Marquette University
e-Publications@Marquette

Management Faculty Research and Publications | Business Administration, College of

1-1-2000

The Distributed User Modeling Shell System (DUMSS): A Conceptual Framework for Eliciting User Models

Thawatchai Piyawat
University of Maryland, Baltimore County, piyawat@umbc.edu

Monica Adya
Marquette University, monica.adya@marquette.edu

Anthony F. Norcio
University of Maryland, Baltimore County, norcio@umbc.edu

Published version. Published as part of the proceedings of the conference, *Americas Conference on Information Systems 2000*, 2000: 330-336. Publisher link. © Association for Information Systems 2000. Used with permission.
Monica Adya was affiliated with the University of Maryland - Baltimore County at the time of publication.

The Distributed User Modeling Shell System (DUMSS): A Conceptual Framework for Eliciting User Models

Thawatchai Piyawat, Monica Adya, and Anthony F. Norcio
Department of Information Systems
University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
{piyawat, adya, norcio}@umbc.edu

Abstract

With the advances in communication technology, large volumes of information can transfer across continents within a fraction of a second. Nevertheless, computer users still suffer from unpleasant situations when they interact with systems and are required to adapt to systems rather than the other way round. User modeling aims to overcome this problem by enabling computer systems to interact with users according to the users' models, i.e., goals, knowledge, and preferences of users. Although, user modeling has shown invaluable benefits, methods of capturing user information to build precise and useful user models are still in their early states. This paper proposes a new approach for gathering user information by pooling the information from different systems. This concept, entitled Distributed User Modeling (DUM) is based on a method in which sensors built into each system contribute specific user information to the pooling. Having multiple sources of user information increases the possibility that a system can generate reliable user models. A general model of DUM is presented in this paper. The conceptual framework of the Distributed Fuzzy Object-Oriented User Modeling System (D_{FOO}UMS) that uses DUM as its basis structure is also presented.

Category: Technical

Keywords: User Modeling, User Adaptation, Fuzzy Logic, Distributed Systems, Object-oriented Implementation

Introduction

> ISSAQUAH, Wash. (AP) – A 43-year-old man was coaxed out of his home by police after he pulled a gun on his personal computer and shot it several times, apparently in frustration. (Gershenfeld, 1999)

In this 1997 incident, the individual was reported to have shot the computer four times through the CPU and once through the monitor. He was then undertaken for mental evaluation.

"Frustration" may most aptly describe peoples' feeling while interacting with computers. Current systems pose large cognitive demands on users because users are expected to adapt themselves to the system rather than vice versa. One of the primary goals of research on user modeling is to make computing experiences pleasant and productive for the users. This is done by retrieving and maintaining the users' goals, knowledge, and preferences as parts of user-system interaction. This paper presents a metaphor for enabling different computer systems to cooperate and use their user models to supply their users with better computing experiences.

Definition of User Modeling and User Models

Orwant humorously defines User modeling as "… nothing more than a fancy term for automated personalization" (Orwant, 1996). From this perspective, a system adapts itself according user preferences to provide users the comfort of operating the system. However, personalization is not the only purpose of user modeling. It involves eliciting and modeling the users' goals, knowledge, and preferences as parts of user-system interaction. Consequently, these elements must be recognized as important elements of user modeling.

In order to accomplish the objectives of user modeling, a system must maintain information about its user. These sets of information are called user models which represent all the necessary information and assumptions about particular users that are required in the modeling processes (Kobsa, 1994). While all aspects of a user's interaction with the system are valuable for user modeling, the use of the same piece of information may vary from one system to another as also the effectiveness of the usage may differ.

History and Current Status of the Research and Development

Research in user modeling may date back to Rich's proposal of stereotype-based user modeling (Rich, 1979). Rich presents *Grundy*, a librarian system that recommends books to users based on their personality. In a stereotype, it has a characteristic element, called *trigger*, that is used to choose the stereotype from a hierarchy of

stereotypes. Rich's original methods of obtaining user information and selecting a stereotype are relatively simple. The methods were later enhanced by many researchers. For instance, Chiu, *et al* (Chiu, et al., 1999) and Farinholt and Norcio (Farinholt and Norcio, 1996) propose the use of fuzzy logic to acquire user knowledge. Chen and Norcio (Chen and Norcio, 1991; Chen and Norcio, 1992; Chen and Norcio, 1997), Chiu, *et al*. (Chiu and Norcio, 1991), and Jennings and Higuchi (Jennings and Higuchi, 1993) present the application of neural network methodology to model users. In her later work, Rich admitted to immeasurable individual differences and factored this in her work (Rich, 1989; Rich, 1999).

Since Rich's *Grundy* system, a large number of systems equipped with user-modeling capabilities were introduced in the literature. For example, *um Toolkit* (Kay, 1995) allows users to understand their user models; *BGP-MS* (Kobsa and Pohl, 1995) and *Doppelganger* (Orwant, 1996) discuss the concept of user modeling servers. In additions to presenting user-modeling systems, several articles have focused on particular aspects of user modeling, such as logical representation of user models (Pohl, 1999) and a methodology for dynamically tracking user expertise (Chiu, et al., 1995).

Indeed, a significant amount of literature in user modeling emphasize natural language dialog systems. By including the requirement of having natural language processing capacity, it makes user modeling more challenging and distracts its from its original purpose by aligning it more with artificial intelligence.

User Modeling Techniques

Although several user-modeling techniques are proposed in the literature, they can be grouped into two categories: interaction-based modeling techniques are useful for extracting one user fact at a time while stereotype-based modeling techniques are useful in inferring a cluster of facts about the user at a single point in time. Orwant's *Doppelganger* system (Orwant, 1996) may be considered an interaction-based modeling system whereas Rich's *Grundy* system (Rich, 1979) is a stereotype-based modeling system.

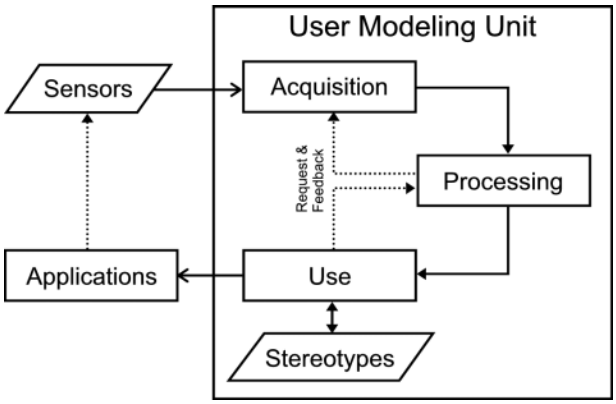
Interaction-based User Modeling: User modeling techniques that fall into this category are primarily based on information retrieved from interactions between users and systems. The information may be short-term, i.e., gathering at the current interactions, or long-term, i.e., stored from previous interactions. These techniques are not suited for systems that are not frequently used since the systems may not be able to create user models on time from actual interactions. Additionally, user-modeling procedures using these techniques are operationally expensive comparing with the stereotype-based methods.

Stereotype-based User Modeling: Interaction-based user modeling has a limitation when the user model is complex. Collecting all the information required to form a user model within a reasonable time is challenging. Nevertheless, human characteristics and behaviors are not completely different. Instead, they can be grouped into clusters. One reason is the cause and effect relationships. For example, persons who enjoy reading books are likely to have higher tolerant for reading detailed information. The reoccurrences of traits can be grouped into stereotypes which are collections of human characteristics and behaviors that normally occur all together. User behavior clusters overlap and are not mutually exclusive. With this feature, a hierarchy of stereotypes can be formed. The hierarchy structure begins with very general stereotype, which has very few attributes and their values. By adding facets, stereotypes in the structure become gradually complex and specific.

The Process of User Modeling

User modeling contains three iterative and concurrent processes that are occurred repeatedly throughout the lifespan of a user modeling system. These processes is depicted in following illustration:

Figure 1. Model of General User Modeling Unit



User Model Acquisition

In user modeling, the availability of data of users is not the concern (Orwant, 1996). Users continuously reveal their information through every interaction with systems. Therefore, the challenge of user modeling is how to obtain and use the information effectively.

Generally, a user-modeling unit gathers information from users via sensors (Orwant, 1996). The terminology "sensor" used here is broadly defined as everything used to retrieve user information and send it to user-modeling unit. They may be hardware, e.g., video camera, microphone, or software such as a program that monitors user usage behavior. A sensor may even be the application itself. Each kind of sensors gives a different type of user information to the system. Besides, each individual sensor may have different capability of collecting information.

Kobsa (Kobsa, 1990) categorizes user model acquisition into four components: eliciting default assumptions, eliciting an initial individual user model, eliciting assumptions based on user's input, and eliciting assumptions from dialog contributions made by the system. Orwant (Orwant, 1996) simply divides user model acquisition to *active* and *passive* methods. The active method asks users their preferences directly whereas the passive method collectively investigates user interaction with system. Active method can gain more user information than the passive one. Besides, information acquired using active method tends to be more accurate than does another. However, the active method may become annoyance to its user. Indeed, users should perceive user modeling in the same way they recognize soundtracks of movie in such that they only notice of its existence when it is missing (Orwant, 1996). Additionally, evidences presented in the psychological literature indicate users are not reliable source of their own information (Rich, 1999). Based on these reasons, passive method is preferred.

User Model Processing

Being able to retrieve accurate and reliable information about users from a large number of sources is considered as a challenge to researchers. Each piece of information acquired from users is unstructured and seemingly unrelated to others. Hence, the user-modeling processing task is to figure out the structures and relations of all pieces of information then compile them to be useful knowledge of users. Thus far, statistics and artificial intelligence techniques have been proposed for user modeling because they are capable to handle data that contains ambiguity and uncertainty as expected from the user modeling acquisition process (Orwant, 1996).

User Model Usage

Considered as the easiest process of user modeling, this module provides user models to applications as requested. This module has two sources of information: user model processing module and stereotypes. It is dependent on user modeling technique in which source is principally used. For stereotype-based user modeling, information from user model processing module may use to select a stereotype. Then, the user model is primarily derived from the stereotype. On the other hand, for interaction-based user modeling, results from user model processing module is the main content of user models and the stereotype is only the fulfillment of missing information.

The Potential of User Modeling

User modeling can be implemented in many types of systems. User modeling may be also used for purpose of control. For instance, in a critical system that requires precise procedures, modeling the person who operates it may help to indicate a problem before it really occurs. Additionally, for performance measurement, by modeling its users, a system can show the users' performance of operating it. The performance value can be used in various tasks, e.g., system improvement or organizational management.

Concerns of User Modeling

Users must be informed that the system they are using gathers information about them. More importantly, they must have control over the contents and the use of their user models before the models are employed. User-modeling systems must provide users the opportunity to disable the user-modeling component. These issues are critical for user acceptance of their models.

Short-term and long-term user models may be used inappropriately. A user's model may be a complex representation of both long-term facts such the ability of users to understand complex mathematical formulae, and short-term facts, such as user responses to a recent dialogue box. The misuse of short-term user information is less serious than the abuse of long-term information which gets outdated fairly quickly.

Issues with reliability of user models focuses on the misunderstandings that occur during user modeling processes. Mistaken assumptions arise regardless of the sophistication of the user modeling methods. Eliciting incorrect user models affects systems usability. When systems frequently use incorrect user models, users gradually develop their distrust of the systems.

The last concern involves legal restrictions. User modeling is the process of dealing with personal information. As nature of data in this type, legal restrictions must be considered. Each country has different restrictions of using personal data. Those law and ethical issues must be well considered.

Intelligent Agents in User Modeling

Many recent papers regarding to agent computing as appeared in AGENT'97, AGENT'98, and AGENT'99 such as (Barrett, et al., 1997; Chen and Sycara, 1998; Elliott, 1997; Ghosh, et al., 1999; Grand, et al., 1997; Hirsh and Davison, 1997; Segal and Kephart, 1999) discuss the agent-user interaction and the methodology agents use to gather information of users. They imply the strong requirement of user modeling in agent computing research society.

User modeling is considered as the most important part of IUI and clearly, both domains have to gain significantly from each other.

User Modeling Shell Systems and User Modeling Servers

Developing a system that has user-modeling capability is extremely expensive and time consuming.

Consequently, user modeling shell systems have been proposed. A shell system is a component that can be included in application systems to supply user-modeling functionality. Software developers can simply interface their systems to the user modeling shell component via pre-defined functions. Comparing with developing a user-modeling module from scratch, this concept dramatically reduces the amount of efforts needed for building modeling equipped systems.

GUMS (Finin, 1989), one of early user modeling shell system, is a stereotype-based system. Knowledge that researchers gain from *GUMS* is expanded in the later shell systems. *um Toolkit* (Kay, 1995) allows users to review and edit their user models. Aimed to improve both quality and understandability of the user models, it relies co-operation between users and system to build user models. *UMT* (Kobsa, 1993), a stereotype-based user modeling shell system, highlights mechanisms of cooperation between user modeling shell systems and applications. *BGP-MS* (Kobsa and Pohl, 1995) allows having more than one type of user assumptions, namely beliefs, goals, and plans. The development of *BGP-MS* puts some innovative concepts to the research in user modeling.

Another concept to facilitate user-modeling process is to adapt client-server computing paradigm. Applications act as clients requesting user models from user modeling servers. The servers process all user-modeling tasks based on user information provided by applications. There are a few systems implementing this concept. With the current advancement of computer network technology, user-modeling servers may provide better user modeling quality and may be more flexible than user modeling shell systems. *Doppelganger* (Orwant, 1996) operates alone as a user-modeling server instead of being a part of applications as shell systems. *Doppelganger* can serve many applications simultaneously. It initiates a new working structure of user modeling and proposes the use of statistics.

User Modeling and Future Computing

User Modeling and Intelligent Agents

Recently, researchers are veering into the area of agent computing and the paradigm starts to shift from getting users to manipulate computer systems to using software agents for completing tasks on behalf of users (Shneiderman and Maes, 1997). The idea of agent-based computing is initiated by Alan Kay (Kay, 1990). Kay proposes *indirect management* as an opposition of *direct manipulation*, which is coined by Ben Shneiderman (Shneiderman, 1983). In a direct manipulation interface system, every action occurred is activated by its user. On the other hand, in indirect management metaphor, tasks, goals, and interests of users will be accomplished by agents (Petrie and Wiggins, 1997).

By definition, an agent means software that can perform tasks on behalf of users or guide users to complete the tasks (Petrie and Wiggins, 1997). An agent is indeed a program that is proactive, personalized, and adapted (Beale and Wood, 1994). An agent knows user interests, habits, and goals. By using that information, an agent actively helps its user by giving suggestions and completing tasks that it has predicted they will be useful of supporting the user (Petrie and Wiggins, 1997). An agent may communicate with other agents in order to achieve its goal (Beale and Wood, 1994).

The main distinction between an agent and routine computer programs is the sense of itself as an independent entity (Maes, 1997). Specifically, an agent differs from typical software by following properties (Shneiderman and Maes, 1997). (1) An agent knows each user's habits, preferences, and interests. (2) An agent is proactive. It can initiate tasks according to user's preferences. (3) An agent's running time is longer. It also runs autonomously. (4) An agent is adaptive by tracking the change of user's information over time.

Consequently, an agent is a piece of software that gathers each user's preference and adapts itself according to the retrieved information. Based on user's preference, it runs autonomously to accomplish user's tasks and give guidance to the user. An agent may communicate with other agents to collaborate and share information.

In conjunction with the developments in agent computing, researchers are beginning to examine interaction between agents and users. Agents of this type are called user interface agents or Intelligent User Interface (IUI). Maybury (Maybury, 1999) indicates that "…Intelligent User Interfaces (IUI) are human-machine interfaces that aim to improve the efficiency, effectiveness, and naturalness of human-machine interaction by representing, reasoning, and acting on models of the user, domain, task, discourse, and media."

Obviously, agents need user models in order to achieve its goals. As stated previously, an agent must know its user's habits, preferences, interests, and goals. Using that knowledge, it accomplishes tasks on behalf of the user or gives suggestions to the user to achieve the tasks with ease and less effort. Hence, the primary information an agent must obtain before it can perform any of its tasks is user models of its users. Furthermore, it must maintain the user models to be accurate and identical to its users as much as possible at all time. Having more precise user models means it can serve its users better. In conclusion, with reasons as shown, user modeling is a critical element of agent computing.

User Modeling and Ubiquitous Computing

Ubiquitous computing is a framework to enhance computer use by making several computers be invisibly available to users via physical environment (Weiser,

1993). This computing concept relies on the fact that "a good tool is an invisible tool" (Weiser, 1994). According to this computing model, users use computers intuitively without notice of its existence. This metaphor refers to the way users use alphabets. Most people do not perceive alphabets as communication tools when they use them to communicate as alphabets are broadly embedded in almost everything in the world. Consequently, ubiquitous computers are (1) invisible to their users and (2) available to use via physical environment when needed.

According to ubiquitous properties explained, ubiquitous computers must be invisibly available to their users when they need. Apparently, ubiquitous computers must know their users requirements and maintain them to be updated at all time so that they can serve their users as they are expected. Therefore, ubiquitous computers must have user-modeling module. The module constantly monitors their users. Once it finds the assistance from computers is appropriated, it activates the computers to serve the users.

The Requirement of the Future Computing

Studies on Intelligent Agents and Ubiquitous computing aim to support users in completing their tasks as easily and comfortably as possible. To do so, they need to model their users continuously using all available sensors they have. However, each system certainly has limited sensors. A simple but effective way to overcome this limitation is to pool their user models together. By cooperating among them in exchanging user models, their capability of acquiring user models will undoubtedly improved. The requirement of exchanging user models among systems leads to the project D_{FOO}UMS discussed latter in this paper.

Project D_{FOO}UMS: The Big Picture

The project D_{FOO}UMS (Distributed Fuzzy Object-Oriented User Modeling System – pronounced diffuse-U-M-S) is an attempt to utilize the advantages of object-oriented methodology, fuzzy logic, and distributed computing environment to user modeling activity. D_{FOO}UMS system consists of four conceptual components as follows:

Object-Oriented User Models (OOUM) is the framework to facilitate the implementation of user models to be more effective, intuitive, and effortless by applying object-oriented methodology to the construction and the application of user models.

Fuzzy Object-Oriented User Modeling (FOOUM) is the framework of applying fuzzy logic to OOUM. In the mechanism of OOUM, a class of a user object, which is an instance of an individual user, will be created as needed. Properties of the class will be determined on the fly. In addition, during the interaction with the user, a property may add or drop from the class according to information that the system gathers from the user. Fuzzy logic is employed here to select properties of a class based on uncertainty and incompleteness of user information. Conclusively, fuzzy logic is the mechanism that drives the decision of D_{FOO}UMS.

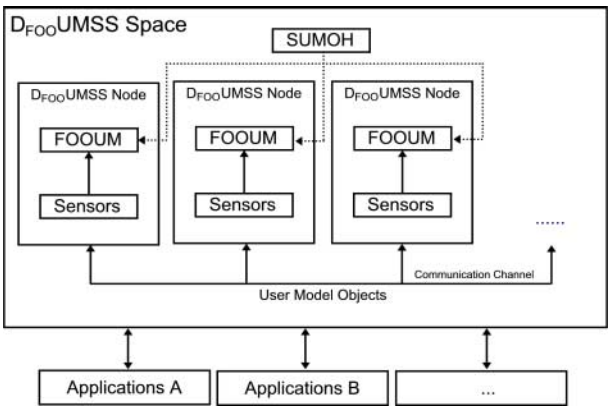
Standard User Model Object Hierarchy (SUMOH) is a hierarchy of meta-class for specifying property classes in D_{FOO}UMS system. Property classes contain user information and its operator. As stated previously, D_{FOO}UMS system is aimed to be a distributed user modeling shell system. In order to achieve this goal, a standard method of defining objects in D_{FOO}UMS must be defined. Then, objects created according to the standard are able to distribute across networks of systems running D_{FOO}UMS.

Distributed User Modeling Shell System (DUMSS) is a user modeling shell system designed for distributed computing environment. The user models from DUMSS come from collaborations of many DUMSS nodes. The assumption of DUMSS is *more DUMSS nodes participating in the collaboration, more accurate the user models can be.*

Each DUMSS node has its sensors to gather information from its user. When it requires information in which it cannot retrieve by itself, it requests the information from other DUMSS node that has it. For information that it has its own sensor to gather, it may also ask the same information from other nodes for validation with its current information. Additionally, different node of DUMSS may have different mechanism to analyze information. By pooling results together, the accurate rate of the user model is certainly increased. User models will be transferred among DUMSS nodes in the form of objects created from a class that is in turn defined based on SUMOH.

The relationship of these components can be depicted as in Figure 2.

Figure 2. Conceptual architecture of D_{FOO}UMS



As shown in Figure 2, applications virtually interact with D_{FOO}UMSS space instead of particular node as architectures of other user-modeling servers do. Each D_{FOO}UMSS has its own sensors. The sensors may be similar or may differ from sensors of other nodes. All nodes are connected together with a communication channel. The channel is broadly defined here as any kind of connections they may have. Communication among nodes is done in the form of user model objects, which is created according to SUMOH.

References

Barrett, R., Maglio, P.P. and Kellem, D.C. "WBI a confederation of agents that personalize the Web," *Proceedings of the first international conference on Autonomous agents*, 1997, pp. 496-499.

Beale, R. and Wood, A. "Agent-based Interaction," *Proceedings of the People and computers IX: HCI's 94*, 1994, pp. 239-245.

Chen, L. and Sycara, K. "WebMate: a personal agent for browsing and searching," *Proceedings of the second international conference on Autonomous agents*, 1998, pp. 132-139.

Chen, Q. and Norcio, A.F. "A neural network approach for user modeling," *Proceedings of the IEEE 1991 International Conference on Systems, Man, and Cybernetics*, 1991, pp. 1429-1434.

Chen, Q. and Norcio, A.F. "Modeling users with neural architectures," *Proceedings of the International Joint Conference on Neural Networks*, 1992, pp. 547-552.

Chen, Q. and Norcio, A.F. "Modeling a user's domain knowledge with neural networks," *International Journal of Human-Computer Interaction* (9:1), 1997, pp. 25-40.

Chiu, C., Hsu, C.-I. and Norcio, A.F. "Dynamically tracking a user's progression along novice-expert continuum," *Proceedings of the 1995 IEEE Singapore International Conference on Networks and International Conference on Information Engineering*, 1995, pp. 605-607.

Chiu, C., Hsu, P.-L. and Norcio, A.F. "A fuzzy reasoning approach to support knowledge acquisition," *International Journal of General Systems* (28:2-3), 1999, pp. 227-241.

Chiu, C. and Norcio, A.F. "Using neural networks and expert systems to model users in an object-oriented environment," *Proceedings of the IEEE 1991 International Conference on Systems, Man, and Cybernetics*, 1991, pp. 1943-1948.

Elliott, C. "I picked up Catapia and other stories a multimodal approach to expressivity for "emotionally intelligent" agents," *Proceedings of the first international conference on Autonomous agents*, 1997, pp. 451-457.

Farinholt, K.G. and Norcio, A.F. "Defining fuzzy user models using the GoM methodology," *Proceedings of the 9th Florida Artificial Intelligence Research Symposium*, 1996.

Finin, T.W. "GUMS -- A general user modeling shell," In *User Models in Dialog Systems*, A. Kobsa and W. Wahlster (Ed.), Springer, Heidelberg, 1989, pp. 411-430.

Gershenfeld, N. *When Things Start to Think*, Henry Holt and Company, New York, 1999.

Ghosh, S., Mundhe, M., Hernandez, K. and Sen, S. "Voting for movies: the anatomy of a recommender system," *Proceedings of the third annual conference on on Autonomous Agents*, 1999, pp. 434-435.

Grand, S., Cliff, D. and Malhotra, A. "Creatures artificial life autonomous software agents for home entertainment," *Proceedings of the first international conference on Autonomous agents*, 1997, pp. 22-29.

Hirsh, H. and Davison, B.D. "An adaptive UNIX command-line assistant," *Proceedings of the first international conference on Autonomous agents*, 1997, pp. 542-543.

Jennings, A. and Higuchi, H. "A user model neural network for a personal news service," *User Modeling and User-Adapted Interaction* (3), 1993, pp. 1-25.

Kay, A. "User Interface: A Personal View," In *The Art of Human Computer Interface Design*, B. Laurel (Ed.), Addison-Wesley, CA, 1990, pp. 191-207.

Kay, J. "The um Toolkit for Cooperative User Modeling," *User Modeling and User-Adapted Interaction* (4), 1995, pp. 149-196.

Kobsa, A. "User Modeling in Dialog Systems: Potentials and Hazards," In *AI & Society*, 4, Springer-Verlag, London, 1990, pp. 214-240.

Kobsa, A. "User modeling: recent work, prospects and hazards," In *Adaptive user interfaces: principle and practice*, M. Schneider-Hufschmidt, T. Kuhme and U. Malinowski (Ed.), North-Holland, Amsterdam, 1993.

Kobsa, A. "User Modeling and User-Adapted Interaction," *Proceedings of the CHI '94 conference companion on Human factors in computing systems*, 1994, pp. 415-416.

Kobsa, A. and Pohl, W. "The user modeling shell system BGP-MS," *User Modeling and User-Adapted Interaction* (4:2), 1995, pp. 59-106.

Maes, P. "Intelligent Software," *Proceedings of the 1997 international conference on Intelligent user interfaces*, 1997, pp. 41-43.

Maybury, M. "Intelligent user interfaces: an introduction," *Proceedings of the 1999 international conference on Intelligent user interfaces*, 1999, pp. 3-4.

Orwant, J. "For want of a bit the user was lost: Cheap user modeling," *IBM Systems Journal* (35:3&4), 1996, pp. 398-416.

Petrie, C. and Wiggins, M. "PATTIE MAES on Software Agents: Humanizing the Global Computer," *IEEE Internet Computing* (1:4), 1997.

Pohl, W. "Logic-based representation and reasoning for user modeling shell systems," *User Modeling and User-adapted Interaction* (9), 1999, pp. 217-282.

Rich, E. "User Modeling via Stereotypes," *Cognitive Science* (3), 1979, pp. 329-354.

Rich, E. "Stereotypes and user modeling," In *User Models in Dialog Systems*, A. Kobsa and W. Wahlster (Ed.), Springer, Heidelberg, 1989, pp. 35-51.

Rich, E. "Users are individuals: individualizing user models," *International Journal of Human-Computer Studies* (51), 1999, pp. 323-338.

Segal, R.B. and Kephart, J.O. "MailCat: an intelligent assistant for organizing e-mail," *Proceedings of the third annual conference on Autonomous Agents*, 1999, pp. 276-282.

Shneiderman, B. "Direct Manipulation: a step beyond programming languages," *IEEE Computer* (16:8), 1983, pp. 57-69.

Shneiderman, B. and Maes, P. "Direct manipulation vs. interface agents," *Interaction* (4:6), 1997, pp. 42-61.

Weiser, M. "Some computer science issues in ubiquitous computing," *Communication of ACM* (36:7), 1993, pp. 75-84.

Weiser, M. "The world is not a desktop," *Interaction* (1:1), 1994, pp. 7-8.