Ubicomp Systems at 20:
Progress, Opportunities, and Challenges

Ramón Cáceres
AT&T Labs
Florham Park, NJ, USA
ramon@research.att.com

Adrian Friday
Lancaster University
Lancaster, UK
adrian@comp.lancs.ac.uk

***Abstract*—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.**

I. INTRODUCTION

Twenty years ago, Mark Weiser set forth his compelling vision of *ubiquitous computing* (ubicomp) [43], giving rise to a rich multifaceted area of research. This community has spawned several major conferences, journals, and magazines, including *IEEE Pervasive Computing* ten years ago. At this milestone, we take stock of where ubicomp systems research has journeyed and postulate the major challenges for going 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 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.

II. THE ORIGINAL VISION

Let's first review the early notions of ubicomp—those that inspired our community and 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 that each pose their own forecasts of technological futures with varying degrees of specificity and alignment with these two.

A. Weiser's Calm, Integrated World

In 1991, Weiser envisioned a world where interaction occurs with everyday but computationally augmented artefacts using natural interactions, our senses, and the spoken word—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 neighbourhood's movements, then 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; 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 customises her environment: computational 'tabs' are intentionally shared—she has a view on her colleague Joe's tab that she can bring into focus if need be. *The environment is programmable*—Sal 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.

B. The Ambient Home of 2020

Similarly, stemming from an attempt to predict the user-friendly home of 2010, the Information Society Technologies Advisory Group in 2001 envision Ambient Intelligence, in which devices support ubiquitous information, communication and entertainment [14]. Ambient Intelligence emphasises similar views of efficiency and user-empowerment: 'Maria' is identified by 'the ambient' and walks effortlessly through airport security.

Ambient Intelligence implies a similar level of digital-physical integration as 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.

C. 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, unimaginable increases in available computation, storage and communication available to us, especially on personal devices—these visions of ubicomp have still not been achieved. Why is this?

These visions do important work in galvanising the research community, but aren't intended to be templates for ubicomp. They are 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 [36]:

> *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 [18].

Genevieve Bell and Paul Dourish question whether ubicomp is always destined to be framed as an artefact of an unachievable 'near future' and point out exemplars of technologies that have reached some degree of ubiquity by the actual 21^{st} century [4]. 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 [30].

III. 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 the authors believe are important for constructing more general ubicomp systems.

A. Ubiquitous Data

In the current social and political climate, we cannot imagine any kind of technology being relied upon as a sufficient guarantor to enable us to 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 ambient, 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:

1) *When can we infer with certainty?* One reason we might not trust ubicomp with recognising 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 need to weigh this evidence and make a judgement of when and how to react. The severity and importance of the outcome is certainly application- and context-dependent. Context toolkit [38] importantly abstracts sensing from the application. PersonisAD [2] separates rules from application code and makes these open to scrutiny by users. There is ample scope for better programming models that help us reason more cleanly about uncertainty and lead us to more clearly articulated reactions from ubicomp environments [13].
2) *Where is ubiquitous data located?* Ubiquitous access to data raises the important question of where ubiquitous data lives. Certainly, a global ubiquitous data store is not practicable 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 enforcement of 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." [23]. As far as we know, there has been little attempt to address the issue of data location during the last decade.
3) *How long should data persist and who can access it?* 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 free space in the car park. Intuitively it seems acceptable that this is public information. However, if the data is more intrusive and can identify particular vehicles or persons, 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's kept and forgotten by the system is often under articulated, so this issue is not 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 as often notions of identity are weak.
4) *How do we express our privacy wishes?* What is tracked and shared by the environment, and what remains private to the individual, is still very much unsolved. Langheinrich established some principles for guiding privacy-aware design [29]. Canny and Duan build on these to establish a prototype trustworthy smart environment with audio-video capture that respects an individual's "data discretion" by encoding access rights in the sensor data [8]. More recently Krumm [25] focuses on preserving identity and home-location privacy from inference attacks based on large-scale observations of past journeys. Krumm nicely highlights how even with minimal information our very 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, i.e. data that can only be observed if the individual or device is present in a given environment. This has been used to authenticate the presence of an individual, for instance Kindberg's formative work on context authentication [24]. At a smaller scale, Rene Mayrhofer and Hans Gellersen elegantly use shared observations of context to securely link two devices that are shaken together [31].

B. Understanding the World

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

Microsoft's EasyLiving required a detailed 3D model of the room, enabling content to be sensibly placed to suit its users. [5]. In Sentient Computing actions are triggered based on spatial context and relations such as view and proximity of persons and artefacts. [20] This necessitates fine-grained indoor positioning—the availability and calibration of which is still regarded as challenging and expensive [19]. One interesting and recent approach is to explore 'open data' where models are built collaboratively and refined incrementally by multiple users. Jun-Geun Park et al. apply this open crowdsourcing-based approach for building a location system based on WiFi fingerprinting [33].

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 [44] for context-aware applications such as active maps and locating artefacts and persons. Lamming detects relative co-presence, rather than absolute position, by tagging artefacts with simple IR beacons called 'SPECs' that record the identities of other SPECs they encounter [27]. This enables simple contextual applications based on patterns and changes of association and dwell time.

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 behaviour. Keintz et al. has shown how context sensing can be applied to support children with autism and their caregivers [21]. Patterson used a combination of body-worn sensors with location tracking and mobile communications to offer support with transportation choices to individuals with mild cognitive disabilities. These exemplify a new class of ubicomp as utilitarian, educational, and assistive technologies, rather than actors of convenience or efficiency. There are considerable challenges to improve the generality and portability of classifiers, and to reduce the size of the training datasets [28].

Outdoors, over the last decade, location has become easy to track to within a few tens of meters due to GPS, A-GPS, WiFi fingerprinting, and cellular antenna fingerprinting capabilities available on many mobile phone handsets. These widely available platforms have ensured that location, its protection and sharing, remains an important research topic [6]. The smartphone has proved itself to be a versatile tool suitable for a wide range of ubicomp applications, including recognising aspects of location and emotional state, and communicating this automatically with one's online social network [32].

C. 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 interests of expediency, researchers have to build with the hardware and software components available to them—and this is entirely reasonable. However, this does lead to unique and complex systems that are hard to transfer to new environments, due to the 'magic' required to configure and use them [3], and their tight binding to their particular building blocks and environment. There have been platform-oriented projects such as UIUC's Gaia [37] and MIT's Oxygen [16] that have sought a principled separation between application and system [39]. Such projects have enabled researchers to write smart-room applications and deploy them in multiple locations (e.g., UIUC was at one point linked internationally to the Tokuda Lab at Keio University in Japan).

A key challenge of creating ubicomp systems that can be deployed in more than one environment and for substantial periods of time is the degree of change or volatility experienced [12, ch. 16]:

1) *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 these components. 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 [39]. Hewlett-Packard's Cooltown took a human-centric approach based on web technologies—artefacts tagged with URIs were resolved when users triggered some action [22]. The user simply tried again when an action didn't occur.
2) *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. [16]. Stanford's Interactive Workspaces followed a data-driven approach where parts of the smart environment communicate 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 achieved this by using a 'patch panel' application [35]. Adaptation is also triggered when context changes in context-aware systems. We'd like to note that ubicomp environments are *almost always shared*—yet adapting to groups of

users and devices continues to be systematically underexplored in the literature.

3) *Evolution: Adapting to the unexpected.* The goal should be 'open loop' adaptation, i.e. the ability for the environment to cope with users, devices and software it has not seen before. Evolving to incorporate unanticipated elements at runtime (and in ubicomp, it's always runtime!) is a fascinating unsolved challenge. We may 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, Bayou cleverly exploits serendipitous communication when in range to resolve shared state in 'anti-entropy sessions' [40]. Aura's 'cyber foraging' enables mobile users to exploit proximate compute and storage resources to reduce battery demand [17]. And SpeakEasy transfers mobile code to enable 'recombinant computing' so that users can walk up and interact with devices they've not hitherto encountered [15]. 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 [41].

D. Programming Ubicomp

It's not currently possible to write a portable ubicomp application. There's neither a common runtime 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 [23]? Superficially at least, we're no closer to knowing the answer than we were then.

This doesn't mean that nascent ubicomp environments are not programmable; they are 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 behaviours be created from system components, or to trigger certain actions when events occur (as in Sentient Computing, Context Toolkit, and PersonisAD), so that components written by experts can be assembled by those who are less technical. Ultimately, this direction leads towards end-user programming (as in CAMP [42]).

E. Reducing the Infrastructure Burden

By its nature, ubicomp requires low-latency interaction with users and environments. At least part of a ubicomp application needs to be tightly bound to the infrastructure local to where this 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 or need for rolling out dedicated infrastructure to enable a ubiquitous application. Jennifer Mankoff has demonstrated how to use very 'low-fidelity' sensing (e.g., shopping receipts) to improve awareness of nutrition. Shwetak Patel and his colleagues have also shown several cunning examples where machine learning is applied to 'single point sensing' of existing domestic infrastructure such as mains wiring, plumbing and HVAC ducting, to reveal behaviour throughout the home without the need for extra sensors [34].

We can also exploit pre-existing infrastructure. RF fingerprints can create location systems such as RightSPOT [26]. Mixed-reality games exploit GPS urban-canyon effects to enhance gameplay, and 'seamful' games exploit WiFi access points as landmarks for games played on city streets [9].

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 recognised using commodity webcams (now widely deployed and integrated into most laptops) and vision techniques [11]. Furthermore, 3D printing technology is becoming a commodity within budgetary reach of at least schools and businesses if not homes. Printable electronics is an extremely active field of research that is 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. Analysing this data (e.g. photographs) often requires human perception. An increasing number of projects draw on Amazon's Mechanical Turk to reduce the burden of digitising and classifying such input. This access to 'human processors' is exciting for research projects, but does not scale ubiquitously, and raises the more general question of who pays for ubicomp services (we return to this question later).

F. Energy Impact & Awareness

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

IV. OPPORTUNITIES FOR GROWING THE INFRASTRUCTURE

As we have noted, long-lived and large-scale ubicomp infrastructure has so far failed to materialize for a variety of reasons. However, we see opportunities to grow the needed infrastructure by leveraging recent progress in technology areas not directly motivated by ubicomp. We next discuss two such areas: utility computing hosted in the cloud, and the Internet of Things.

A. Utility Computing in the Cloud

Utility computing can help realize the ubiquitous computing 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 [10]. This notion has become 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 to 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 imporant advantages, including improved flexibility, portability, longevity, and privacy [7]. 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 backend resources for ubicomp applications, it can't provide all the necessary infrastructure. In particular, ubicomp also requires widespread infrastructure that is 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.

B. Internet of Things

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

The world we inhabit is 'getting smarter all the time' on its own accord, courtesy of government and industry. Buildings incorporate sensors and actuators for control of HVAC, motion-triggered lighting, intruder detection, fault detection, and so on. Even domestic 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 (e.g. elevators, copiers) can already 'call home' for engineering support in the event of failure. Our cars are becoming increasingly densely sensed, not only for measuring the ongoing operation of the car and the environment it encounters, but also increasingly for the safety and comfort of passengers. Then there are sensors in our civic infrastructure and roads. All this increasing 'smartness' is surely an opportunity to the ubicomp community, providing this infrastructure were open to us.

Kevin Ashton, co-founder of AutoID Labs, first used the term 'Internet of Things' (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 2009 [1]:

> *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 now a subject of international conferences and it's gaining momentum. The UK government have just invested £0.5m to conduct formative studies in the formation of an open application and services ecosystem with 'open availability of data from 'things'' and 'harmonised access [...] across organisations'.

There is an alternative definition of IoT: Web sites 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 proved 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 programme 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, while potential infrastructure is out there and growing. We only need to harness it! We do not advocate that this should be done clumsily, nor that everything be ubiquitously open to all—the potential for misuse for surveillance, control, and cyber terrorism purposes is real. Taking advantage of these opportunities will require new ways of opening up access to otherwise private or enclosed infrastructures, which is neither technically nor politically trivial.

V. OUTLOOK FOR LARGE-SCALE DEPLOYMENT

While there have been many successful research prototypes of ubicomp systems, the technology in such prototypes won't see wide adoption until a number of 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 infrastructure 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 consumer privacy. These conflicts are reflected in the terms of service of popular free online services, which typically grant the provider various rights to the data contributed by 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 [7]. 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 ubicomp systems. Individual consumers have long proven unwilling or unable to manage their personal computer systems well. There is no reason to believe they will prove any better at managing ubicomp systems. This observation argues for service providers to do the managing on behalf of consumers.

Managed services would reduce complexity for the end user and help technology fade into the background—a central aim of Weiser's original ubicomp vision. However, such managed 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.

VI. CONCLUSIONS

In the twenty years since Weiser articulated the ubiquitous computing 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 are complementary to 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 to be overcome 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] K. Ashton. That 'internet of things' thing. *RFID Journal*, June 2006.

[2] M. Assad, D. J. Carmichael, J. Kay, and B. Kummerfeld. Personisad: Distributed, active, scrutable model framework for context-aware services. In A. LaMarca, M. Langheinrich, and K. N. Truong, editors, *Pervasive Computing, 5th International Conference, PERVASIVE 2007, Toronto, Canada, May 13-16, 2007, Proceedings*, volume 4480 of *Lecture Notes in Computer Science*, pages 55–72. Springer, 2007.

[3] J. Barton and J. Pierce. Quantifying magic in ubicomp systems scenarios. *Position Paper for UbiSys Workshop at Ubicomp 2006*, 2006.

[4] G. Bell and P. Dourish. Yesterday's tomorrows: notes on ubiquitous computing's dominant vision. *Personal and Ubiquitous Computing*, 11(2):133–143, 2007.

[5] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. Easyliving: Technologies for intelligent environments. In *Handheld and Ubiquitous Computing*, pages 97–119. Springer, 2000.

[6] A. J. B. Brush, J. Krumm, and J. Scott. Exploring end user preferences for location obfuscation, location-based services, and the value of location. In J. E. Bardram, M. Langheinrich, K. N. Truong, and P. Nixon, editors, *UbiComp 2010: Ubiquitous Computing, 12th International Conference, UbiComp 2010, Copenhagen, Denmark, September 26-29, 2019, Proceedings*, pages 95–104. ACM, 2010.

[7] R. Cáceres, L. Cox, H. Lim, A. Shakimov, and A. Varshavsky. Virtual individual servers as privacy-preserving proxies for mobile devices. In *Proc. of MobiHeld*, pages 37–42, New York, NY, USA, 2009. ACM.

[8] J. Canny and T. Duan. Protecting user data in ubiquitous computing environments: towards trustworthy environments. In *Proceedings of privacy-enhancing technologies (PET)*, Toronto, 2004.

[9] M. Chalmers, M. Bell, B. Brown, M. Hall, S. Sherwood, and P. Tennent. Gaming on the edge: using seams in ubicomp games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 306–309. ACM, 2005.

[10] F. Corbató and V. A. Vyssotsky. Introduction and overview of the multics system. In *Proc. of Fall Joint Computer Conference*, 1965.

[11] E. Costanza, M. Giaccone, O. Kueng, S. Shelley, and J. Huang. Ubicomp to the masses: a large-scale study of two tangible interfaces for download. In J. E. Bardram, M. Langheinrich, K. N. Truong, and P. Nixon, editors, *UbiComp 2010: Ubiquitous Computing, 12th International Conference, UbiComp 2010, Copenhagen, Denmark, September 26-29, 2019, Proceedings*, pages 173–182. ACM, 2010.

[12] G. Coulouris, J. Dollimore, T. Kindberg, and G. Blair. *Distributed Systems: Concepts and Design (5th Edition)*. Addison Wesley, 5 edition, May 2011.

[13] S. Dobson and P. Nixon. More principled design of pervasive computing systems. In R. Bastide, P. Palanque, and J. Roth, editors, *Engineering Human Computer Interaction and Interactive Systems*, volume 3425 of *Lecture Notes in Computer Science*, pages 103–111. Springer Berlin / Heidelberg, 2005.

[14] K. Ducatel and B. D. G. d. l. S. d. l. I. Comisión Europea (Bruselas. *Scenarios for ambient intelligence in 2010*. Office for official publications of the European Communities, 2001.

[15] W. K. Edwards, M. W. Newman, J. Sedivy, T. Smith, and S. Izadi. Challenge: recombinant computing and the speakeasy approach. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, MobiCom '02, pages 279–286, New York, NY, USA, 2002. ACM.

[16] K. Gajos, L. Weisman, and H. Shrobe. Design principles for resource management systems for intelligent spaces. In *Proceedings of the 2nd international conference on Self-adaptive software: applications*, pages 198–215. Springer-Verlag, 2001.

[17] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste. Project aura: Toward distraction-free pervasive computing. *IEEE Pervasive computing*, pages 22–31, 2002.

[18] W. Gaver. The video window: my life with a ludic system. *Personal and Ubiquitous Computing*, 10(2-3):60–65, 2006.

[19] R. Harle and A. Hopper. Building World Models By Ray-tracing Within Ceiling-Mounted Positioning Systems. In *Proceedings of the Fifth International Conference on Ubiquitous Computing (UbiComp 03)*, Oct. 2003. Conference Paper.

[20] A. Hopper. Sentient Computing - abridged and updated version of the Royal Society Clifford Paterson Lecture, 1999. In K. S.-J. Andrew Herbert, editor, *Computer Systems: Theory, Technology, and Applications: A Tribuite to Roger Needham*, Monographs in Computer Science, pages 125–131, Dec. 2003. Conference Paper.

[21] J. A. Kientz, G. R. Hayes, T. L. Westeyn, T. Starner, and G. D. Abowd. Pervasive computing and autism: Assisting caregivers of children with special needs. *IEEE Pervasive Computing*, 6(1):28–35, 2007.

[22] T. Kindberg, J. Barton, J. Morgan, G. Becker, D. Caswell, P. Debaty, G. Gopal, M. Frid, V. Krishnan, H. Morris, J. Schettino, B. Serra, and M. Spasojevic. People, places, things: web presence for the real world. *Mob. Netw. Appl.*, 7(5):365–376, October 2002.

[23] T. Kindberg and A. Fox. System software for ubiquitous computing. *IEEE Pervasive Computing*, 1:70–81, January 2002.

[24] T. Kindberg and K. Zhang. Context authentication using constrained channels. In *In Fourth IEEE Workshop on Mobile Computing Systems and Applications*, pages 14–21. IEEE Computer Society, 2002.

[25] J. Krumm. Inference attacks on location tracks. In A. LaMarca, M. Langheinrich, and K. N. Truong, editors, *Pervasive Computing, 5th International Conference, PERVASIVE 2007, Toronto, Canada, May 13-16, 2007, Proceedings*, volume 4480 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2007.

[26] J. Krumm, G. Cermak, and E. Horvitz. Rightspot: A novel sense of location for a smart personal object. In A. K. Dey, A. Schmidt, and J. F. McCarthy, editors, *Ubicomp*, volume 2864 of *Lecture Notes in Computer Science*, pages 36–43. Springer, 2003.

[27] M. Lamming and D. Bohm. Specs: Another approach to human context and activity sensing research, using tiny peer-to-peer wireless computers. In A. Dey, A. Schmidt, and J. McCarthy, editors, *UbiComp 2003: Ubiquitous Computing*, volume 2864 of *Lecture Notes in Computer Science*, pages 192–199. Springer Berlin / Heidelberg, 2003.

[28] N. D. Lane, D. Lymberopoulos, F. Zhao, and A. T. Campbell. Hapori: context-based local search for mobile phones using community behavioral modeling and similarity. In J. E. Bardram, M. Langheinrich, K. N. Truong, and P. Nixon, editors, *UbiComp 2010: Ubiquitous Computing, 12th International Conference, UbiComp 2010, Copenhagen, Denmark, September 26-29, 2019, Proceedings*, pages 109–118. ACM, 2010.

[29] M. Langheinrich. Privacy by design - principles of privacy-aware ubiquitous systems. In G. D. Abowd, B. Brumitt, and S. A. Shafer, editors, *Ubicomp*, volume 2201 of *Lecture Notes in Computer Science*, pages 273–291. Springer, 2001.

[30] S. Mainwaring, M. Chang, and K. Anderson. Infrastructures and their discontents: Implications for ubicomp. *UbiComp 2004: Ubiquitous Computing*, pages 418–432, 2004.

[31] R. Mayrhofer and H. Gellersen. Shake well before use: Intuitive and secure pairing of mobile devices. *IEEE Trans. Mob. Comput.*, pages 792–806, 2009.

[32] E. Miluzzo, N. Lane, S. Eisenman, and A. Campbell. Cenceme–injecting sensing presence into social networking applications. *Smart Sensing and Context*, pages 1–28, 2007.

[33] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie. Growing an organic indoor location system. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 271–284. ACM, 2010.

[34] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd. At the flick of a switch: detecting and classifying unique electrical events on the residential power line. In *Proceedings of the 9th international conference on Ubiquitous computing*, UbiComp '07, pages 271–288, Berlin, Heidelberg, 2007. Springer-Verlag.

[35] S. Ponnekanti, B. Johanson, E. Kiciman, and A. Fox. Portability, extensibility and robustness in iROS. *Proc. of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 11–19, March 2003.

[36] Y. Rogers. Moving on from weiser's vision of calm computing: Engaging ubicomp experiences. In *UbiComp 2006: Ubiquitous Computing*, volume 4206, pages 404–421, 2006.

[37] M. Roman and R. H. Campbell. Gaia: enabling active spaces. In *EW 9: Proceedings of the 9th workshop on ACM SIGOPS European workshop*, pages 229–234, New York, NY, USA, 2000. ACM.

[38] D. Salber, A. Dey, and G. Abowd. The context toolkit: aiding the development of context-enabled applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 434–441. ACM, 1999.

[39] G. Schiele, M. Handte, and C. Becker. Pervasive computing middleware. *Handbook of Ambient Intelligence and Smart Environments*, pages 201–227, 2010.

[40] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser. Managing update conflicts in bayou, a weakly connected replicated storage system. In *SOSP '95: Proceedings of the fifteenth ACM symposium on Operating systems principles*, pages 172–182, New York, NY, USA, 1995. ACM.

[41] H. Tokuda, K. Takashio, J. Nakazawa, K. Matsumiya, M. Ito, and M. Saito. Sf2: Smart furniture for creating ubiquitous applications. In *Applications and the Internet Workshops, 2004. SAINT 2004 Workshops. 2004 International Symposium on*, pages 423–429. IEEE, 2004.

[42] 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. In N. Davies, E. D. Mynatt, and I. Siio, editors, *Ubicomp*, volume 3205 of *Lecture Notes in Computer Science*, pages 143–160. Springer, 2004.

[43] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):66–75, September 1991.

[44] E. Welbourne, K. Koscher, E. Soroush, M. Balazinska, and G. Borriello. Longitudinal study of a building-scale rfid ecosystem. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, pages 69–82, New York, NY, USA, 2009. ACM.