Ubicomp Systems at 20: Progress, Opportunities, and Challenges

This retrospective on 20 years of ubiquitous computing research identifies opportunities for leveraging utility computing and the Internet of Things to grow the ubicomp infrastructure, and discusses remaining challenges to taking ubicomp systems to where they indeed become ubiquitous.

Ramón Cáceres
AT&T Labs

Adrian Friday
Lancaster University

Twenty years ago, Mark Weiser set forth his compelling vision of ubiquitous computing (ubicomp),[1] giving rise to a rich, multifaceted research area. This community has spawned several major conferences, journals, and magazines, including *IEEE Pervasive Computing* 10 years ago. At this milestone, we take stock of where ubicomp systems research has journeyed and postulate the major challenges for moving forward.

Reflecting on the challenges facing ubicomp systems in order for them to become truly ubiquitous, we use work taken from the research literature as exemplars of what has been achieved and emphasize the areas where there aren't yet sufficient solutions. We then present some opportunities for leveraging emerging industrial trends that could greatly hasten wider ubicomp deployment and discuss two remaining challenges relating to the payment and management of ubicomp services.

The Original Vision

Let's first review the early notions of ubicomp—those that inspired our community and that we're ostensibly trying to realize. For the sake of brevity, we outline two key visions—calm computing and ambient intelligence—although there are innumerable projects, each posing their own forecasts of technological futures with varying degrees of specificity and of alignment with these two.

Weiser's Calm, Integrated World

In 1991, Weiser envisioned a world where interaction occurs with everyday—but computationally augmented—artifacts through natural interactions, our senses, and the spoken word.[1] As Weiser described, "Sal's" alarm clock senses when to interact with her to trigger the brewing of coffee; augmented-reality window displays add first to her perception of her neighborhood's movements and then to the activity of her remote colleagues. It's a calm world where information seamlessly moves in and out of attention as automation gives way to human interaction.

Weiser also envisioned the digital and physical being tightly integrated: Sal locates a missing manual by virtue of its embedded tag, and her "foreview mirror" helps her transit to work and park more efficiently. Particularly radical at the time, Sal accesses not one computer but many, and these work together as a single, seamless entity. Sal customizes her environment using computational "tabs" that are intentionally shared—she has a view of her colleague Joe's tab that she can bring into focus if need be. The environment is also programmable—Sal

Published by the IEEE CS ■ 1536-1268/12/$31.00 © 2012 IEEE

programs the "telltale" by the door to alert her when fresh coffee is brewed.

It's a future where computation augments the senses. Furthermore, the interconnectedness of information, the environment, and devices enables them to work in concert to support everyday life—for convenience and enhanced productivity.

The Ambient Home of 2020

Similarly, stemming from an attempt to predict the user-friendly home of 2020, the Information Society Technologies Advisory Group's (ISTAG's) 2001 "Scenarios for Ambient Intelligence in 2010" envisioned ambient intelligence (AmI), in which devices support ubiquitous information, communication, and entertainment.[2] AmI emphasizes similar views of efficiency and user empowerment: "Maria" is identified by "the ambient" and walks effortlessly through airport security.

AmI implies a similar level of digital-physical integration to calm computing but has a more concentrated idea of proactive services "in the ether" working on our behalf—Carmen's "agents" negotiate a rate and pay for her taxi automatically; Dimitrios's "Digital Me" handles incoming calls when he's busy.

Why Aren't We There Yet?

Despite the passage of 20 years, and it's been quite a 20 years—witnessing the birth of the World Wide Web; global mobile telephony; smartphones; and unimaginable increases in available computation, storage, and communication, especially on personal devices—we have yet to achieve these ubicomp visions. Why is this?

The visions do important work in galvanizing the research community but aren't intended to be templates for ubicomp. They're works in progress that are rightly being called into question. Yvonne Rogers challenges the emphasis on calmness and proactive environments that remove the need for humans to think for themselves, preferring a more human-centric viewpoint:[3]

> *We should consider how ubicomp technologies can be designed to augment the human intellect so that people can perform ever greater feats, extending their ability to learn, make decisions, reason, create, solve complex problems and generate innovative ideas.*

Rogers also points out the contribution to science and learning that could be made by ubicomp's extended sensorial purview.

William Gaver explores "ludic behavior," introducing playful systems into our daily lives, encouraging us to re-experience our environments in new ways. In his work, he considers the "aesthetic, utilitarian, and practical issues" involved in a particular ludic system's creation, and details what it can offer.[4]

Genevieve Bell and Paul Dourish question whether ubicomp is always destined to be framed as an artifact of an unachievable "near future" and point out exemplars of technologies, such as mobile communications, that have reached some degree of ubiquity by the 21st century.[5] They also direct us to question the uniformity of these visions—pointing out Scott Manwaring's observation that real-world infrastructures, however uniform they might appear, are often messy underneath and subject to ongoing work by professionals.

Despite the passage of 20 years, we have yet to achieve these ubicomp visions. Why is this?

State of the Nation

There are nascent building blocks of the ubicomp visions that hint at the more tractable problems and help us identify the many challenges that stubbornly remain. We briefly review areas that are important to constructing more general ubicomp systems.

Ubiquitous Data

In the current social and political climate, we can't imagine any kind of technology being relied upon as a sufficient guarantor to let us pass through airport security without close scrutiny. Could we ever trust a ubicomp environment to do this? The very notion of a ubiquitous, global biometric database, or indeed, even an omnipresent ambience, fills many with an Orwellian dystopian dread. But this scenario raises several important questions about the data in ubicomp, its trustworthiness, and access to it.

When can we infer with certainty? One reason we might not trust ubicomp systems to recognize our identity is that sensed interactions are imprecise observations of the world, often taken from multiple sensors and at varying points in time. Ubicomp environments must weigh this evidence and make a judgment of when and how to react.

The severity and importance of the outcome is certainly application and context dependent. The Context Toolkit importantly abstracts sensing from the application,[6] and PersonisAD separates rules from application code, opening them up to user scrutiny.[7] There's ample scope for better programming models that help us reason more cleanly about uncertainty, leading us to more clearly articulated reactions from ubicomp environments.[8]

Where Is ubiquitous data located? Ubiquitous access to data raises the important question of where ubiquitous data lives. Certainly, a global ubiquitous data store isn't practical for capacity, bandwidth, latency, and availability reasons—but neither is it desirable.

For many environments, such as rooms, homes, companies, and hospitals, the demands for security and privacy require enforcing conventional or physical boundaries. Tim Kindberg and Armando Fox refer to this need as the "boundary principle" in which "ubicomp system designers should divide the ubicomp world into environments with boundaries that demarcate their content."[9] As far as we know, there has been little attempt to address the issue of data location during the last decade.

We have yet to resolve what the environment tracks and shares and what remains private.

How long should data persist? What does the environment know about us? What should it know and what should we trust it with? How long should data be retained? What is transient and what should persist? Can we delete it, and can it be forgotten?

To enact the foreview mirror, Sal needs access to information about open spaces in the parking lot. Intuitively, it seems this should be public information. However, if the data is more intrusive and can identify particular vehicles or people, then suddenly the uses to which it can be put are more insidious, and the need for tighter control more exigent.

In typical ubicomp systems, data is closed to the experimenters who deploy the system or experiment, and the choice as to what the system keeps or forgets is often underarticulated, so this issue isn't addressed. To enable open scientific use of ubicomp sensing, or even ubicomp crime-scene forensics, the issue of data persistence and access control comes to the fore. This is even more challenging in ubicomp, where notions of identity are often weak.

How do we express our privacy wishes? We have yet to resolve the issue of what the environment tracks and shares and what remains private to the individual. Marc Langheinrich established some principles for guiding privacy-aware design.[10] John Canny and Yitao Duan build on these to establish the prototype of a trustworthy smart environment with audio-video capture that respects an individual's "data discretion" by encoding access rights in the sensor data.[11]

More recently, John Krumm focuses on preserving identity and home-location privacy from inference attacks based on large-scale observations of past journeys.[12] Krumm nicely highlights how, even with minimal information, our routines betray us, raising key questions for systems philosophy about when to release personal data, to whom, and when data can be regarded as "ours" in the first place.

Interestingly, in all this discussion about remote access, it's worth mentioning the converse—data that can only be observed if the individual or device is present in a given environment. This has been used to authenticate an individual's presence, as in Kindberg's formative work on context authentication. On a smaller scale, Rene Mayrhofer and Hans Gellersen elegantly use shared observations of context to securely link two devices that are shaken together.[13]

Understanding the World

The rich simpatico between the physical and digital that ubicomp exhibits implies comprehensive knowledge of users' whereabouts and activity. Where does this computational understanding come from and how does it keep pace with changes?

Microsoft's EasyLiving project required a detailed 3D model of the room, enabling content to be sensibly placed to suit its users. In sentient computing, actions are triggered based on spatial context and relations, such as the view and proximity of people and artifacts. This necessitates fine-grained indoor positioning—the availability and calibration of which is still regarded as challenging and expensive.[14] One interesting and recent approach is to explore "open data," where models are built collaboratively and refined incrementally by multiple users. Jun-Geun Park and his colleagues applied this open crowdsourcing-based approach to build a location system based on Wi-Fi fingerprinting.[15]

Over the last decade, we've seen various applications emerge that relax this need for precise models of space and the environment. Evan Wellbourne explores worn passive RFID tags and an array of situated readers for context-aware applications,[16] such as active maps and locating artifacts and people. Mik Lamming and Denis Bohm detect relative co-presence, rather than absolute position, by tagging artifacts with simple IR beacons that record the identities of other artifacts they encounter.[17] This enables simple contextual applications based on patterns and changes of association.

An alternative to infrastructure sensing is on-body or wearable sensing. Wearable accelerometers have been used with machine-learning algorithms to determine finer points of context and behavior. Julie Keintz and her colleagues have shown how you can apply context sensing to support children with autism and their caregivers.[18] Don Patterson used a combination of body-worn sensors with location tracking and mobile communication to helps individuals with mild cognitive disabilities select appropriate methods of transportation.[19] These exemplify a new class of ubicomp as utilitarian, educational, and assistive technologies, rather than actors of convenience or efficiency. Yet there are considerable challenges to improving the classifiers' generality and portability and reducing the training datasets' size.[20]

Outdoors, over the last decade, location has become easy to track to within a few tens of meters because of GPS, Assisted GPS, Wi-Fi fingerprinting, and cellular-antenna fingerprinting capabilities available on many mobile phone handsets. These widely available platforms have ensured that location and its protection and sharing remain an important research topic.[21] The smartphone has proven itself to be a versatile tool suitable for a wide range of ubicomp applications, including recognizing aspects of location and emotional state and communicating this automatically with the user's online social network.[22]

Systems for a Changing World

The work we've surveyed thus far can be considered, for the most part, to be self-contained: the system *is* the application. Often in the interest of expediency, researchers must build with the hardware and software components available to them, which is entirely reasonable. However, this leads to unique and complex systems that are hard to transfer to new environments, owing to the "magic" required to configure and use them,[23] and their tight binding to their particular building blocks and environment.

There have been platform-oriented projects, such as the University of Illinois at Urbana-Champaign's Gaia and MIT's Oxygen, which have sought a principled separation between application and system.[24] Such projects have enabled researchers to write smart-room applications and deploy them in multiple locations (UIUC was at one point linked internationally to the Tokuda Lab at the Keio University in Japan).

A key challenge of creating ubicomp systems that can be deployed in more than one environment and for substantial time periods is the degree of change or volatility experienced.[25]

Volatility—the changing environment. Not only does the world change (and thus so should our computational understanding of it, as alluded to earlier), but so do the set of users, devices, and software components in an environment—far more frequently in ubicomp systems than in conventional distributed systems. This implies the creation and destruction of associations—logical communication relationships—between software components resident on the devices. It also implies failures where communication is no longer possible between them.

Change also brings opportunities, as new resources with different capabilities come into play.

But change also brings opportunities, as new resources with different capabilities come into play. The system must be designed to incorporate change and failure within normal operating parameters and gracefully adapt or degrade appropriately.[24] HP's Cooltown took a human-centric approach based on Web technologies—artifacts tagged with URIs were resolved when users triggered some action. The user simply tried again when an action didn't occur.

Adaptation—responding to volatility. Several research projects have focused on adaptation—the discovery and reconfiguration of services and their associations to maintain smart environments.

Gaia provided adaptation at the communication, application-framework, and resource-composition levels. Oxygen used a multiagent-based system that automatically assembled compatible services from components and reassembled them after failures. Stanford's Interactive Workspaces project followed a data-driven approach, where parts of the smart environment communicated via an event heap rather than directly. An attraction of this model is the ability to rewrite events to introduce new elements and applications into the environment. Users can actively do this using a "patch panel" application.

Adaptation is also triggered when the context changes in context-aware systems. Ubicomp environments are almost always shared—yet adapting to groups of users and devices continues to be systematically underexplored in the literature.

Evolution—adapting to the unexpected. The goal should be "open loop" adaptation—that is, a ubicomp environment should be able to cope with users, devices, and software it hasn't seen before. Evolving to incorporate unanticipated elements at runtime (and in ubicomp, it's always runtime!) is a fascinating unsolved challenge.

We might find elements we can build upon. An interesting paradigm found in delay-tolerant networks is asynchronous processing—communication proceeds opportunistically when the resources are mutually available. Similarly, the Bayou system cleverly exploits serendipitous communication when in range to resolve a "shared state" in "anti-entropy sessions."[26] Aura's cyberforaging lets mobile users exploit proximate compute and storage resources to reduce battery demand. And SpeakEasy transfers mobile code to enable recombinant computing so that users can walk up and interact with devices that they've not hitherto encountered.[27] Interestingly, Smart Furniture assembles ubicomp environments on the fly from peer-smart furniture components, rather than relying on capabilities embedded into a smart building fabric.[28]

Programming Ubicomp

It's not currently possible to write a portable ubicomp application. There's neither a common runtime on which to share and build nor any ubicomp OS vendors. Ten years ago, Kindberg and Fox posited the question: What does it mean to program hello world (the archetypal first program) for ubicomp?[9] Superficially at least, we're no closer now to knowing the answer than we were then.

This doesn't mean that nascent ubicomp environments aren't programmable; they're often inherently composed of many computational elements, which can each be individually programmed. More mature environments (such as Gaia, Oxygen, iROS, and Equip) have programming interfaces that let high-level application-like behaviors be created from system components or that can trigger certain actions when events occur (sentient computing, Context Toolkit, PersonisAD), so that components written by experts can be assembled by those who are less technical. Ultimately, this direction leads us toward tools to enable programming by end users (as in CAMP[29]).

Reducing the Infrastructure Burden

By its nature, ubicomp requires low-latency interaction with users and environments. At least part of a ubicomp application must be tightly bound to the infrastructure near where the interaction is happening. This requirement for local infrastructure is a barrier to wide-scale adoption.

There have been several clever attempts to reduce the burden of or need for rolling out dedicated infrastructure for enabling a ubiquitous application. Jennifer Mankoff demonstrates how to use very-low-fidelity sensing (shopping receipts, for example) to improve nutritional awareness. Shwetak Patel and colleagues have also shown several cunning examples in which machine learning is applied to "single-point sensing" of existing domestic infrastructures—such as mains wiring, plumbing, and heating, ventilation, and air conditioning (HVAC) ducting—to reveal behavior throughout the home without the need for extra sensors.[30]

You can also exploit preexisting infrastructures. RF fingerprints can create location systems such as RightSPOT.[31] Mixed-reality games exploit GPS urban-canyon effects to enhance gameplay, and "seamful" games exploit the boundaries of otherwise invisible infrastructure such as Wi-Fi access points as landmarks for games played on city streets.[32] People can even be invited to create their own infrastructures. Enrico Costanza and his colleagues create downloadable interfaces that consist of feducials printed on paper that are recognized using commodity webcams (now widely deployed and integrated into most laptops) and vision techniques.[33]

Furthermore, 3D printing technology is becoming a commodity within the budgetary reach of at least schools and businesses if not homes. Printable electronics is an extremely active research field that's making great strides. Downloadable sensors, RFID tags, and displays will likely follow in the next decade or so.

Very-low-tech sensing is possible by asking users to contribute data directly or indirectly. Analyzing this data (photographs, for example) often requires human perception. An increasing number of projects draw on Amazon's Mechanical Turk to reduce the burden of digitizing and classifying such input. This access to "human processors" is exciting for research projects but doesn't scale ubiquitously and raises the more general question of who pays for ubicomp services (we return to this later).

> A key problem with wider ubicomp adoption is the tight coupling with particular embedded infrastructures.

Energy Impact and Awareness

A major challenge going forward will be how to address ubicomp's energy impact as we enter an era in which we'll have to justify the energy used—and "always on" will no longer be deemed acceptable. Designing infrastructures to be "mostly off" raises profound challenges in terms of how we structure and partition ubicomp systems, especially ones presumed ready to interact ubiquitously and always provide access to our data.

Opportunities to Grow the Infrastructure

Although a long-lived and large-scale ubicomp infrastructure has yet to materialize, we see opportunities to grow the needed infrastructure in technology areas not directly motivated by ubicomp.

Utility Computing in the Cloud

Utility computing can help realize the ubicomp vision by providing large-scale and long-lived storage and processing resources for personal ubicomp applications. The notion of a utility that makes computing resources available to the public, analogous to an electric or telecommunications utility, goes back at least to the 1960s and the Multics project.[34] This notion has become a reality in recent years with the rise of cloud computing services, such as Amazon's Elastic Compute Cloud, which offers pay-as-you-use resources in the form of virtual machines.

Utility computing offerings to date have been mostly aimed at enterprises, but we believe that offerings aimed at individual consumers will proliferate. Cloud computing services for individuals make natural companions to personal mobile devices and future ubicomp applications. An early example of these services is Apple's MobileMe, which is evolving into iCloud. MobileMe and iCloud provide storage for common types of personal data (such as music, photos, and calendars), and they can synchronize this data among mobile devices and personal computers.

One limitation of MobileMe, iCloud, and similar services is that the set of data types and applications they support are restricted to those approved by a single service provider. This limitation could be avoided by giving individuals control over their own virtual machines hosted in the cloud. Such Virtual Individual Servers (VISs) offer individuals a number of advantages, including improved flexibility, portability, longevity, and privacy.[35] For example, individuals could choose which software packages to install on their virtual machines, ensure the use of open and portable data formats, and control who has access to what data. VISs would thus help address many of the data location, persistence, and privacy issues discussed earlier.

While utility computing in the cloud can provide important back-end resources for ubicomp applications, it can't provide all of the necessary infrastructure. In particular, ubicomp also requires widespread infrastructure that's local to its users—for example, sensors and actuators in the users' immediate environment. To grow this local infrastructure, we'll need to develop more device-centric technologies such as the Internet of Things (IoT).

Internet of Things

A key problem with wider ubicomp adoption is the tight coupling with particular embedded infrastructures. As discussed, this has led researchers to become increasingly ingenious in considering how they might exploit existing infrastructures for other purposes—such as the cell phone network, power lines, and even smartphones and users themselves.

The world we inhabit is getting smarter all the time of its own accord, courtesy of the government and industry. Buildings incorporate sensors and actuators for HVAC control, motion-triggered lighting, intruder detection, fault detection, and so on. Even homes increasingly have security and heating systems with room-level sensors that detect motion and the opening and closing of windows and doors. Some commercial appliances (including elevators and copiers) can already "call home" for engineering support in the event of failure. Our cars are becoming densely sensed, not only to monitor the car's operation and the environment it encounters but also increasingly for passenger safety and comfort. There are also sensors in our civic infrastructure and roads. All this increasing "smartness" is surely an opportunity to the ubicomp community, provided that this infrastructure is open to us.

Kevin Ashton, co-founder of AutoID Labs, first used the term IoT in 1999, envisioning it as a supply chain with RFID-tagged or barcoded items (things) offering greater efficiency and accountability to businesses. The AutoID consortium continues to investigate tags with embedded sensors and actuators. As Ashton wrote in *RFID Journal* (22 June 2009):

> *If we had computers that knew everything there was to know about things—using data they gathered without any help from us—we would be able to track and count everything, and greatly reduce waste, loss and cost. We would know when things needed replacing, repairing or recalling, and whether they were fresh or past their best.*

While this vision might seem familiar to ubicomp researchers, IoT has primarily been about automated, thing-to-thing interaction. However, IoT is evolving and increasingly implies openness—it's gaining momentum and is now the subject of international conferences. The UK government just invested £500,000 to conduct studies on forming an open application and services ecosystem with "open availability of data from 'things' and 'harmonized access […] across organizations.'"[36]

There is an alternative definition of IoT: websites such as pachube.com make feeds from sensors available using regular Web protocols. Anyone can deploy a sensor, even using hobbyist electronics. This has already proven to be extremely valuable. Following the devastating Tohoku earthquake and the ongoing disaster at the Fukushima Dai-ichi nuclear power plant, a volunteer effort called Tokyo Hackerspace initiated a program of workshops on how to build Internet-enabled Geiger counters whose deployment helped generate live radiation maps.

In short, ubicomp is low on deployed infrastructure, but potential infrastructure is growing—we just need to harness it! We don't advocate doing this clumsily, and we don't think that everything should be ubiquitously open to all—the potential for misuse for surveillance, control, and cyberterrorist purposes is real. Exploiting these opportunities will require new ways of opening up access to otherwise private or enclosed infrastructures, which is neither technically nor politically trivial.

> Two of the main issues are more economic than technical: Who will pay for ubicomp systems, and who will manage them?

Outlook for Large-Scale Deployment

Although there have been many successful research prototypes of ubicomp systems, the technology in such prototypes won't see wide adoption until several difficult issues are resolved. Two of the main issues are more economic than technical: Who will pay for ubicomp systems, and who will manage them?

Rolling out ubicomp systems on a large scale will require a great deal of industry involvement. Ubicomp might grow in a grass-roots fashion from the experiments put forth by the research community, but it's more likely that large investments in infrastructure will be required, much as they were needed to achieve ubiquitous telecommunications services. It's expensive to keep such infrastructures working, both because faulty components must be repaired or replaced, and because components must be upgraded as the underlying technology evolves.

The way these investments are repaid has important privacy implications. One economic model for online services is driven by advertising revenue—many online social networking services use this model. It's attractive to consumers because the resulting services are typically free. However, for the service provider, it creates an unavoidable conflict of interest between making money and protecting customer privacy. These conflicts are reflected in the terms of service of popular free online services, which typically grant the provider various rights to use the data contributed by the service's users.

A different model with better privacy properties is one where the consumer pays the provider for the resources used—the utility computing services based on virtual machines mentioned earlier use this model.[35] It removes the conflict of interest, as reflected in terms of service that don't grant providers any rights to user-contributed content. Of course, for this model to succeed, ubicomp systems and their applications must provide enough value to their users that payment will seem worthwhile. Consumers are willing to pay for services they consider valuable—such as mobile phone service.

Related to the question of who will pay is who will manage the ubicomp systems. Individual consumers have long proven unwilling or unable to manage their personal computer systems well. There's no reason to believe they'll prove any better at managing ubicomp systems. This observation supports having service providers do the managing on behalf of consumers.

Managed services would reduce complexity for the end user and help technology fade into the background. However, such services introduce their own tension between manageability and cost for the provider versus flexibility and control for the end user. As an example, who should own and control ubicomp devices such as home thermostats connected to a smart power grid: the end user or the service provider? What happens when the user's (local) desires to be comfortable conflict with the provider's (global) goals to conserve energy? It will take time to resolve these payment and management issues.

In the 20 years since Weiser articulated the ubicomp vision, a large and vibrant research community has grown around the ubicomp concept. Numerous successful prototypes have been built and evaluated, demonstrating the utility of many different aspects of ubicomp systems. In that same timeframe, digital technology has made great advances, enabling products and services that complement the ubicomp vision and have become part of the everyday lives of billions of people.

Arguably the most successful of these products is the mobile phone, which places increasing amounts of computing, sensing, and communication capabilities in the hands of a significant portion of the earth's population. However, despite this progress and the continuing opportunities for further advances, formidable challenges remain before we can realize many of the core ubicomp scenarios—such as calm computing and ambient intelligence—on any large scale.

We're hopeful that the research community, the technology industry, and society as a whole will combine to overcome these obstacles in the years to come.

REFERENCES

1. M. Weiser, "The Computer for the 21st Century," *Scientific Am.*, vol. 265, no. 3, 1991, pp. 66–75.
2. K. Ducatel et al., *Scenarios for Ambient Intelligence in 2010*, tech. report, Comisión Europea (office for official publications of the European Communities), 2001; ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf.
3. Y. Rogers, "Moving on from Weiser's Vision of Calm Computing: Engaging Ubicomp Experiences," *UbiComp 2006: Ubiquitous Computing*, vol. 4206, Springer, 2006, pp. 404–421.
4. W. Gaver, "The Video Window: My Life with a Ludic System," *Personal and Ubiquitous Computing*, vol. 10, nos. 2–3, 2006, pp. 60–65.
5. G. Bell and P. Dourish, "Yesterday's Tomorrows: Notes on Ubiquitous Computing's Dominant Vision," *Personal and Ubiquitous Computing*, vol. 11, no. 2, 2007, pp. 133–143.
6. D. Salber, A. Dey, and G. Abowd, "The Context Toolkit: Aiding the Development of Context-Enabled Applications," *Proc. SIGCHI Conf. Human Factors in Computing Systems: The CHI Is the Limit*, ACM Press, 1999, pp. 434–441.
7. M. Assad et al., "Personisad: Distributed, Active, Scrutable Model Framework for Context-Aware Services," *Proc. Pervasive Computing, 5th Int'l Conf.* (Pervasive 07), LNCS 4480, A. LaMarca, M. Langheinrich, and K.N. Truong, eds., Springer, 2007, pp. 55–72.
8. S. Dobson and P. Nixon, "More Principled Design of Pervasive Computing Systems," *Engineering Human-Computer Interaction and Interactive Systems*, LNCS 3425, R. Bastide, P. Palanque, and J. Roth, eds., Springer, 2005, pp. 103–111.
9. T. Kindberg and A. Fox, "System Software for Ubiquitous Computing," *IEEE Pervasive Computing*, vol. 1, no. 1, 2002, pp. 70–81.
10. M. Langheinrich, "Privacy by Design—Principles of Privacy-Aware Ubiquitous Systems," *Ubicomp*, LNCS 2201, G.D. Abowd, B. Brumitt, and S.A. Shafer, eds., Springer, 2001, pp. 273–291.
11. Y. Duan and J. Canny, "Protecting User Data in Ubiquitous Computing Environments:

Towards Trustworthy Environments," LNCS 3424, Springer, 2005, pp. 167–185.

12. J. Krumm, "Inference Attacks on Location Tracks," *Proc. Fifth Int'l Conf. Pervasive Computing* (Pervasive 07), LNCS 4480, A. LaMarca, M. Langheinrich, and K.N. Truong, eds., Springer, 2007, pp. 127–143.

13. R. Mayrhofer and H. Gellersen, "Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices," *IEEE Trans. Mobile Computing*, vol. 8, no. 6, 2009, pp. 792–806.

14. R. Harle and A. Hopper, "Building World Models by Ray-Tracing Within Ceiling-Mounted Positioning Systems," *Proc. 5th Int'l Conf. Ubiquitous Computing* (UbiComp 03), Springer, 2003, pp. 1–17.

15. J. Park et al., "Growing an Organic Indoor Location System," *Proc. 8th Int'l Conf. Mobile Systems, Applications, and Services*, ACM Press, 2010, pp. 271–284.

16. E. Welbourne et al., "Longitudinal Study of a Building-Scale FRID Ecosystem," *Proc. 7th Int'l Conf. Mobile Systems, Applications, and Services* (MobiSys 09), ACM Press, 2009, pp. 69–82.

17. M. Lamming and D. Bohm, "Specs: Another Approach to Human Context and Activity Sensing Research, Using Tiny Peer-to-Peer Wireless Computers," *UbiComp 2003: Ubiquitous Computing*, LNCS 2864, A. Dey, A. Schmidt, and J. McCarthy, eds., Springer, 2003, pp. 192–199.

18. J.A. Kientz et al., "Pervasive Computing and Autism: Assisting Caregivers of Children with Special Needs," *IEEE Pervasive Computing*, vol. 6, no. 1, 2007, pp. 28–35.

19. D.J. Patterson et al., "Opportunity Knocks: A System to Provide Cognitive Assistance with Transportation Services," *UbiComp 2004: Ubiquitous Computing*, LNCS 3205, Springer, 2004, pp. 433–50.

20. N.D. Lane et al., "Hapori: Context-Based Local Search for Mobile Phones Using Community Behavioral Modeling and Similarity," *Proc. 12th Int'l Conf. Ubiquitous Computing* (UbiComp 2010), ACM Press, 2010, pp. 109–118.

21. A.J.B. Brush, J. Krumm, and J. Scott, "Exploring End User Preferences for Location Obfuscation, Location-Based Services, and the Value of Location," *Proc. 12th Int'l Conf. Ubiquitous Computing* (UbiComp 2010), ACM Press, 2010, pp. 95–104.

22. E. Miluzzo et al., "Cenceme—Injecting Sensing Presence into Social Networking Applications," *Smart Sensing and Context*, vol. 4793, no. 5, 2007, pp. 1–28.

23. J. Barton and J. Pierce, "Quantifying Magic in Ubicomp Systems Scenarios," position paper, UbiSys Workshop at Ubicomp 2006, 2006; http://tinyurl.com/magicinubi.

24. G. Schiele, M. Handte, and C. Becker, "Pervasive Computing Middleware," *Handbook of Ambient Intelligence and Smart Environments*, Springer, 2010, pp. 201–227.

25. G. Coulouris et al., *Distributed Systems: Concepts and Design*, 5th ed., Addison Wesley, 2011.

26. D.B. Terry et al., "Managing Update Conflicts in Bayou, A Weakly Connected Replicated Storage System," *Proc. 15th ACM Symp. Operating Systems Principles* (SOSP 95), ACM Press, 1995, pp. 172–182.

27. W.K. Edwards et al., "Challenge: Recombinant Computing and the Speakeasy Approach," *Proc. 8th Ann. Int'l Conf. Mobile Computing and Networking* (MobiCom 02), ACM Press, 2002, pp. 279–286.

28. H. Tokuda et al., "Sf2: Smart Furniture for Creating Ubiquitous Applications," *Proc. Int'l Symp. Applications and the Internet Workshops* (SAINT 04), IEEE Press, 2004, pp. 423–429.

29. K.N. Truong, E.M. Huang, and G.D. Abowd, "Camp: A Magnetic Poetry Interface for End-User Programming of Capture Applications for the Home," *Ubicomp*, LNCS 3205, N. Davies, E.D. Mynatt, and I. Siio, eds., Springer, 2004, pp. 143–160.

30. S.N. Patel et al., "At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line," *Proc. 9th Int'l Conf. Ubiquitous Computing* (UbiComp 07), Springer, 2007, pp. 271–288.

31. J. Krumm, G. Cermak, and E. Horvitz, "Rightspot: A Novel Sense of Location for a Smart Personal Object," *Ubicomp*, LNCS 2864, A.K. Dey, A. Schmidt, and J.F. McCarthy, eds., Springer, 2003, pp. 36–43.

32. M. Chalmers et al., "Gaming on the Edge: Using Seams in Ubicomp Games," *Proc. 2005 ACM SIGCHI Int'l Conf. Advances in Computer Entertainment Technology*, ACM Press, 2005, pp. 306–309.

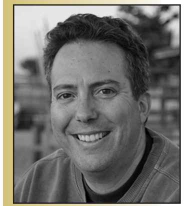
33. E. Costanza et al., "Ubicomp to the Masses: A Large-Scale Study of Two Tangible Interfaces for Download," *Proc. 12th Int'l Conf. Ubiquitous Computing* (UbiComp 2010), ACM Press, 2010, pp. 173–182.

34. F. Corbató and V.A. Vyssotsky, "Introduction and Overview of the Multics System," *Proc. Fall Joint Computer Conf.*, ACM Press, 1965, pp. 185–196.

35. R. Cáceres et al., "Virtual Individual Servers as Privacy-Preserving Proxies for Mobile Devices," *Proc. MobiHeld*, ACM Press, 2009, pp. 37–42.

36. J. Sully, "TSB's Internet of Things Convergence £4m Funding Call Announced," *Connect*, 26 July 2011; https://connect.innovateuk.org/web/creativektn/articles/-/blogs/4771613.

the AUTHORS



Ramón Cáceres is a lead member of the technical staff at AT&T Labs with more than 20 years of experience in computer systems research and development. His research interests include mobile and ubiquitous computing, human mobility, virtualization, security, and privacy. Cáceres holds a PhD in computer science from the University of California, Berkeley, and is an ACM Distinguished Scientist. Contact him at ramon@research.att.com.

Adrian Friday is a senior lecturer in the School of Computing and Communications at Lancaster University and an active researcher with more than 18 years of experience in developing infrastructure to support mobile and ubiquitous computing. His research interests include the challenges of ubicomp deployment, open public display networks, and the application of ubiquitous technologies to mitigating climate change. Friday received his PhD in distributed systems support for mobile computing from Lancaster University. Contact him at a.friday@lancaster.ac.uk.