in User Models*. Dept. of Mathematics and Computer Science, University of Essen, Germany.
- Shortliffe, E. H.: 1976, *Computer-Based Medical Consultations: MYCIN*. North-Holland, New York.
- Sleeman, D.: 1985, UMFE: A user modelling front-end subsystem. *International Journal of Man-Machine Studies* **23**, 71–88.
- Sleeman, D. and Brown, J. S.: 1982, *Intelligent Tutoring Systems*. Academic Press, New York.
- Taylor, J. A., Carletta, J. and Mellish, C.: 1996, Requirements for belief models in co-operative dialogue. *User Modeling and User-Adapted Interaction* **6**(1), 23–68.
- van Melle, W.: 1982, *System Aids in Constructing Consultation Programs: EMYCIN*. UMI Research Press, Ann Arbor, MI.
- Vergara, H.: 1994, PROTUM: A Prolog based tool for user modeling. *WIS-Report 10*, WG Knowledge-Based Information Systems, Department of Information Science, University of Konstanz, Germany.
- Yimam, D. and Kobsa, A.: 2001, Expert finding systems for organizations: Problem and domain analysis and the DEMOIR approach. In: M. Ackerman, A. Cohen, V. Pipek and V. Wulf (eds.), *Beyond Knowledge Management: Sharing Expertise*, forthcoming.
- Zukerman, I. and Litman, D.: 2001, Natural language processing and user modeling: synergies and limitations. *User Modeling and User-Adapted Interaction* **11**(1–2), 129–158 (this issue).

**Author's Vita**

**Alfred Kobsa** (<http://www.ics.uci.edu/~kobsa>) is an Associate Professor at the Department of Information and Computer Science of the University of California, Irvine, CA and a Professor of Computer Science at the University of Essen, Germany. He was formerly a Director of the Institute FIT at GMD – German National Research Center for Information Technology, and an Associate Professor of Information Systems at the University of Konstanz, Germany. He received his Ph.D. in Computer Science from the Technical University of Vienna, Austria. Dr Kobsa's research focuses on user-adaptive information environments, user modeling, information visualization, expert finders, multimedia educational software, and user interfaces for handicapped and elderly people. He is the Editor-in-Chief of *User Modeling and User-Adapted Interaction*, an editorial board member of *World Wide Web* and *Universal Access in the Information Society*, and was the founding president of *User Modeling, Inc.*