# User Modeling: The Long and Winding Road

Gerhard Fischer

Center for LifeLong Learning and Design ($L^3D$), Department of Computer Science
and Institute of Cognitive Science
University of Colorado, Boulder, CO, USA

**Abstract.** The long and winding road of user modeling is grounded in different epistemological assumptions exploring different dimensions of the problem. User-modeling research has explored different domains, identified important distinctions underlying different approaches within user modeling research, and created a number of challenging research problems. These issues are explored in the context of high-functionality applications and how our research over the last ten years has addressed the problems of making high-functionality applications more usable, more useful, and more learnable with a variety of different user modeling approaches.

## 1. Introduction

User modeling is one of a number of research areas that intuitively seem to be winning propositions and worthwhile investments based on their obvious need and potential payoff. One area comparable to user modeling is software reuse. The approaches seem to be appealing, natural, theoretically justifiable, desirable, and needed (e.g., reuse can be justified by the fact that complex systems develop faster if they can build on stable subsystems). But in reality, progress in these areas has been slow and difficult, and success stories are rare.

The research area has

– explored different domains such as: natural language dialog, human computer interaction, intelligent assistants, information retrieval, and high-functionality applications;

– identified important distinctions such as: adaptive versus adaptable components, explicit versus implicit modeling techniques, user models versus task models, canonical versus individual models, and long-term versus short-term models;

– created a number of challenging research problems, such as how to: (1) integrate different modeling techniques; (2) capture the larger (often unarticulated) context and what users are doing (especially beyond the direct interaction with the computer system); (3) identify user goals from low-level interactions; (4) reduce information overload by making information relevant to the task at hand; (5) support differential descriptions by relating new information to known information and concepts; and (6) reach a better balance for task distributions between systems and users.

Some of these challenges will be illustrated in the context of the work that we have done to make high-functionality applications more usable, more useful, and more learnable.

## 2. High-Functionality Applications

High-functionality applications (HFAs) (such as Unix, MS-Office, Photoshop, Eudora, etc.) are used to model parts of the world and not just to implement algorithms. They are complex systems because "reality is not user friendly". If you ask a 100 different people what features they would like to have in a particular application, you end up with a very large number of features. The design of HFAs must address two problems: (1) the unused functionality must not get in the way and (2) unknown existing functionality must be accessible or delivered at times when it is needed.

We have conducted a variety of empirical studies to determine the usage patterns of HFAs, their structure, their associated help and learning mechanisms. All of these studies have led us to the identification of the qualitative relationships between usage patterns of HFAs as illustrated in **Figure 1**.

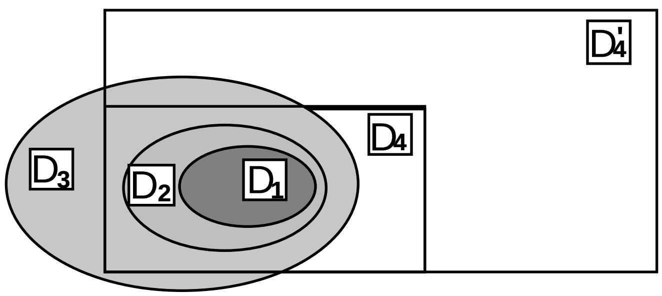


**Figure 1:** Levels of Users' Knowledge about a System's Information Spaces

The ovals represent users' knowledge about the system's information space. $D_1$ represents concepts well known, easily employed, and used regularly by a user. $D_2$ contains concepts known vaguely and used only occasionally, often requiring passive help systems. $D_3$ represents concepts users *believe* to exist in the system, some of which lie outside the actual information space. The rectangle $D_4$ represents the actual information space of a system. As the functionality of HFAs increases to $D_{4'}$, little is gained for users unless there are mechanisms to help them relate the additional functionality to their needs.

The area of $D_4$ that is not part of $D_3$ is of specific interest to research in user modeling. This is system functionality, whose existence is unknown to users. For the "$D_4$ and not $D_3$" domain, information access (the user-initiated location of information when they perceive a need for an operation) is not sufficient, but information delivery (the system volunteering information that it inferred to be relevant to the users' task at hand) is required. Active help systems and critics are required to point out to users functionality that may be useful for their tasks and to help users to avoid to get stuck on suboptimal plateaus.

**Figure 1** shows usage patterns of HFAs without taking the users' tasks into account. There is no reason for users to worry about additional existing functionality in $D_4$, if this functionality is not relevant to their tasks. However, if the system *does* provide functionality in $D_4$ related to users' tasks, it is desirable to avoid having users unable to perform the task or do so in a suboptimal or error-prone way because they do not know about this functionality. In **Figure 2** the gray rectangle T represents the information that is relevant to the users' task at hand, and the dots represent different pieces of functionality. *Passive* intelligent support systems supporting information access can help users to explore pieces of functionality that are contained in $D_3$ and T, whereas *active* intelligent systems supporting information delivery are

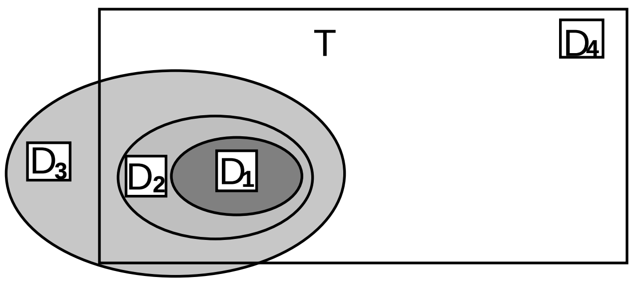


**Figure 2:** Functionality and its Relevancy to the Task at Hand

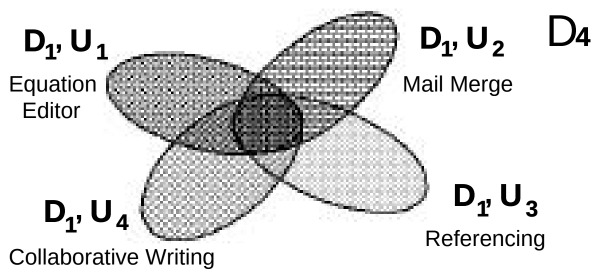


**Figure 3**: Distributed Expertise in HFAs

needed for the functionality contained in T and not in $D_3$. The functionality in $D_4$ outside of T is often offered by push systems such as "Did You Know" (DYK) systems [Owen, 1986] or Microsoft's "Tip of the Day" [Horvitz, 1997], which throw decontextualized concepts at users.

***"Experts" and Expertise in HFA.*** "Experts" (users who know everything about a system) no longer exist in HFAs. Being an "expert" is at best an attribute of a specific context, rather than a personal attribute. The different spaces of expertise (determined by individual interest) are illustrated in **Figure 3**. In this *multi-kernel model*, {$D_1$, $U_i$} means the area of functionality that is well known to a particular user $U_i$; for example: $U_1$ knows about the equation editor; $U_2$ about mail-merge functionality; $U_3$ uses a bibliography system for references, and $U_4$ knows about collaborative writing tools.

## 3. How Our Research Address the Problems Created by HFAs

Our research related to user modeling has attempted to address the challenges created by HFAs. *Active help systems* [Fischer et al., 1985] were an early attempt to analyze the behavior of users and infer higher-level goals from low-level operations [Horvitz, 1997; Nardi et al., 1998].

HFAs, as argued above, came into existence as environments that are useful for a large number of different users (see **Figure 3**). In order to reduce their complexity, HFAs have often migrated to a collection of domain-oriented subsystems each with their own templates, forms, and their own associated wizards thereby being able to provide additional support for user modeling and assistance not available in more general systems.

In our own research, we have taken this approach further by developing *domain-oriented design environments* [Fischer et al., 1998]. These are environments which model specific domains (such as computer networks, user interfaces, kitchens, and voice dialog design [Sumner et al., 1997]) by allowing designers to engage in authentic tasks from their own respective work practices. Domain-oriented design environments make computers invisible and enable designers to communicate with domain-specific concepts, representations and tools. The domain orientation of these environments makes HFAs more usable, more useful, and more learnable by bringing the objects closer to the conceptual world of their users. In support of user modeling, domain-oriented design environments provide

- *specification components* [Nakakoji, 1993] to allow users to enrich the description of their tasks, and
- *critiquing components* [Fischer et al., 1998] to analyze and infer the task at hand in order to be able to detect and identify the potential for a design information need and then present stored knowledge for designers. We have used the artifact (including its partial construction and partial specification) combined with domain knowledge in the design environment as an index to infer high-level goals from simple user actions. The existence of the specification component has allowed us to complement generic critics (which can be defined at design time) with specific critics whose behavior is dependent on the information provided by individual users at use time.

Domain-oriented design environments integrate a number of components, and the research built around them has tried to developed principles and arguments for the trade-off between adaptive and adaptable approaches [Fischer, 1993; Thomas, 1996]. **Figure 4** shows a screen image from the Voice Dialog Design Environment [Sumner et al., 1997], which supports adaptation mechanisms to allow users to select specific rule sets for critiquing and to determine the intervention strategy for the intrusiveness of the critics ranging from active behavior (every step is immediately critiqued) through intermediate levels to passive behavior (users have to explicitly invoke the critiquing system). Similar solutions can be found in modern spelling correction programs. The need to customize and tailor critiquing systems to individual users' objectives has also been explored in the domain of business graphs and use of color [Gutkauf, 1998].

Domain-oriented design environments pose a number of challenging problems for user modeling. Contrary to Intelligent Tutoring Systems [Burton & Brown, 1982], they model (open-ended) domains in support of self-directed learning and user-directed activities within a domain. They are able to exploit domain models for user modeling [Hollan, 1990; Mastaglio, 1990]. At design time (when the system is developed), domain models including generic critiquing knowledge and support for specification, and end-user adaptation is provided. In these open-ended systems, the task that users engage in cannot be anticipated: they have to be inferred or articulated at use time. Design environments support both adaptable and adaptive components and tune system behavior to the specific needs of individual users and their tasks [Gutkauf, 1998; Nakakoji, 1993].

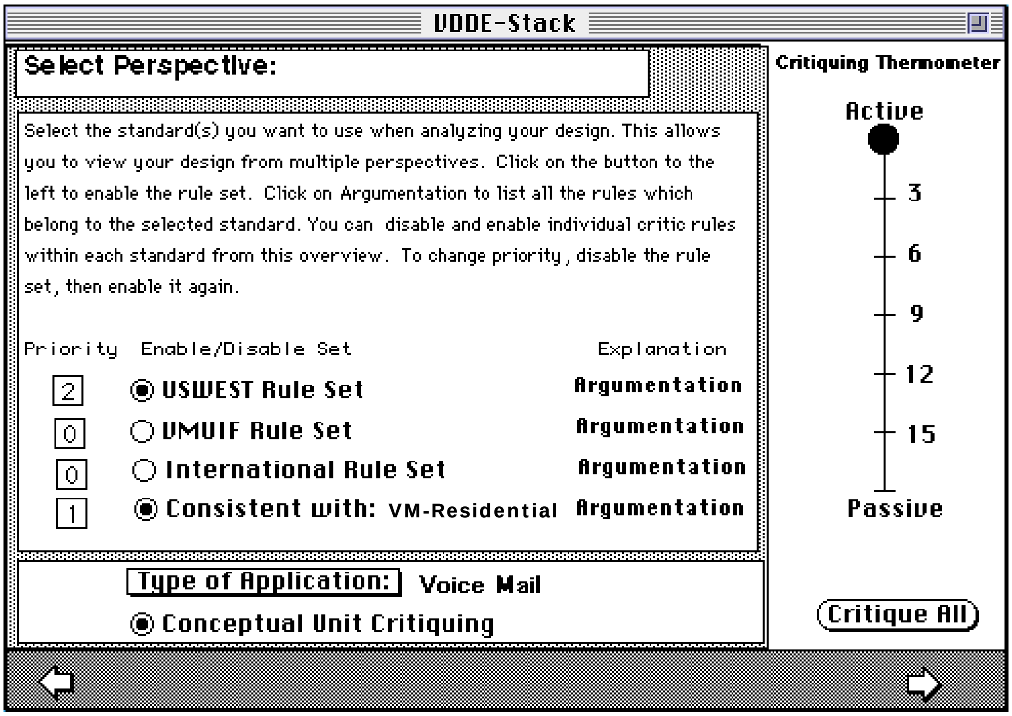


**Figure 4:** Adaptation Mechanism to Control Different Critiquing Rule Sets and Different Intervention Strategies

HFAs create challenging *learning problems* that are representative for numerous complex systems. As illustrated with the above figures, nobody learns these systems completely, but users acquire some base functionality and learn additional functionality on demand. User-modeling techniques can effectively support learning on demand [Fischer, 1991] by helping users to identify opportunities to learn additional functionality relevant to their task at hand and to avoid people becoming stuck on suboptimal plateaus. User modeling techniques based on logged user data can support the organization-wide learning of HFAs [Linton et al., 1998].

## 4. User Modeling and Human-Computer Collaboration

Some of the beginnings of the long and winding road of user modeling were derived from the need and desire to provide better support for human-computer collaboration. Collaboration in this context is defined as *"a process in which two or more agents work together to achieve shared goals"* [Terveen, 1995]. Some fundamental issues (such as shared goals, shared context, control, (co)-adaptation, (co)-evolution, and learning) can be derived from this definition. Human-computer collaboration can be approached from two different perspectives: an emulation and a complementing approach. The emulation approach is based on the metaphor that to improve human-computer collaboration is to endow computers with "human-like abilities". The complementing approach is based on the fact that computers are not human and that human-centered design should exploit the asymmetry of human and computer by developing new interaction and collaboration possibilities [Suchman, 1987].

Historically, the major emphasis in user modeling has focused on the human emulation approach (see for example, [Kobsa & Wahlster, 1989], and the section "user and discourse modeling" in [Maybury & Wahlster, 1998]). However, based on the limited success of the emulating approach, the interest has shifted more and more to the complementing approach [Bobrow, 1991; Fischer, 1990]. There is growing evidence that the problems of user modeling in the complementing approach are more tractable, more feasible, and more desirable, as evidenced by their increasing influence in the design of commercial high-functionality applications [Horvitz, 1997]. As the complexity of commercially available HFAs grows, and as we see more computational assistants (such as agents, advisors, coaches, and critics) appear in widely available commercial applications, a detailed, state-of-the-art analysis and understanding of how people learn, work and collaborate with and around HFAs will provide us with new requirements for the design of user modeling components.

In the long and winding road that is not only behind us but also before us, our understanding of the trade-offs, the promises, and the pitfalls between adaptive and adaptable systems, between push (information delivery) and pull (information access) technologies, and between contextualized information representation and serendipity will hopefully lead us to new ideas and new insights in the design of future human-centered systems supported by adequate user modeling techniques.

**Acknowledgments.** I would like to thank the members of the Center for LifeLong Learning and Design ($L^3D$) who have contributed to the ideas and perspectives expressed in this paper. A special *thank you* goes to my former students who, in their Ph.D. theses, explored important issues in user modeling as they occur in high-functionality applications, domain-oriented design environments, and critiquing systems.

## References

Bobrow, D. G. (1991) "Dimensions of Interaction," *AI Magazine*, 12(3), pp. 64-80.

Burton, R. R. & Brown, J. S. (1982) "An Investigation of Computer Coaching for Informal Learning Activities." In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent Tutoring Systems*, Academic Press, London - New York, pp. 79-98.

Fischer, G. (1990) "Communication Requirements for Cooperative Problem Solving Systems," *Information Systems*, 15(1), pp. 21-36.

Fischer, G. (1991) "Supporting Learning on Demand with Design Environments." In L. Birnbaum (Ed.) *International Conference on the Learning Sciences (Evanston, IL)*, Association for the Advancement of Computing in Education, pp. 165-172.

Fischer, G. (1993) "Shared Knowledge in Cooperative Problem-Solving Systems - Integrating Adaptive and Adaptable Components." In M. Schneider-Hufschmidt, T. Kuehme, & U. Malinowski (Eds.), *Adaptive User Interfaces - Principles and Practice*, Elsevier Science Publishers, Amsterdam, pp. 49-68.

Fischer, G., Lemke, A. C., & Schwab, T. (1985) "Knowledge-Based Help Systems." In L. Borman & B. Curtis (Eds.), *Proceedings of CHI'85 Conference on Human Factors in Computing Systems*, ACM, New York, pp. 161-167.

Fischer, G., Nakakoji, K., Ostwald, J., Stahl, G., & Sumner, T. (1998) "Embedding Critics in Design Environments." In M. T. Maybury & W. Wahlster (Eds.), *Readings in Intelligent User Interfaces*, Morgan Kaufmann, San Francisco, pp. 537-561.

Gutkauf, B. (1998) *Improving Design & Communication of Business Graphs through User Adaptive Critiquing*, Ph.D. Dissertation, Mathematics and Computer Science, Universität-GH Paderborn, Paderborn, Germany.

Hollan, J. D. (1990) "User Models and User Interfaces: A Case for Domain Models, Task Models, and Tailorability." In *Proceedings of AAAI-90, Eighth National Conference on Artificial Intelligence*, AAAI Press/The MIT Press, Cambridge, MA, p. 1137.

Horvitz, E. (1997) "Agents with Beliefs: Reflections on Bayesian Methods for User Modeling." In C. P. A. Jameson, C. Tasso (Ed.) *Proceedings of the Sixth International Conference on User Modeling (UM'97)*, Springer Wien-New York, Sardegna, Italy, pp. 441-442.

Kobsa, A. & Wahlster, W. (Eds.) (1989) *User Models in Dialog Systems*, Springer-Verlag, New York.

Linton, F., Charron, A., & Joy, D. (1998) *OWL: A Recommender System for Organization-Wide Learning*, at http://www.mitre.org/technology/tech_tats/modeling/owl/Coaching_Software_Skills.pdf.

Mastaglio, T. (1990) *A User-Modelling Approach to Computer-Based Critiquing*, Ph.D. Dissertation, Department of Computer Science, University of Colorado at Boulder, Boulder, CO.

Maybury, M. T. & Wahlster, W. (1998) *Readings in Intelligent User Interfaces*, Morgan Kaufmann, San Francisco.

Nakakoji, K. (1993) *Increasing Shared Understanding of a Design Task Between Designers and Design Environments: The Role of a Specification Component*, Ph.D. Dissertation, Department of Computer Science, University of Colorado at Boulder, Boulder, CO.

Nardi, B. A., Miller, J. R., & Wright, D. J. (1998) "Collaborative, Programmable Intelligent Agents," *Communications of the ACM*, 41(3), pp. 96-104.

Owen, D. (1986) "Answers First, Then Questions." In D. A. Norman & S. W. Draper (Eds.), *User-Centered System Design, New Perspectives on Human-Computer Interaction*, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, pp. 361-375.

Suchman, L. A. (1987) *Plans and Situated Actions*, Cambridge University Press, Cambridge, UK.

Sumner, T., Bonnardel, N., & Kallak, B. H. (1997) "The Cognitive Ergonomics of Knowledge-Based Design Support Systems." In S. Pemberton (Ed.) *Proceedings of CHI 97 Conference on Human Factors in Computing Systems*, ACM/Addison-Wesley, pp. 83-90.

Terveen, L. G. (1995) "An Overview of Human-Computer Collaboration," *Knowledge-Based Systems Journal, Special Issue on Human-Computer Collaboration*, 8(2-3), pp. 67-81.

Thomas, C. G. (1996) *To Assist the User: On the Embedding of Adaptive and Agent-Based Mechanisms*, R. Oldenburg Verlag, München/Wien.