# User Interface Goals, AI Opportunities

*Henry Lieberman*

■ *This is an opinion piece about the relationship between the fields of human-computer interaction (HCI) and artificial intelligence (AI). The ultimate goal of both fields is to make user interfaces more effective and easier for people to use. But historically, researchers have disagreed about whether "intelligence" or "direct manipulation" is the better route to achieving this. There is an unjustified perception in HCI that AI is unreliable. There is an unjustified perception in AI that interfaces are merely cosmetic. This disagreement is counterproductive.*

*This article argues that AI's goals of intelligent interfaces would benefit enormously by the user-centered design and testing principles of HCI. It argues that HCI's stated goals of meeting the needs of users and interacting in natural ways would be best served by application of AI. Peace.*

Is AI antithetical to good user interface design? From the earliest times in the development of computers, activities in human-computer interaction (HCI) and AI have been intertwined. But as subfields of computer science, HCI and AI have always had a love-hate relationship. The goal of HCI is to make computers easier to use and more helpful to their users. The goal of artificial intelligence is to model human thinking and to embody those mechanisms in computers. How are these goals related?

Some in HCI have seen these goals sometimes in opposition. They worry that the heuristic nature of many AI algorithms will lead to unreliability in the interface. They worry that AI's emphasis on mimicking human decision-making functions might usurp the decision-making prerogative of the human user.

These concerns are not completely without merit. There are certainly many examples of failed attempts to prematurely foist AI on the public. These attempts gave AI a bad name, at least at the time.

But so too have there been failed attempts to popularize new HCI approaches. The first commercial versions of window systems, such as the Xerox Star and early versions of Microsoft Windows, weren't well accepted at the time of their introduction. Later design iterations of window systems, such as the Macintosh and Windows 3.0, finally achieved success. Key was that these early failures did not lead their developers to conclude window systems were a bad idea. Researchers shouldn't construe these (perceived) AI failures as a refutation of the idea of AI in interfaces.

Modern PDA, smartphone, and tablet computers are now beginning to have quite usable handwriting recognition. Voice recognition is being increasingly employed on phones, and even in the noisy environment of cars. Animated agents, more polite, less intrusive, and better thought out, might also make a comeback. Researchers think that AI goals are important enough that it is worthwhile to persevere.

## The "Small Size of Infinity" and "Disappearing AI"

A historical perspective shows that ambitious AI goals for the interface, that may at first seem impossible, are often indeed accomplished over long periods of time. AI people attribute this to two well-known sociological phenomena: "the small size of infinity" and "disappearing AI."

The "small size of infinity" refers to the following dilemma. Looking at a major AI problem, like natural language understanding, or speech recognition, collecting human commonsense knowledge, it is easy to get discouraged. One might imagine that there are a virtually "infinite" number of ways to say something, an infinite number of facts that people know, or an infinite number of accents or speech patterns. The question is, just how infinite? In many domains, nobody knows.

Trying to classify things into a few categories, as early AI programs did, simply might not work, accounting for many of the early failures. But it may (or may not) also turn out that there aren't an infinite number of variations, just a very large, but manageable number.

Over time, Moore's law makes large numbers feasible more rapidly than people can intuitively grasp. Thus statistical language models are beginning to have some success in natural language processing and speech recognition. Since a person only lives for about three billion seconds, and it is unlikely that human learning takes place at a faster rate than a small number of chunks of knowledge a second, it's not impossible that a contemporary computer could store as much knowledge as the average person has.

The "disappearing AI" problem is that, as soon as AI achieves a success, that success is assimilated into the application domain, and so AI, as a field, finds it difficult to claim credit for the success.

Now, it's hard to buy a US $100 scanner that does *not* come bundled with good optical character recognition software, but that was the subject of multimillion-dollar (D)ARPA research projects in the 1960s and 1970s. People don't think of search engines like Google as AI, but they do use many text-processing techniques, such as stemming, part-of-speech tagging, and others that were originally developed for general natural language understanding. Now, that's just considered part of information retrieval. Spam filters using Bayesian learning are effective. The Trackpoint pointing stick on the original IBM ThinkPad laptops depended on a predictive model of human cognitive and physiological capabilities to improve its ability to make precise GUI selections. Descendents of expert systems approve your credit card purchases. Recommender systems on Amazon, Netflix, and other sites are now just "electronic commerce." Most people think of a Roomba more as a household appliance than a robot.

## Taming the Savage Beast of AI

Many AI projects do suffer from poor user interface design. This is often a consequence of the haste in which user interfaces to AI systems are put together by people whose main interest is, say, in working on the details of a learning algorithm rather than the interface itself.

This is a shame, however, because it can lead to poor results in user tests and consequent rejection of the very idea of the utility of AI by end users and funding sources. Suitable attention to the principles of user-centered design (Norman and Draper 1986, Shneiderman and Plaisant 2005) can result in a dramatic difference in user acceptance of AI systems. These principles involve studying users in advance to determine their goals, needs, and work practices; and iterative testing and redesign after initial implementation to understand how personal and social contexts affect the success of computer use.

This simultaneous attention to the AI and HCI aspects of an application is the cornerstone of the "binocular" view advocated by Jameson (Jameson 2009). The history of several recent AI projects aimed at augmenting the user interface shows how attention to HCI issues can reap benefits. They generally started out with a simple plan of embedding some kind of predictive algorithm within a conventional user interface. They initially encountered some confusion on the part of users, of the sort that might have led conventional HCI practitioners to abandon the idea of using prediction.

However, the developers responded by providing more user feedback in the system, visualizations, more explanatory facilities, better setting of user expectations, deeper understanding of the context of the user, and other improvements that drastically improved usability. This story played out with systems such as RADAR (Steinfeld et al. 2007), CALO (Weber and Yorke-Smith 2008), and the systems Sheepdog, DABL, and CoScripter, described by Lau (2009).

Furthermore, some of the characteristics of AI programs perceived negatively by HCI evaluators actually represent *trade-offs*. Rather than consider unpredictability of AI systems as a reason to reject AI, it is more productive to admit that there might be a trade-off between predictability of the interface, admittedly good for consistency, and the convenience afforded by prediction of user intent. Systematic work can evaluate this trade-off and provide ways of managing it, as shown in the case studies by Bunt (Bunt, Conati, and McGrenere 2009) and Findlater and Gajos (Findlater and Gajos 2009).

But one of the best reasons for AI to pay more attention to HCI is that it will result in improving the science of AI itself. After all, AI is trying to imitate human reasoning, so shouldn't researchers study people, the best example so far of intelligent systems, to see how it works in practice? HCI's "feedback loop" of observing users, designing systems around their needs and behavior, then testing them with users provides a great testbed for discovering the principles behind the intelligent problem solving it is hoped to achieve with AI systems. AI has a long tradition of being inspired by, and feeding into, cognitive science, for its understanding of human behavior. HCI is, in some measure, a kind of "applied cognitive science," applied to the interaction between people and computer hardware and software as it exists today.

## Challenges in HCI Design for AI Applications

Because AI systems may contain complicated decision-making algorithms, such as learning or inference, they pose special interface-design problems that are absent, or less serious, for more conventional tool-based interfaces. Conventional design and evaluation methodologies might need rethinking when it comes to AI interfaces. Rather than use these challenges as a reason to reject AI, HCI should meet these problems head-on. Progress on these issues will undoubtedly reap benefits for more conventional interfaces as well.

First, the underlying complexity of AI algorithms means that interface design needs to pay special attention to transparency and explanation. Since complex algorithms might indeed be "smarter than the user" (and therein may lie their value), both the programs and their users may need to invest some effort in communication about what the program does and how it can be of service to its users. This means AI programs might require more tutorial interfaces or progressive introduction of features in order to acquaint the user with its capabilities. Users may initially need some patience with the program until they understand how its capabilities may be used most effectively.

Shneiderman (1989) has famously argued that the use of AI in interfaces reduces predictability, which he believes to be essential for usability. That may indeed be so in some instances, but predictability is not the only determinant of usability. Conventional direct-manipulation software has already outstripped the limits of predictability. When you insert a picture into Microsoft Word, can you predict exactly where it will be placed? While predictability correlates well with usability for simple interfaces, for complex interfaces like the Windows or Macintosh interface, reasonableness of behavior is a better criterion, rather than predictability by itself. And adding learning, adaptation, and domain knowledge may well increase predictability, and usability, rather than decrease it.

In order for users to develop trust in the program, they have to be able to understand it at some level (if not fully). That is, the program must be in a position to explain what it did and why it did what it did. Fortunately, self-awareness and reflection in programs has been a topic of study in AI, but perhaps it deserves more emphasis when AI meets the user interface. Explanation generation has also been a topic of study in AI, and more of what AI has learned in this area needs to be mobilized for effective AI interfaces. Visualization techniques in HCI serve as a nice complement, and also increase the level of understanding that users have of the operation of a program.

Next, AI interfaces invest more in long-term interactions than the more short-term perspective of conventional interfaces. Learning algorithms may need to be "trained up" before the benefit of their learning becomes apparent. Systems that can accept user feedback or end-user programming may need a while for that advice to take effect or for the user to determine what kinds of feedback work best. Algorithms that are knowledge intensive or data intensive may need the time to acquire sufficient knowledge so that the aggregative or inferential capabilities of their algorithms demonstrate their full effect.

Finally, evaluation of AI interfaces is challenging. Conventional interface evaluation makes assumptions that AI interfaces tend to break. As noted above, AI interfaces may not do so well in short-term comparisons, since they may need time to develop the system-user collaboration that is essential to their success. Longitudinal studies are better. AI interfaces may be so different from conventional interfaces that there may not be any easy way to make side-by-side comparison with existing conventional interfaces. They may be directed at problems that conventional interfaces do not even attempt to tackle. Personalized interfaces may make it difficult to compare performance across different users. Knowledge-based interfaces may perform better on tasks where coverage of their knowledge is better, so any evaluation is relative to the program's current state of knowledge. But none of these challenges are insurmountable.

The issues discussed in the section are dealt with in more detail in the theme articles on usability side effects (Jameson 2009) and usability engineering methods (Spaulding and Weber 2009) in this issue.

## Taming the Savage Beast of HCI

HCI, like AI, has a problem with the gap between what it promises and what, in the short term, it can reasonably deliver. HCI holds out to the public the promise of easy-to-use, well-designed systems that meet real user needs. Many companies, despite considerable investment in best-practice HCI technologies, still have difficulty achieving ease of use in practice. Microsoft, for example, has about 8,000 people testing software (Page, Johnston, and Rollison 2008), yet routinely releases software with hundreds of known usability problems, many serious. If AI has to answer for "Where are those intelligent computers you keep talking about after all these years," HCI also has to answer for "Why do you keep giving us so many hard-to-use interfaces after all these years?"

Fortunately, AI and HCI can team up to fulfill the promise of both fields. What this means is that neither field has to go it alone. AI interfaces don't have to solve the user's problem entirely by themselves; they can fall back on explicit direct-manipulation interaction when they don't know what to do or need confirmation or correction. HCI interfaces can use AI techniques to recognize the user's intent implicitly, so they don't have to provide explicit menu or icon operations for every single thing the user might want to do.

## Revisiting HCI's Stated Goals

In the following subsections, I will revisit some of the stated goals of HCI in detail and show how AI provides opportunities and techniques that could potentially contribute to these goals.

**Systems Should Help Users Accomplish their Personal Goals.** It is a tenet of the philosophy of user-centered design in HCI that systems should be oriented toward helping users achieve their personal and work goals and tasks. Ethnographic and other kinds of studies in HCI are aimed at observing users in their natural work or home contexts; understanding what they are trying to do in their daily activities; and trying to align the design of computer interfaces in order to support those goals.

This stands in contrast to a more technology-centered design process, common in industry. In technology-centered design, an engineer might draw up a list of product features that they think users would find attractive or useful. They then implement those features and convey them to the users in advertisements and user manuals. This runs the risk that the implemented features may not in fact be useful or even understandable to the users, or that capabilities critical to the users will remain missing.

In ethnographic or other kinds of user studies, recognition of user goals is performed manually by HCI personnel, who then interact with the programmers of the systems. But AI can contribute the ability to perform automatic goal recognition from direct observation of the users' actions. Partial order planning can provide recognition of plans for accomplishing these goals, and plan generation can aid production of user help, guides, or automated assistants for these plans (Rich et al. 2005, Lieberman and Espinosa 2006). User studies can provide the vocabulary and framework for the AI systems, and the AI systems can provide ways to concretely implement the insights of these studies directly in working systems.

**Systems Should Be Designed to Fit in with Users' Existing or Desired Work Practices.** Studies, especially in the HCI subfield of computer-supported cooperative work, often seek to describe the real-world work practices of users and to assure that socio-technical design of HCI systems is consonant with these practices.

AI provides a means to explicitly represent such work practices as workflow descriptions and document content in both procedural and declarative form, giving life to the representations that are otherwise coded only implicitly. It also provides the opportunity for systems to do reflection or inference on the content of these practices themselves.

There are untapped similarities between the kind of work done in ethnographic studies and knowledge engineering in AI. Knowledge-acquisition tools, such as those often presented at the annual Knowledge Capture (K-Cap) conference, might prove useful. This might give rise to a new subfield of interface knowledge engineering, spanning both AI and HCI.

**Technology Should Adapt Itself to the Users, Not the Other Way Around.** Again, HCI recommends studying users and transferring the results from HCI professionals to user interface designers and implementors at design time. While it is certainly good to do this, AI provides the potential to move at least part of this feedback loop from design time to run time. Users' needs constantly change, so building in an adaptive capability gives the system flexibility as the need arises, without having to wait for the next user study or the next version of the software. There are many examples of AI systems that learn from user actions in the interface in real time (Lieberman 2001).

**Interfaces Should Accomplish Interactions in Ways That Make Efficient User of User Input.** HCI has long investigated models such as GOMS (goals, operators, methods, and selection rules) or the keystroke-level model (Card, Moran, and Newell 1983) that provide tools for modeling the relationship between user tasks and the user interface operations needed to accomplish them in a given interface. These models serve as measures for evaluating the efficiency of user interfaces. Users should be able to get the most of what they want done, while having to provide the least input to a system.

Many AI technologies can act as "input amplifiers." They can let the users state their intent at a high level and translate it into the (perhaps many) low-level operations needed to accomplish the task. Natural language and speech recognition interfaces can let the users "just say" what they want, and the system can interpret the users' intent. Knowledge-based interfaces provide domain-specific common vocabularies between the machine and the user that can aid communication about what needs to be done. Recording interfaces such as programming by example or other learning interfaces can watch what the user does and offer to automate similar actions in the future. Intent recognition and intelligent defaults can get users "in the neighborhood" of what they want to do, where further direct-manipulation operations can finish arriving at the goal. All these techniques can avoid getting users stuck in "target practice"—long chains of repetitive direct-manipulation operations to accomplish high-level tasks.

AI approaches can also achieve more flexibility in input to computer systems. Conventional HCI interfaces tend to be rigid in the order in which they accept instructions and the level of detail that they require. Human communication, though, tends to be more flexible, in that people can freely mix communicating high-level intention and advice with detailed and specific instructions or questions. Recognition of intent, classification of speech acts, learning, and mixed-initiative interfaces can let systems accept a broader range of human input and be more forgiving of people's idiosyncrasies.

**Interfaces Should Use Methods of Input and Output That Are Natural for the Human User.** Graphical direct-manipulation interfaces were a big advance because they allowed users to provide input and receive output from systems by making use of real-world metaphors—icons representing files, and so on. The kind of input and output most natural for people is that which engages their natural ability to sense and interact with the world.

A considerable amount of work in AI surrounds natural language understanding, speech recognition, and computer vision, including face recognition and gesture recognition. Now that cheap and capable sensors are becoming common in ubiquitous computing, activity recognition—trying to understand what high-level activities people are performing from sensor input—is also fast becoming an active area.

Traditional HCI design methods have been reluctant to incorporate natural input methods such as unconstrained natural language input or unconstrained speech input, because designers worry that these systems will make mistakes in interpretation that will leave users frustrated. That is indeed a danger, and the scope of such systems needs to be carefully considered.

In the past, some HCI projects have felt "burned" by overambitious AI claims to be able to interpret unconstrained natural language or visual input; many such systems eventually didn't test well in practice. Such initial failures were sometimes met by claims that just a little more work was needed; often, those secondary claims also proved illusory. But, as a community, we have gotten too used to the idea that "they'll never work" and have largely abandoned the attempt to solve these general problems.

In fact, over long periods of time since the initial AI claims of the 1970s, interpretation of natural input modalities has made steady progress. Through a combination of better understanding of the cognitive problems, and the inexorable march of Moore's law, approaches that didn't work decades ago are beginning to work now. Directory assistance uses speech recognition to respond to requests for phone numbers. Consumer cameras incorporate face recognizers and red-eye correction. In many cases, HCI design approaches can provide application- and user-context constraints that reduce demand on the general algorithms, improving reliability. Fail-soft approaches can fall back on explicit user interaction, such as the "How May I Help You" speech recognition that routes failures in automatic recognition to a human respondent (Gorin, Riccardi, and Wright 1997).

**Interfaces Should Facilitate Collaboration.** Computer-supported cooperative work (CSCW) has long been an active subfield of HCI, whose goal is to use the computer as a facilitator to mediate communication between people. Social software of all kinds is now undergoing an explosion as a domain for HCI work. An ongoing theme of HCI is the use of the computer to encourage collaboration.

While traditional HCI focuses on collaboration between remotely-located people, AI brings to the picture collaboration between the user and the computer, with the computer taking a more active role. Think of the computer as being one of the collaborators, rather than simply a conduit for collaboration between people. Mixed-initiative interfaces, where the user and the system share the initiative, perhaps at different times, mean that the division of labor between the users and the system can be decided dynamically (see the case study by Bunt, Conati, and McGrenere [2009] in this issue).

Many of the collaboration issues that come up in human collaboration—understanding and communicating goals, generating and recognizing plans for accomplishing these goals, understanding when things are working, and correcting them when they are not—also come up in AI when trying to model machine problem solving. A better understanding of these issues can help us improve interfaces for collaboration. Two heads are better than one, even if one of those heads happens to be a machine.

## Related Work

This article is not the first to explore the relationship between HCI and AI. Jameson's article in this issue (Jameson 2009) and several previous papers (Lieberman and Maulsby 1996, Horvitz 1999, Isbell and Pierce 2005) have put forward the notion that there is a smooth spectrum between HCI and AI. Gradually increasing doses of knowledge, inference, proactivity, and autonomy move along this spectrum. At the HCI end, virtues of simplicity, control, and predictability prized by HCI researchers dominate. At the AI end, convenience for the user, power of knowledge and inference, and ability to deal with complexity are the advantages.

In contrast, taking a perspective more skeptical of AI, Winograd (2006) tries to separate the viewpoints as much as possible. He identifies a "rationalistic view" versus a "design view," before coming down on the design side. We don't view rationality and design as being opposed, obviating the need for a sharp separation.

Once the notion of a convenience versus control spectrum is accepted, the discussion shifts to the criteria for making these trade-offs. The Horvitz article in particular supplies principles for determining the trade-offs in what AI refers to as mixed-initiative interfaces. The balance may depend not only on the task, but even on the user's personal preferences, style, and risk tolerance. Unfamiliar interfaces might start out with low levels of AI, gradually ramping it up as the user becomes more familiar with the system and trusts it more. Lieberman and Maulsby discuss the "sweet spot" in the spectrum where an intelligent agent has both the capability to automate a task, and to be explicitly instructed by a user.

## Integrating AI into Interfaces

In summary, the future is bright for integrating AI into user interfaces, combining the best of user-centered design with the representation, reasoning, and recognition that AI has to offer.

A recent development in the AI community is that more and more of AI software is being packaged into easily integratible toolkits that make it easier for interface implementors to experiment with AI capabilities. Examples are Intel's OpenCV for computer vision (Bradski and Kaehler 2008), the semantic web movement, and its reasoners such as Racer (Haarslev and Möller 2003), commonsense knowledge bases such as OpenMind / Divisi (Lieberman et al. 2004; Speer, Havasi, and Lieberman 2008) and OpenCyc / ResearchCyc,[1] and machine-learning packages such as Weka (Witten and Frank 2005). A recent CHI paper explores the developer experience integrating one AI package into an interface (Patel et al. 2008).

The AI community needs to pay more attention to HCI issues, rather than just concentrate on algorithms and leave the interface "until later." The HCI community needs to get over its paralyzing fear of AI failures and take a hard look at the opportunities for making interfaces more intelligent. Researchers need to celebrate the interface successes that AI has already had, getting the interface community to look past "disappearing AI." Together, the community can make user interfaces a little less stupid and frustrating than they are today.

## Notes

[1] See Overview of Open Cyc, www.cyc .com/cyc/opencyc/overview, March 2009.

## References

Bradski, G., and Kaehler, A. 2008. *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA: O'Reilly Press.

Bunt, A.; Conati, C.; and McGrenere, J. 2009. A Mixed-Initiative Approach to Interface Personalization. *AI Magazine* 30(4).

Card, S. K.; Moran, T. P.; and Newell, A. 1983. *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Erlbaum.

Glass, A.; McGuinness, D. L.; and Wolverton, M. 2009. Toward Establishing Trust in Adaptive Agents. In *Proceedings of the 2008 International Conference on Intelligent User Interfaces (IUI2008)*, Canary Islands, Spain, January 13–16, 2008. New York: Association for Computing Machinery.

Findlater, L., and Gajos, K. Z. 2009. Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces. *AI Magazine* 30(4).

Gorin, A., L.; Riccardi, G.; and Wright, J. H. 1997. How May I Help You? *Speech Communication* 23(1–2): 113–127.

Haarslev, V., and Möller, R. 2003. Racer: A Core Inference Engine for the Semantic Web. Paper presented at the Second International Workshop on Evaluation of Ontology-based Tools (EON-2003), Sanibel Island, FL.

Horvitz, E. 1999. Principles of Mixed-Initiative User Interfaces. In *Proceedings of the ACM Conference on Human-Computer Interaction (CHI-99)*, 159–166. New York: Association for Computing Machinery.

Isbell, C., and Pierce, J. 2005. An IP Continuum for Adaptive Interface Design. In *Proceedings of the Eleventh International Conference on Human-Computer Interaction (HCII)*. New York: Association for Computing Machinery.

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

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

Lieberman, H. 2001. *Your Wish Is My Command: Programming by Example*, San Francisco: Morgan Kaufmann, 2001.

Lieberman, H., and Espinosa, J. 2006. A Goal-Oriented Interface for Consumer Electronics Using Planning and Commonsense Reasoning. In *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI-06)*, Sydney, Australia, January. New York: Association for Computing Machinery.

Lieberman, H., and Maulsby, D. 1996. Instructible Agents: Software That Just Keeps Getting Better, *IBM Systems Journal* 35(3–4): 539–556.

Lieberman, H., Singh, P.; Liu, H.; and Barry, B. 2004. Beating Common Sense into Interactive Applications. *AI Magazine* 25(4): 63–76.

Norman, D., and Draper, S. 1986. *User-Centered System Design*. Hillsdale, NJ: Lawrence Erlbaum.

Page, A.; Johnston, K.; and Rollison, B. 2008. *How We Test Software at Microsoft*. Redmond, WA: Microsoft Press.

Patel, K.; Fogarty, J.; Landay, J.; and Harrison, B. 2008. Investigating Statistical Machine Learning as a Tool for Software Development. In *Proceedings of the ACM Conference on Human-Computer Interaction (CHI-08)*, Florence, Italy, April. New York: Association for Computing Machinery.

Rich, C.; Sidner, C.; Lesh, N.; Garland, A.; Booth, S.; and Chimani, M. 2005. DiamondHelp: A Graphical User Interface Framework for Human-Computer Collaboration. In *Proceedings of the IEEE International Conference on Distributed Computing Systems Workshops*, 514–519, June. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Speer, R.; Havasi, C.; Lieberman, H. 2008. AnalogySpace: Reducing the Dimensionality of Commonsense Knowledge. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence* (AAAI-08), Chicago, July. Menlo Park, CA: AAAI Press.

Shneiderman, B. 1989. Intelligent Interfaces: From Fantasy to Fact. In *Proceedings of the IFIP Eleventh World Computer Congress,* San Francisco, August. Laxenburg, Austria: The International Federation for Information Processing.

Shneiderman, B., and Plaisant, C. 2005. *Designing the User Interface*, 5th ed. Reading, MA: Addison Wesley.

Spaulding, A., and Weber, J. 2009. Usability Engineering Methods for Interactive Intelligent Systems. *AI Magazine* 30(4).

Steinfeld, A.; Bennett, R.; Cunningham, K.; Lahut, M.; Quinones, P.-A.; Wexler, D.; Sieworek, D.; Hayes, J.; Cohen, P.; Fitzgerald, J.; Hansson, O.; Pool, M.; and Drummond, M. 2007. Evaluation of an Integrated Multi-Task Machine Learning System with Humans in the Loop. *NIST Performance Metrics for Intelligent Systems Workshop*. Washington, D.C.: National Institute of Standards and Technology.

Weber, J., and Yorke-Smith, N. 2008. Time Management with Adaptive Reminders. Paper presented at the Workshop on Usable Artificial Intelligence, ACM Conference on Human-Computer Interaction (CHI-08), Florence, Italy.

Winograd, T. 2006. Shifting Viewpoints: AI and HCI. *Artificial Intelligence Journal*, 170, pp. 1256-1258, Elsevier.

Witten, I., and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. San Francisco: Morgan Kaufmann Publishers.

*Henry Lieberman* is a research scientist at the Media Lab of the Massachusetts Institute of Technology, where he heads the Software Agents Group. His principal research interest at the moment is commonsense reasoning for interactive applications. He has written over 100 research papers, edited or coedited three books, and holds a Habilitation degree from the University of Paris VI. He was twice program chair of the ACM Conference on Intelligent User Interfaces (IUI). He spends his time going to conferences like AAAI, where he exhorts people to pay more attention to interface issues, and interface conferences like CHI, where he exhorts people to pay more attention to AI. He can be reached at lieber@media.mit. edu.