Editor: Vince Stanford ■ vince-stanford@users.sourceforge.net

Toward a Framework for Evaluating Ubiquitous Computing Applications

Jean Scholtz and Sunny Consolvo

EDITOR'S INTRODUCTION

By integrating object-oriented programming, an event-driven windowed desktop, the mouse, Ethernet, and laser printers in a single platform, the Xerox Alto defined computer interfaces for a generation after its introduction in 1973. The computer industry spent much of the following 20 years copying and refining this classic benchmark. Only in the late 1990s did pervasive computing—featuring compact, portable communication, processing, and interface elements—come forth with truly fundamental new advances.

Key issues of good system design, integration, and especially usability evaluation remain rich pervasive computing research areas. This department has covered many usability and quality factors of pervasive systems, including the restaurant ordering system (Jan.–Mar. 2003), where we discussed acceptance in terms of ergonomics, cognitive loading, stakeholder groups, workflow improvements, and economics. Earlier, we looked at privacy and trust issues in health care information systems (Apr.–Jun. 2002). Attention, conceptual models, and workflow in relation to a wearable voice communication badge caught our attention late last year (Jul. –Sept. 2003).

This installment presents a framework for evaluating pervasive applications, which offers a systematic way of looking at key usability and acceptance issues we have discussed extensively but intuitively. I hope you enjoy it as much as I did. *—Vince Stanford*

Although they routinely evaluate pervasive or ubiquitous computing applications, researchers have difficulty comparing results rigorously and quantitatively. The lack of a widely accepted framework for user evaluations of ubiquitous applications hampers their efforts. By proposing such a framework, we hope to help researchers compare results, create ubiquitous computing design guidelines, develop effective discount evaluation techniques, understand the appropriateness of different evaluation techniques, and develop a more complete structure so they can avoid overlooking key areas of evaluation.

THE NEED FOR A FRAMEWORK

To improve comparability across research efforts, a user evaluation framework must be developed specifically for ubiquitous computing. A framework can create explicit structures, which can be made complete and comprehensive by repeated investigation over time. It can also contribute a consistent terminology to describe results, which should facilitate sharing and help establish design guidelines and techniques applicable to different evaluation areas.

Eventually, it should also lead to the development of ubiquitous computing-specific discount evaluation techniques for quicker and less costly evaluations, exemplified by recent work on heuristic evaluation of ambient displays by Jennifer Mankoff and her colleagues.[1] The importance of a common framework is underlined by the numerous attempts made to structure ubiquitous computing evaluations. As we will see, some focus on particular areas, such as sensing systems, while others focus on areas such as values.

Anthony Jameson proposed five usability challenges for adaptive interfaces: predictability and transparency, controllability, unobtrusiveness, privacy, and breadth of experience. His work focuses on usability and adaptive interfaces, or systems that learn from the user's behavior and react accordingly.[2]

Victoria Bellotti and her colleagues suggest five interaction challenges for designers and researchers of sensing systems:

- *Address*—directing communication to a system
- *Attention*—establishing that the system is attending
- *Action*—defining what is to be done with the system
- *Alignment*—monitoring system response
- *Accident*—avoiding or recovering from errors or misunderstandings[3]

They focus on challenges for the system designer and communicative aspects of interaction in sensing systems—specifically, interactions that are not GUI-based.

Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/04/$20.00

THE UBIQUITOUS COMPUTING MODEL

The considerable differences between the mature field of desktop computing and the newer ubiquitous computing model have presented difficulties to researchers attempting to directly apply many of its evaluation methodologies and guidelines.

Mark Weiser's vision was of ubiquitous computing so integrated into everyday objects that it becomes invisible to users.[1] Today, diverse applications exist, ranging from help for commuters to find train and bus schedules to smart laboratories, smart museums, and instrumented classrooms.[2–5]

Tom Moran and Paul Dourish note that ubiquitous computing efforts "move the site and style of interaction beyond the desktop and into the larger real world where we live and act."[6] They also suggest that "the design challenge, then, is to make computation useful in the various situations that can be encountered in the real world—the ever changing context of use." So ubiquitous computing environments often have more stringent and constrained usability requirements; this challenging design environment motivates our user evaluation framework.

The questions of how one user's interactions might affect another user, and how and if ubiquitous computing impacts the normal social situation are also important.[4] Modalities might include speech, gestures, and even physical interactions. Likewise, feedback to users is not limited to a particular display, or any display at all, and user behavior can cause physical world actions. For example, lying down in an intelligent room could cause drapes to close, lights to dim, and music to fade out.[7] Moreover, both input and output in a ubiquitous computing environment can be distributed.

As with desktop computing, direct and indirect stakeholders must be considered:

> *Direct stakeholders refer to parties—individuals or organizations—who interact directly with [the system] or its output. Indirect stakeholders refer to all other parties who are affected by the use of the system. Often, indirect stakeholders are ignored in the design process.*[8]

For the general public to adopt ubiquitous applications, it is crucial for evaluators to consider all stakeholders, not just direct ones.

Many ubiquitous applications are context-aware, with the application's behavior changing with user activity. Anind Day, Daniel Salber, and Gregory Abowd define context as "any information that characterizes a situation related to the interaction between humans, applications, and the surrounding environment."[9] In practice, different types of sensory data serve to infer context; for example, user location as a contextual attribute is used in context-aware applications such as mobile tour guides.[10,11]

REFERENCES

1. M. Weiser, "The Computer for the 21st Century," *Scientific Am.*, vol. 265, 1991, pp. 94–104.
2. T. Lunde and A. Larsen, "KISS the Tram: Exploring the PDA as Support for Everyday Activities," *Proc. 3rd Int'l Conf. Ubiquitous Computing*, Springer-Verlag, 2001, pp. 232–239.
3. L.F. Arnstein et al., "Labscape: Design of a Smart Environment for the Cell Biology laboratory," *IEEE Pervasive Computing*, vol. 1, no. 3, 2002, pp. 13–21.
4. M. Fleck et al., "Rememberer: A Tool for Capturing Museum Visits," *Proc. 4th Int'l Conf. Ubiquitous Computing*, Springer-Verlag, 2002, pp. 48–55.
5. J.A. Brotherton and G.A. Abowd, "Lessons Learned from eClass: Assessing Automated Capture and Access in the Classroom," to be published in *ACM Trans. Computer-Human Interaction*, 2004.
6. T. Moran and P. Dourish, "Introduction to This Special Issue on Context-Aware Computing," *Human-Computer Interaction*, vol. 16, nos. 2–4, 2001, pp. 87–97.
7. R.A. Brooks, "The Intelligent Room Project," *Proc. 2nd Int'l Cognitive Technology Conf.*, IEEE CS Press, 1997, pp. 271–278.
8. B. Friedman and P.H. Kahn, Jr., "Human Values, Ethics, and Design," *The Human-Computer Interaction Handbook*, Lawrence Erlbaum Assoc., 2003, pp. 1177–1201.
9. A. Dey, G. Abowd, and D. Salber, "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Human-Computer Interaction*, vol. 16, nos. 2–4, 2001, pp. 97–166.
10. G. Abowd et al., "Cyberguide: A Mobile Context-Aware Tour Guide,." *ACM Wireless Networks*, vol. 5, no. 3, 1997, pp. 421–433.
11. S. Feiner et al., "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment," *Personal Technologies*, July, 1997, pp. 208–217.

Batya Friedman, Peter Kahn, and Alan Borning suggest 12 key human values with ethical import: human welfare, ownership and property, freedom from bias, privacy, universal usability, trust, autonomy, informed consent, accountability, identity, calmness, and environmental sustainability.[4] Their values serve the entire human-computer interaction field, including Web sites, traditional desktop software, and pervasive computing.

We have incorporated areas from this research, and from desktop computing, into our proposed framework. The "Ubiquitous Computing Model" sidebar discusses the context for our work on frameworks.

A FRAMEWORK OF UBIQUITOUS COMPUTING EVALUATION AREAS

We have developed a set of evaluation areas and sample metrics and measures, which we call u*biquitous computing* e*valuation* a*reas* (UEAs), assembling them from personal experiences, input from colleagues, and the literature. Our framework presents several metrics and conceptual measures. A *conceptual measure* is an observable value. A *metric* associates meaning to that value by applying human judgment. We use the term conceptual measure as opposed to implementation-specific measure. An evaluator using this framework must decide how to collect particular conceptual measures; that instantiation becomes the implementation-specific measure.

To use the proposed framework, evaluators must identify groups of users who will be affected by the application—the direct and indirect stakeholders.[4] The evaluator must also decide if she needs to establish a baseline or control group. This offers a means of comparing the technology under evaluation to the user's normal environment.

Table 1 and the "Ubiquitous Computing Evaluation Areas" sidebar discuss these UEAs in detail:

- Attention
- Adoption
- Trust
- Conceptual Models
- Interaction
- Invisibility
- Impact and Side Effects
- Appeal
- Application Robustness

INTERPRETING UEA METRICS

In typical desktop usability evaluations, measures of effectiveness, efficiency, and user satisfaction are not equally important. If applications are discretionary, emphasis might be on user satisfaction; if time-critical, efficiency is optimized; when errors are unacceptable, as in life-critical situations, effectiveness is most important. However, the unique characteristics of pervasive applications might require more comprehensive measurements. We offer some predictions:

- "Walk-up-and-use" systems should score well in interaction and conceptual models
- Systems for social settings should score well in interaction and conceptual models, but have low side-effect scores
- Personal information processing should score high in trust
- Life- or time-critical processes should score well on interaction and attention
- Context-aware systems might have low scores in predictability and conceptual models, but high efficiency and effectiveness scores
- Ambient displays should score well in the areas of attention and invisibility

To assess our framework's utility, we have fit several published evaluations by practicing researchers to our framework in the case studies presented next.

THE CAMPUS-AWARE SYSTEM

Jenna Burrell and her colleagues developed a campus tour guide that tracks user location and provides information about surroundings.[5] Developed for visitors, primarily prospective students, it displays a map on a PDA showing notes contributed by students, faculty, and staff. The evaluation focused on the annotation features, the balance of attention between the device and the physical environment, and context-aware device functionality. Data collected included observations, feedback from subjects, and the number and categories of notes created. Here, we list several of their findings, suggesting in parentheses appropriate UEAs from our framework and their related metrics:

- The devices were distracting. Users looked at the devices, not local surroundings. (Attention—focus; Interaction—distraction)
- Users contributed an average of 3.7 notes. Over half factual, but opinions and advice were also posted. (Adoption—value)
- Users contributed because it was easy and they found value in the information posted by others. But not all notes were judged to be accurate or useful. (Adoption—value; Interaction—effectiveness and user satisfaction)
- Users wanted more functionality. For example, directions to other places on campus were needed but unavailable from the system. (Adoption—value)
- The application did not always sense user location accurately, which caused confusion about the buildings actually described by the notes. (Invisibility—accuracy)

Additionally, scalability should be considered, as described in the Interaction UEA, because many individuals will be providing and reading notes simultaneously. The Trust and the Impact and Side Effects UEAs should also be examined. How do users feel about leaving opinions as notes without knowing who will view them later? Will there be less interaction between visitors and residents because information is available electronically? Will this eliminate the need for campus tours?

PERSONAL INTERACTION POINTS SYSTEM

Personal interaction points (PIPs) let users personalize shared devices such as fax machines, printers, and copiers.[6] The investigators studied customization with embedded displays in the environment or portable devices. The embedded approach used touch screens and the portable one used cell phones. These were used with a large plasma display, a copier, and a conference room PC. The evaluation reported on usability, utility, availability, trust, and privacy of the two PIP methods. Findings included:

- Embedded user interfaces were more usable than portable interfaces (Interaction—efficiency, effectiveness, user satisfaction, customization, and transparency)
- Portable interfaces had more utility than embedded interfaces (Interaction—efficiency, effectiveness, and user satisfaction; Impact—utility)
- Embedded interfaces were more reliable, hence more available (Adoption—availability)
- No differences in trust between the two PIPs were found (Trust—privacy)
- Privacy was related not just to the PIP but to the device being customized (Trust—privacy)

TABLE 1
Framework of Ubiquitous Computing Evaluation Areas (UEAs).

UEA	Metric	Conceptual measures
Attention	Focus	Number of times a user must change focus due to technology; number of displays/actions users need to accomplish, or to check progress, of an interaction; number of events not noticed by a user in acceptable times
	Overhead	Percent of time user spends switching among foci; workload imposed on user attributable to focus
Adoption	Rate	New users/unit of time; user rationale for using the application over an alternative; technology usage statistics
	Value	Changes in productivity; perceived cost/benefit; continuity for user; amount of user sacrifice
	Cost	User willingness to purchase technology; typical time spent setting up and maintaining the technology
	Availability	Number of actual users from each target user group; technology supply source; categories of users in post-deployment
	Flexibility	Number of tasks user can accomplish that were not originally envisioned; user ability to modify as improvements and features are added
Trust	Privacy	Type of information user has to divulge to obtain value from application; availability of the user's information to other users of the system or third parties
	Awareness	Ease of coordination with others in multi-user application; number of collisions with activities of others; user understanding about how recorded data is used; user understanding inferences that can be drawn about him or her by the application
	Control	Ability for users to manage how and by whom their data is used; types of recourse available to user in the event that his or her data is misused
Conceptual Models	Predictability of application behavior	Degree of match between user model and behavior of application
	Awareness of application capabilities	Degree of match between user's model and actual functionality of the application; degree of match between user's understanding of his or her responsibilities, system responsibilities, and the actual situation; degree to which user understands the application's boundary
	Vocabulary awareness	Degree of match between user's model and the syntax used by the application
Interaction	Effectiveness	Percentage of task completion
	Efficiency	Time to complete a task
	User satisfaction	User rating of performing the task
	Distraction	Time taken from the primary task; degradation of performance in primary task; level of user frustration
	Interaction transparency	Effectiveness comparisons on different sets of I/O devices
	Scalability	Effectiveness of interactions with large numbers of entities or users
	Collaborative interaction	Number of conflicts; percentage of conflicts resolved by the application; user feelings about conflicts and how they are resolved; user ability to recover from conflicts
Invisibility	Intelligibility	User's understanding of the system explanation
	Control	Effectiveness of interactions provided for user control of system initiative
	Accuracy	Match between the system's contextual model and the actual situation; appropriateness of action; match between the system action and the action the user would have requested
	Customization	Time to explicitly enter personalization information; time for the system to learn and adapt to the user's preferences
Impact and Side Effects	Utility	Changes in productivity or performance; changes in output quality
	Behavior changes	Type, frequency, and duration; willingness to modify behavior or tasks to use application; comfort ratings of wearable system components
	Social acceptance	Requirements placed on user outside of social norms; aesthetic ratings of system components
	Environment change	Type, frequency and duration; user's willingness to modify his environment to accommodate system
Appeal	Fun	Enjoyment level when using the application; level of anticipation prior to using the application; sense of loss when the application is unavailable
	Aesthetics	Ratings of application look and feel
	Status	Pride in using and owning the application; peer pressure felt to use or own the application
Application Robustness	Robustness	Percentage of transient faults that were invisible to user
	Performance speed	Measures of time from user interaction to feedback for user
	Volatility	Measures of interruptions based on dynamic set of users, hardware, or software

APPLICATIONS

UBIQUITOUS COMPUTING EVALUATION AREAS

For each UEA in Table 1, we offer a definition, discussion, and some sample metrics here.

Attention

Attention is "increased awareness directed at a particular event or action to select it for increased processing."[1] The idea of *attention* has been explored in depth in desktop computing. For example, Sara Bly and Jarrett Rosenberg investigated tiled versus overlapping windows to determine which was more efficient.[2] Early studies also investigated ways to "grab" the user's attention, and derived guidelines for highlighting and color.[3,4] In these and other studies, desktop system designers learned to manage many attention issues.

However, attention is a more complex issue in ubiquitous computing, because users handle other physical or mental tasks while interacting with pervasive devices, which they might use in a variety of situations.

Attention metrics include focus and overhead. Focus refers to the point of user attention. Designers must carefully investigate affects on user focus to seamlessly blend applications into the environment or to provide peripheral awareness. Overhead refers to the wasted time introduced by the technology. For example, you could measure the time users spend switching between the technology and other foci, compare the time it takes to complete the task with and without the technology, or ask users about the affect on task efficiency.

Adoption

Adoption refers to willingness to use an application and rates of use. Jonathan Grudin discussed adopting computer-supported cooperative work applications, noting how a critical mass is needed to successfully deploy collaboration technologies.[5]

Geoffrey Moore discussed how technology is adopted, pointing out the value of observable referents to determine the utility of a technology before adopting it.[6] Larry Downes and Chunka Mui gave 12 rules for designing radical technologies.[7] Two are applicable measures for adoption: user continuity and user sacrifice. Electronic shopping is an example of user continuity. To a catalog shopper, electronic shopping is a reasonable extension. The user can get the same service more quickly using the Web than by mail order. User sacrifice refers to the services or value that users actually get compared to what they really wanted.

Another factor driving adoption is that applications are frequently used for unanticipated purposes; systems flexible enough to let this happen might be more readily adopted. So evaluators should recognize these unanticipated uses, particularly as the technology matures. Flexible systems can also support user modification with changing needs. Categories and sample metrics for adoption include rate, value, cost, availability, and flexibility.

Trust

Trust is user belief that a system will use the personal data it collects appropriately and not to cause harm. It entails issues of awareness, privacy, and control. Paul Dourish and Victoria Bellotti define awareness as "an understanding of the activities of others, which provides a context for your own activities."[8] For example, to understand the risk entailed in using an application, users must understand inferences others can make about them. However, awareness can be antithetical to privacy because it is increased by distributing information about system users. For example, in a tour guide, there is value in knowing what venues other users found interesting. However, having information about your visits and whereabouts saved might be disconcerting if privacy is important. Giving the user control over when, how, and by whom their information can be used might increase user trust.

Conceptual Models

As defined by Kevin Mullet, a conceptual model provides a basis for understanding an interactive device or program. It describes the various components and explains what they do and how they work together to accomplish tasks. Understanding the conceptual model lets users anticipate the behavior of the application and infer correct ways of doing things and diagnosing problems.

Although designers and developers have different conceptual models for the same application, for our purposes, we are interested in the user's conceptual model. For example, analogies or metaphors, such as the desktop metaphor, offer affordances that support conceptual models of activities. However, distributed ubiquitous computing can make it challenging for users to build unified models of behaviors and interactions. For example, how do users know when they are in a smart room, and how will they know how to interact with such a room?

Interaction

Interaction is how users and systems work together. Desktop usability evaluations have used the metrics of effectiveness, efficiency, and user satisfaction. While these three metrics are also applicable in ubiquitous computing, evaluators must consider differences between desktop and ubiquitous computing. Steven Shafer, Barry Brummit, and J.J. Cadiz suggest these differences:[9]

- Interactions in ubiquitous computing can be physically embedded.
- Input and output devices can be dynamic rather than static.

Confusion about how to use the portable devices could also be measured using Conceptual Model metrics.

ECLASSROOM

eClassroom, used since 1997, has been extensively evaluated.[7] It gives students remote access to electronic notes from previous classes, including through whiteboard, audio, and video. The evaluation

- If multiple devices are used, they require multiple focal points.
- There can be multiple users simultaneously.

Additional measures are needed to evaluate these aspects of ubiquitous computing. Guidelines have been developed for designing graphical user interactions based on mouse and keyboard input and a single display as output. The pervasive computing community needs studies and evaluations for distributed, multimodal interactions in distributed computing environments.

Invisibility

Invisibility refers to the integration of a system into the user environment. "Smart"—context-aware—applications make inferences about user activities, goals, emotional states, and even the social situation, and might attempt to act on the user's behalf. If the system has sensed and interpreted the context correctly, it can save time and reduce user workloads. However, if the system misjudges the situation, the user might have to interdict its actions, potentially resulting in wasted time, embarrassment, and even danger. Victoria Bellotti and Keith Edwards maintain that context-aware systems must be intelligible and accountable.[10] Systems that sense and use context need to explain their understanding to users who can then judge accuracy. Smart systems might also let users customize responses based on personal preferences by explicit input or by letting systems learn and adapt over a series of interactions.

Impact and Side Effects

Another area includes impact and side effects. First, technology must offer utility for the user. What contribution does it make to the user that was previously unavailable? Even well-designed, functional technology does not always succeed, and a system's unintended consequences can block acceptance. For example, one of us evaluated a system that was rejected because of the change in role that the system imposed on the users by asking for information they did not have.

Social acceptance also plays a role in whether technology is used. David Curtis and his colleagues noted that users of the Boeing wiring system (shown in the Oct.–Dec. 2002 Applications department) were not comfortable being seen by others while they were wearing the "socially unacceptable" goggles needed to use the system.[11]

Appeal

Although user satisfaction is already a component of the interaction UEA, appeal goes beyond this, referring to how attractive the application is to users. Does the application add to the user's enjoyment and quality of life? The Rememberer, which lets users capture museum visits, is an example of an application that should be evaluated for appeal, so metrics such as fun, aesthetics, and status should be considered.[12]

Application Robustness

We are primarily concerned with user-centered measures in this evaluation framework rather than with performance measures. However, performance affects a user's ratings and perceptions of the system. Therefore, we must include system performance metrics so we can interpret other user-centered measures in this context. The metrics here include measures of software robustness and of hardware operation, interoperability, and configuration.[13]

REFERENCES

1. R. Proctor and K. Vu, "Human Information Processing: An Overview for Human-Computer Interaction," *The Human-Computer Interaction Handbook*, Lawrence Erlbaum Assoc., 2003, pp. 35–51.
2. S.A. Bly and J.K. Rosenberg, "A Comparison of Tiles and Overlapping Windows," *Proc. CHI Human Factors in Computing Sys. Conf.*, ACM Press, 1986, pp. 101–106.
3. S.L. Smith and J.N. Mosier, *Guidelines for Designing User Interface Software*, tech. report ESD-TR-86-278, Hanscom Air Force Base, USAF Electronic Systems Division, 1986.
4. W.O. Galitz, *Handbook of Screen Format Design*, QED Information Sciences, 1985.
5. J. Grudin, "Why CSCW Applications Fail," *Proc. Computer-Supported Cooperative Work Conf.*, ACM Press, 1988, pp. 85–93.
6. G. Moore, *Crossing the Chasm*, Harper Business, 1991.
7. L. Downes and C. Mui, *Unleashing the Killer App*, Harvard Business School Press, 1998.
8. P. Dourish and V. Bellotti, "Awareness and Coordination in Shared Workspaces," *Proc. Computer-Supported Cooperative Work Conf.*, ACM Press, 1992, pp. 107–114.
9. S. Shafer, B. Brummitt, and J.J. Cadiz, "Interaction Issues in Context-Aware Intelligent Environments," *Human-Computer Interaction*, vol. 16, nos. 2–4, 2001, pp. 363–378.
10. V. Bellotti and K. Edwards, "Intelligibility and Accountibility: Human Considerations in Context-Aware Systems," *Human-Computer Interaction*, vol. 16, nos. 2–4, 2001, pp. 193–212.
11. D. Curtis et al., "Several Devils in the Details: Making an AR Application Work in the Airplane Factory," *Proc. Augmented Reality: Placing Artificial Objects in Real Scenes* (IWAR'98), AK Peters, 1999, p. 48.
12. M. Fleck et al., "Rememberer: A Tool for Capturing Museum Visits," *Proc. 4th Int'l Conf. Ubiquitous Computing*, Springer-Verlag, 2002, pp. 48–55.
13. T. Kindberg and A. Fox, "System Software for Ubiquitous Computing," *IEEE Pervasive Computing*, vol. 1, no. 1, 2002, pp. 70–81.

encompassed 98 academic courses taught over 13 semesters at two universities. The evaluation relied on qualitative and quantitative measures gathered from instructor experiences, questionnaires, access logs, and comparisons to traditional classrooms. Findings included the following:

- Access to electronic notes increased

after the initial exam (Impact—utility, behavior changes; Adoption—rate).

- Attendance was not significantly affected (Impact—behavior changes).
- Instructors used the notes to review other lectures and student presentations (Adoption—flexibility).
- Electronic notes did not significantly affect exam performance (Impact—utility).
- Access logs showed students used both forward and backward jumps when accessing the notes, necessitating support for both (Interaction—effectiveness, user satisfaction).
- Students took fewer notes by hand (Impact—utility).
- Instructors wanted to edit portions of the lectures (Conceptual Models—awareness of capabilities).

These researchers also presented an extensive evaluation of eClassroom and investigated both direct and indirect stakeholders—students and professors—to determine how well it worked. Their evaluation measures fit into our framework well, increasing our confidence in it as an initial reporting structure.

The framework we have developed will be refined as we and others use it. Also, we will need to answer interesting questions about relationships among the UEAs; some might be easier to evaluate than others, so indirect measures could make evaluations more feasible. P

ACKNOWLEDGMENTS

We wish to acknowledge the help of many colleagues, too numerous to mention here, in the preparation of this framework.

REFERENCES

1. J. Mankoff et al. “Heuristic Evaluation of Ambient Displays,” *Proc. Conf. Human Factors in Computing Sys.*, ACM Press, 2003, pp. 169–176.
2. A. Jameson, “Adaptive Interfaces and Agents,” *The Human-Computer Interaction Handbook*, Lawrence Erlbaum Assoc., 2003, pp. 316–318.
3. V. Bellotti et al., “Making Sense of Sensing Systems: Five Questions for Designers and Researchers,” *Proc. Conf. Human Factors in Computing Sys.*, ACM Press, 2002, pp. 415–422.
4. B. Friedman, P.H. Kahn Jr., and A. Borning, *Value Sensitive Design: Theory and Methods*, tech. report 02-12-01, Univ. Washington, Dec. 2001.
5. J. Burrell et al., “Context-Aware Computing: A Test Case.” *Proc. 4th Int'l Conf. Ubiquitous Computing*, Springer-Verlag, 2002, pp. 1–15.
6. J. Trevor, D. Hilbert, and B. Schilit, “Issues in Personalizing Shared Ubiquitous Devices,” *Proc. 4th Int'l Conf. Ubiquitous Computing*, Springer, 2002, pp. 56–71.
7. J.A. Brotherton and G.A. Abowd, “Lessons Learned from eClass: Assessing Automated Capture and Access in the Classroom,” to be published in *ACM Trans. Computer-Human Interaction*, 2004.

Jean Scholtz is a computer science researcher at the US National Institute of Standards and Technology, where she directs a number of programs involving evaluation of interaction with intelligent software systems. Contact her at jean.scholtz@nist.gov.

Sunny Conolvo's biography appears on page 29.