Usability Engineering Methods for Interactive Intelligent Systems

Aaron Spaulding
and Julie S. Weber

■ *The field of human-computer interaction (HCI) offers designers and developers of interactive systems a large repertoire of methods for ensuring that their systems will be both usable and useful. This article offers a brief introduction to these methods, focusing on the ways in which they may need to be adapted and extended to take into account the characteristic properties of systems that include some sort of AI. The discussion is organized around three types of activity: understanding users' needs, interaction design, and evaluation.*

Some may feel that the usability (and usefulness) of intelligent interactive systems is easy to ensure: just apply design and evaluation methods from the vast repertoire that has been built up by researchers and practitioners in the field of human computer interaction (HCI). The application of these methods is described with terms such as *usability engineering* and *user-centered design*. These activities constitute part of the overall software development process, and they must be carefully coordinated with the other activities involved in that process.

There is considerable validity to this point of view: Anyone who develops systems that are intended for use by *people* can benefit from familiarity with and application of these methods. Accordingly, this article offers a brief introduction to these methods, including examples and suggestions for additional reading (see in particular the Further Reading section).

Even people who are already experts in the application of these methods should be aware of potential adaptations and extensions to the methods when applied to systems that are designed to incorporate significant use of AI. The theme articles by Lieberman (2009) and by Jameson (2009) in this issue discuss some of the ways in which systems that incorporate intelligence tend to differ from systems that do not, both in terms of their potential to help users and in terms of possible side effects. These and other properties of intelligent systems can affect the application of design and evaluation methods in various ways, some of which are illustrated in the case studies of this special issue.

To organize our discussion, we distinguish broadly three types of activity that are involved in usability engineering: understanding users' needs, interaction design, and evaluation. Except for the fact that understanding users' needs tends to occur early in the design process, these activities generally proceed in parallel and complement each other.

 ISSN 0738-4602

Understanding User Needs

A key principle in the user-centered design of interactive systems is that the design should be based on a detailed understanding of the needs of users. Terms like *requirements analysis,* which are also used in connection with the analysis of non-user-related requirements, are often used to refer to the activity of gathering and analyzing information about users' needs.

Users' needs include (1) the goals that the users want to achieve with the system (for example, spend less time managing e-mail); (2) the specific tasks that they want to perform with it (for example, tag individual e-mail messages); (3) the contexts in which they want to perform these tasks (for example, in the office or on the go); (4) the existing work patterns that they may want to maintain (for example, tagging messages at the end of each day, or during bits of idle time); (5) the properties of their computing devices (for example, PC or smartphone); and (6) the criteria that they expect their interaction with the system to fulfill (for example, speed versus accuracy).

There are also less utilitarian but equally important needs, such as the desire to enjoy using the system or to act in a way that is socially and culturally appropriate (for example, not looking obviously engrossed in e-mail processing during a meeting).

Even merely contemplating potential users' needs in detail can be daunting to a designer who aims to enhance interaction through the introduction of intelligence: the intelligence may enable the system to perform certain tasks very well (for example, automatically suggest appropriate tags for e-mail messages), but whether the resulting system will adequately meet the total set of diverse user needs is quite a different question.

Collecting User Data

A large collection of methods for acquiring and analyzing data about user needs has evolved in the HCI field (see, for example, relevant chapters in the HCI texts listed in the Further Reading section). For instance, *contextual inquiry* (Beyer and Holtzblatt 1998) is designed to reveal hidden work structure and identify the possibilities for technology-based improvement through observation and questioning of a small number of representative users in the places where they work. Typically, this takes the form of a partnership, where the analyst takes the role of an "apprentice" who asks the potential user (as an "expert" on the current work practices) to show and explain how the work is currently done. If fielded or prototype systems exist, log analysis and use surveys can provide insight into use patterns. Diary studies (see, for example, Czerwinski, Horvitz, and Wilhite [2004]) yield records of user activities at set points throughout the day and are useful for gathering evidence about the nature and duration of users' activities. Questionnaires can ask about many different types of users' needs, though the data collected is not always as reliable and informative as the more direct methods just listed.

These methods have general applicability, but applying them may require special knowledge when intelligent technology is involved: the team members involved may need not only to have experience with HCI and behavioral research techniques but also at least a basic understanding of the capabilities of intelligent systems and the design issues that they raise. For example, if you are carefully observing a potential user in his or her workplace in order to identify possibilities and constraints for the introduction of a programming by demonstration system, you may not know what to look for and what questions to ask if you don't have enough understanding of such systems (see Lau 2009).

Checking the Understanding of Goals and Needs

An intelligent system may offer opportunities to dramatically change the goals that users can achieve with their systems and raise new questions about the nature of these goals. This is especially an issue in cases in which an intelligent system is capable of performing tasks that have previously been performed only by humans.

As an example, consider a manager who is required to prepare a monthly briefing for a superior. With normal technology, this task entails requesting status updates from several coworkers, consolidating the updates into a template-based PowerPoint presentation, and then making final, last-minute updates to the presentation on the basis of data that are released the day the report is due. There are several potential breakdowns in this workflow; for example, the coworkers could be absent or slow to respond, there could be transcription errors, or the last minute data could be late. A system could be built to address these breakdowns and thus simplify the manager's job of preparing the report. But a new, intelligent system could influence the manager's workflow much more dramatically, for example by continuously updating and presenting a live view of all the data, thus eliminating the need for the manager to create the monthly report in the first place.

But will this radically changed workflow be well received by users? Some managers may love the fully automated solution, while others may feel that assembling the report is a key value that they add to an organization. In this latter case, the managers may view the fully automated solution as a solution that diminishes their control and authority rather than one that provides assistance.

We see that insight into the acceptability of the automated solution requires an understanding of the structure of the users' work, the capabilities of the technology, and the underlying culture and context in which the system will be deployed—and just as importantly, the ways in which users perceive and feel about these factors.

In general, it is not possible to arrive at a fully validated understanding of users' needs by collecting and analyzing data in the ways sketched in the previous subsection. For example, while conducting a contextual inquiry, the observer/interviewer can ask questions like "So you only need to generate this report because the information from all these e-mails has to be consolidated in a single place?" This sort of probing can result in immediate confirmation of observations that occur during a session. But it is not meant to be a way to test design ideas, and delving into hypothetical scenarios during a contextual inquiry session would distract from the observation of users performing their tasks. So a different type of method is required for checking and refining an initial understanding of users' needs.

One recently proposed method for gaining such insight is the *need validation session* (Davidoff et al. 2007): a design team presents to potential users paper storyboards (similar to comic strips) that depict concrete scenarios in which people use the envisaged system. In our monthly-report example, a storyboard might show a manager discussing with her superior a snapshot of the continually generated information update, instead of presenting her usual monthly report. In a need validation session, some scenarios are intentionally chosen to go beyond what the designers currently expect to be acceptable to users, the goal being to get a clearer idea of users' needs on the basis of a mix of positive and negative reactions. In some cases, for example, user needs that had initially been identified as significant by researchers will be judged to be trivial by the intended users. In these cases, a new way to meet those needs is likely to be seen by the users as offering little value. Through a need validation session, researchers can confirm which needs are really felt and acknowledged by the intended users, and they can propose solutions accordingly. In addition, the need validation session may help the users better understand the capabilities of the technology. This improved understanding may lead to new suggestions about how the technology and operations could be improved or streamlined with the new system.

A related use of scenarios and storyboards is described by Petrelli, Dadzie, and Lanfranchi (2009), in this issue. As this case study shows, the very process of creating a storyboard can act as a means of ensuring that both the technology-oriented and the user-oriented members of the design team acquire adequate understanding of the technologies and the intended users.

Interaction Design for Intelligent Systems

Once an adequate initial understanding of users' needs has been achieved, a major activity of usability engineering is *interaction design:* specifying how the interaction between the users and the system will look, first on a high level and then with increasing attention to specific interface elements and user actions. Using the term *interaction design* instead of *interface design* does justice to the fact that it is not just a matter of designing attractive screens and buttons—an activity that AI researchers might understandably prefer to ignore and leave to others (see Zimmerman et al. 2007).

As Jameson, Spaulding, and Yorke-Smith (2009) point out in the introduction to this issue, designers of interactive intelligent systems are often best seen as searching for an optimal combination of intelligent algorithms and interaction design. In particular, new intelligent functionality may not work well within any familiar, accepted interaction style; on the other hand, designers do want to take advantage of what has been learned from decades of experience in interaction design for nonintelligent systems. Therefore, designers typically must move between (1) an exploratory research role and (2) a practitioner role, in which established methods and principles are used to design the user interaction and the interface. An additional complication is that there are often multiple dimensions in play—a solution for a particular problem may arise as a result of adjustments to the interface, the technology, or the user's workflow. Furthermore, improvements in the performance, precision, or accuracy of the underlying algorithms may have ramifications for the interface, the workflow, or both.

Exploratory Design

One challenge with intelligent interactive systems is that we are often designing for rapidly evolving technologies, whose current capabilities may be poorly understood and may advance even while we are working on their incorporation in a given system. As the HCI practitioners are iterating on the interaction design, the technologists may be improving the system's capabilities and its accuracy, precision, and speed. As the system's technical performance improves, a design that works well at a certain technical performance level may become unnecessarily clumsy, or it may fail to take advantage of new opportunities that arise. For instance, an interface for a slow question-answering system may focus on helping users to define their questions precisely or on allowing users to queue up several questions at once. As the system becomes faster, a typo in a question becomes less problematic, as the user can immediately fix it and re-pose

- Is a given approach likely to meet documented user needs?
- Is an approach analogous to an existing system or process?
- Does the user feel that he or she has sufficient control of the system?
- Does the system provide sufficient transparency such that the user has an appropriate understanding of what is happening?
- Does the system enable new workflows? What might these be?
- If the system demands user feedback or training data, is this collected in a manner that minimizes the cognitive load on the user?
- Is there an accuracy or efficiency level below which correcting mistakes is more costly than the perceived benefits?
- Is a mixed initiative design appropriate? If so, when and how should the hand-offs between the system and user occur?
- How might the system perform with a new user? With an advanced user?
- How long will it take the system to learn enough to adapt to a new user?
- Are there different usage models? For instance, some users may be more comfortable with more automation, others with less. Some users may want to delve into the guts of the application (set detailed parameters, and so on) while others may want minimal interaction.

Figure 1. Typical Exploratory Design Questions.

the question. Likewise, a question-queuing feature may become unnecessary, and it may serve only to complicate the interface to a system that can answer queries as fast as the user can pose them.

Designing on the basis of expected future capabilities of a system may reduce some of these problems, but it is often difficult to predict the future (and intermediate) performance accurately. In the question-answering system example, processing times of 10 minutes, 1 minute, and 1 second are all likely to lead to different optimal interaction designs.

One approach to this challenge, which was inspired by the work of Zimmerman, Forlizzi, and Evenson (2007) and realized in the group of the first author of this article, is to introduce an explicit exploratory design phase to help understand the design space and the trade-offs among the various possible solutions. The goal of this phase is to allow the design, technology, and workflows to be refined together. Rather than iteratively improve a single design, we explore the larger design space. This allows us to anticipate what solutions may work best in what cases, and it can even provide system performance benchmarks. For example, if the system can return an answer in less than 2 seconds, we do not need to build a question queuing system; if the response time is under half a second, a status log is not necessary to provide feedback on system progress and time remaining.

In the exploratory design phase, teams—typically consisting of technologists, designers, and HCI practitioners—orient themselves by walking through the knowledge that has been collected about the intended users and familiarizing themselves with the basics of the technology. This ensures that all team members have at least a basic understanding of the users' needs, goals, and contexts, as well as of the capabilities of the technology. Next, the teams develop a set of focus questions. Some typical questions are listed below; others may be relevant to the specific needs of a given project. Initial design concepts and prototypes are developed and used as the basis for formative studies (that is, studies whose purpose is to reveal possibilities for improvement rather than to assess precisely the current quality of the design). These studies can range from observations of users "thinking aloud" to technical evaluations of the performance of a particular component (for example, the question answering component). They can also include the type of need validation session discussed in the previous section. The process repeats until the focus questions have been answered satisfactorily.

Figure 1 lists some typical exploratory design questions; note that several of them refer to usability side effects discussed in the theme article by Jameson (2009).

A successful exploratory design effort should supply inspiration and motivation for what may be built and provide an informed understanding of the trade-offs among the various approaches.

Engineering for Change

A side effect of this multithreaded, iterative approach is that it increases the likelihood of significant design changes well into the development phase. Even if there is a careful technical separation between components of a system, accommodating these design changes can be difficult: While some user interface changes may be simple, other changes—say, adding a button to cancel a long-running operation—have implications that can reach well into the underlying architecture of a system and can be extremely difficult to address. A typical reaction of those responsible for implementation is "We can't change that!" (see, for example, John et al. [2004]). As is explained by these authors, one approach to this problem is to anticipate possible future user requirements even during the architecture design process. For example, an architecture can be chosen that will allow the introduction of a "Cancel This Operation" button later on, even if it is not yet clear whether such a button will be needed by users. This initial analysis can help extend the period of time in the development process in which the necessary changes are economically feasible, thus enabling a larger number of iterative design cycles.

Challenges in Evaluating Intelligent Systems

It is important to conduct user-centered evaluations as soon as the interaction design for a system has been specified at least on a high level. This process allows designers to assess whether the proposed design fulfills the complex set of requirements that has evolved so far and how those design plans may need to be adjusted. Recall that evaluation is not a separate, third phase in the design lifecycle but rather an activity that runs in parallel with the interaction design phase and with which it is closely intertwined. This interdependency between design and evaluation is especially worth emphasizing to an AI audience, because a common practice in the evaluation of AI-based systems with users—if it occurs at all—is to wait until the last phase of the project to conduct a user study (so as to demonstrate to the world how well the system works with and is accepted by users). The problem with this strategy is that, if such a study is the first user-oriented evaluation that is conducted, general usability problems may obscure the effectiveness of the underlying AI. These usability problems not only frustrate users, they can make it difficult for AI researchers to get actionable data on the capabilities of the AI. When the user study is conducted only at the end of a development cycle, there is often no time remaining to make the necessary improvements.

There exists a wide repertoire of methods for user-oriented evaluation (see the suggestions for further reading). These techniques vary in several dimensions: They may be more suitable for the early or for the later stages of the design process. They may focus more on objective variables such as speed of use and the occurrence of errors, or more on subjective variables such as users' opinions and emotional reactions. They may involve observation of users in strictly controlled conditions or in normal contexts of use. And they may concern brief encounters with a system or longer-term use.

The case studies in this special issue (in particular, those of Faulring et al. [2009] and Weber and Yorke-Smith [2009]) offer concrete examples of the application of such methods. Because there exists such a wide range of evaluation methods, instead of attempting to summarize them here, we will discuss some general challenges that must be borne in mind when evaluating systems that include some form of intelligence.

Presenting Intelligent Technology to Users before It Has Been Implemented

Low- and medium-fidelity prototypes are often used for acquiring early feedback from users on a new design conception. These prototypes convey the basic nature of the interaction with the system, even when the functionality has not been (completely) implemented. By way of comparison, for systems that do not include intelligent technology, adequate prototypes can often be developed with tools as simple as PowerPoint. However, when the processing involves intelligence, it may be necessary to include a human as part of the "prototype," employing the well-known Wizard-of-Oz method (see Horvitz et al. [1998], for an instructive example) in which the human simulates an intelligent system. The "wizard" can complete subtasks such as the recognition of user actions and generation of knowledgeable advice. The case study by Bunt, Conati, and McGrenere (2009) in this issue discusses additional approaches to this general problem.

Evaluating System Intelligence Separately

While success of an intelligent interactive system ultimately depends on the results of user interaction, it can be worthwhile to conduct separate technical evaluations of the intelligent components of the system. According to the binocular view introduced by Jameson, Spaulding, and Yorke-Smith (2009), this process requires temporarily "closing one eye" and adopting a monocular focus on the technology. The benefits of doing so include (a) that the technology can often be improved on the basis of the results of the technical evaluation and (b) that any limitations of the technology that cannot be removed (for example, an upper bound on the accuracy of action recognition) can at least be taken into account when alternative interaction designs are being considered and when the system as a whole is evaluated with human users.

More generally, the term *layered evaluation* is often used to refer to a separation of the evaluation of the system's intelligence from the evaluation of other aspects of the system design. Some authors

Further Reading

Overview

This website includes annotated links to a great variety of literature on HCI. See especially the "Recommended Readings" for works of general interest, like the ones listed below.

Perlman, G. 2009. HCI Bibliography: Human-Computer Interaction Resources. Website located at hcibib.org.

Some Widely Used HCI Textbooks

These textbooks are broad in scope, and they differ in style and focus. They all contain considerable material on methods of the sort described in this article.

Dix, A. J.; Finlay, J. E.; Abowd, G. D.; and Beale, R. 2004. *Human-Computer Interaction.* New York: Prentice-Hall.

Sharp, H.; Rogers, Y.; and Preece, J. 2007. *Interaction Design: Beyond Human-Computer Interaction.* West Sussex: Wiley.

Shneiderman, B., and Plaisant, C. 2009. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Reading, MA: Addison-Wesley.

Stone, D.; Jarrett, C.; Woodroffe, M.; and Minocha, S. 2005. *User Interface Design and Evaluation.* San Francisco: Morgan Kaufmann.

Handbook

This thick volume comprises dozens of chapters on a great variety of topics in HCI. A number of them focus on methods such as those discussed in this article. A selection of the chapters focusing on methods was republished in 2009 under the title *Human-Computer Interaction: Development Process* with the same editors and publisher.

Sears, A., and Jacko, J. A., eds. 2008. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications.* Boca Raton, FL: CRC Press.

Articles on Special Methodological Considerations for Intelligent Interactive Systems

Like the present article, the following papers consider some of the special considerations that arise in the design and evaluation of interactive intelligent systems (in the case of the first paper, more specifically: systems that adapt to their users):

Gena, C., and Weibelzahl, S. 2007. Usability Engineering for the Adaptive Web. In *The Adaptive Web: Methods and Strategies of Web Personalization*, ed. P. Brusilovsky, A. Kobsa, and W. Nejdl, 720–766. Berlin: Springer.

Höök, K. 2000. Steps to Take Before IUIs Become Real. *Interacting with Computers* 12(4): 409–426.

advocate a further splitting into additional layers. For example, in connection with systems that model and adapt to their users, it may be reasonable to evaluate separately (1) the acquisition and interpretation of data about the user; (2) the accuracy of the model of the user that is constructed by the system; and (3) how effectively the system exploits this user model when adapting to individual users (see, for example, Paramythis and Weibelzahl [2005]; Brusilovsky, Karagiannidis, and Sampson [2001]).

Evaluating Longer-Term System Use

As discussed in the theme article on usability side effects (Jameson 2009), during the phase of *initial* use an intelligent interactive system may offer limited (or even negative) net benefits to the user, because there has not yet been time for the necessary learning and adaptation on the part of the system and the user. On the other hand, some negative side effects of intelligence may appear only *after* extended use. For these reasons, it is especially valuable to conduct studies of system use that are more extended than the typical 1- or 2-hour study of initial use. A classic example is the study of a learning calendar assistant by Mitchell, Caruana, Freitag, McDermott, and Zabowski (1994), which revealed how the accuracy and usefulness of the assistant increased—and sometimes temporarily decreased—over a period of several months. When such extended studies are practically infeasible—as is often the case—it may be possible to compress the study into a shorter period of time by artificially speeding up the learning processes of the system or the user. For example, we might ask users to specify their preferences to the system explicitly at the beginning of the study, instead of requiring the system to learn them gradually through observation of users' actions. This approach was used in one of the evaluations described by Weber and Yorke-Smith (2009), in this issue.

Another approach to evaluation of long-term system use is to consider systematically, on a theoretical level, what sorts of changes are likely to occur between initial and later use of the system, leveraging accumulated experience with intelligent interactive systems. Several of the discussions of usability side effects presented by Jameson (2009) include ideas about typical changes over time. This sort of extrapolation from the results of a short-term study to the supposed results of a corresponding long-term study is obviously not as reliable as conducting the long-term study itself, but such a technique may be better than ignoring the fact that changes will occur over time or speculating about such changes in an unsystematic way.

Concluding Remarks

We have seen that many established HCI methods are of value for the design and evaluation of interactive intelligent systems, although there are a number of typical issues that need to be borne in mind and extensions and variants of the methods that ought to be considered. There is also room for new methods. Design research (see, for example, Zimmerman, Forlizzi, and Evenson [2007]) can yield new procedures that are relevant and extensible to practitioners who strive to design usable intelligent interactive systems. Behavioral and cognitive scientists can develop new theories and models of users' interactions with intelligent systems. Such models may reduce the extent to which empirical data need to be collected, and they may lead to improved ways of conducting empirical evaluations.

References

Beyer, H., and Holtzblatt, K. 1998. *Contextual Design: Defining Customer-Centered Systems*. San Francisco: Morgan Kaufmann.

Brusilovsky, P.; Karagiannidis, C.; and Sampson, D. 2001. The Benefits of Layered Evaluation of Adaptive Applications and Services. Paper presented at the Empirical Evaluation of Adaptive Systems Workshop at the Eighth International Conference on User Modeling, Sonthofen, Germany, July.

Bunt, A.; Conati, C.; and McGrenere, J. 2009. Mixed-Initiative Interface Personalization as a Case Study in Usable AI. *AI Magazine* 30(4).

Czerwinski, M.; Horvitz, E.; and Wilhite, S. 2004. A Diary Study of Task Switching and Interruptions. In *Human Factors in Computing Systems: CHI 2004 Conference Proceedings*, ed. E. Dykstra-Erickson and M. Tscheligi. New York: Association for Computing Machinery.

Davidoff, S.; Lee, M. K.; Yiu, C.; Zimmerman, J.; and Dey, A. K. 2007. Rapidly Exploring Application Design Through Speed Dating. In *Proceedings of the Ninth Internatinal Conference on Ubiquitous Computing* (Ubicomp 2007), 429–446. Berlin: Springer.

Faulring, A.; Mohnkern, K.; Steinfeld, A.; and Myers, B. 2009. The Design and Evaluation of User Interfaces for the RADAR Learning Personal Assistant. *AI Magazine* 30(4).

Horvitz, E.; Breese, J.; Heckerman, D.; Hovel, D.; and Rommelse, K. 1998. The Lumière Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. In *Uncertainty in Artificial Intelligence: Proceedings of the Fourteenth Conference,* ed. G. Cooper and S. Moral, 256–265. San Francisco: Morgan Kaufmann.

Jameson, A. 2009. Understanding and Dealing with Usability Side Effects of Intelligent Processing. *AI Magazine* 30(4).

Jameson, A.; Spaulding, A.; and Yorke-Smith, N. 2009. Introduction to the Special Issue on Usable AI. *AI Magazine* 30(4).

John, B. E.; Bass, L. J.; Sanchez-Segura, M.; and Adams, R. J. 2005. Bringing Usability Concerns to the Design of Software Architecture. Paper presented at the 9th IFIP Working Conference on Engineering for Human-Computer Interaction and the 11th International Workshop on Design, Specification, and Verification of Interactive Systems.

Lau, T. 2009. Why Programming by Demonstration Systems Fail: Lessons Learned for Usable AI. *AI Magazine* 30(4).

Lieberman, H. 2009. User Interface Goals, AI Opportunities. *AI Magazine* 30(4).

Mitchell, T.; Caruana, R.; Freitag, D.; McDermott, J.; and Zabowski, D. 1994. Experience with a Learning Personal Assistant. *Communications of the ACM* 37(7): 81–91.

Paramythis, A., and Weibelzahl, S. 2005. A Decomposition Model for the Layered Evaluation of Interactive Adaptive Systems. In *User Modeling: Proceedings of the Tenth International Conference,* ed. L. Ardissono, P. Brna, and A. Mitrovic, 438–442. Berlin: Springer.

Petrelli, D.; Dadzie, A.; and Lanfranchi, V. 2009. Mediating between AI and Highly Specialized Users. *AI Magazine* 30(4).

Weber, J., and Yorke-Smith, N. 2009. Designing for Usability of an Adaptive Time Management Assistant. *AI Magazine* 30(4).

Zimmerman, J.; Forlizzi, J.; and Evenson, S. 2007. Research through Design as a Method for Interaction Design Research in HCI. In *Human Factors in Computing Systems: CHI 2007 Conference Proceedings,* ed. B. Begole, S. Payne, E. Churchill, R. S. Amant, D. Gilmore, and M. B. Rosson, 493–502. New York: Association for Computing Machinery.

Aaron Spaulding is a computer scientist and interaction designer at SRI's Artificial Intelligence Center. His work centers on developing usable interfaces for AI systems that meet real user needs, and he was an organizer of the CHI 2008 Workshop on Usable Artificial Intelligence and the CHI 2009 Special Interest Group on Usable Intelligent Interactive Systems.

Julie Sage Weber is a doctoral student in computer science and engineering at the University of Michigan working in the areas of AI and human-computer interaction. Her research interests are in applications of artificial intelligence that benefit society, particularly through assistive technologies and information and communication technologies for development. She received a master's degree in computer science from Tufts University and a bachelor's degree in mathematics and computer science from Wellesley College. Her dissertation supervisor is Professor Martha E. Pollack, Dean of Michigan's School of Information.