Beyond Prototypes: Challenges in Deploying Ubiquitous Systems

The authors discuss the problems in creating the next generation of widely deployed ubiquitous computing systems and articulate current technical and sociological challenges to inspire researchers in the field.

Nigel Davies
Lancaster University and University of Arizona

Hans-Werner Gellersen
Lancaster University

Despite being written over 10 years ago, many aspects of Mark Weiser's vision of ubiquitous computing appear as futuristic today as they did in 1991.[1] The reasons for this apparent lack of progress are manifold, and other articles in this issue explore remaining technical problems in specific areas of ubiquitous computing research. We focus on the technical and sociological challenges of creating systems that are ubiquitous in the more general sense of the word—systems that extend beyond mere laboratory prototypes. Such systems are deployed to the extent that they become an integrated part of our everyday lives. Only when we have achieved this degree of pervasiveness will Weiser's vision become reality. Unfortunately, we are still many years from creating such systems.

Here, we discuss significant research challenges that have yet to be addressed. Central to documenting these challenges is recognizing the context within which we are operating. So, we first describe the technical and social changes of the 1990s that directly affected ubiquitous computing.

The decade of ubiquitous information and communication

Since Mark Weiser wrote his seminal article, technology has advanced along many dimensions. In addition to well-known developments in capabilities such as processing power and storage in portable devices, other significant—but perhaps less obvious—developments have occurred. These developments are likely to affect our ability to deploy ubiquitous computing systems. They include new technologies—such as the Global Positioning System (GPS), smart cards, and radio frequency identification (RFID) tags—and social developments such as the increasingly widespread acceptance of video surveillance in public places. However, the decade's most striking developments (with respect to ubiquitous computing) have undoubtedly been the emergence of the Web as a global information and service resource and the widespread adoption of digital mobile telephony, letting users experience nearly ubiquitous wireless communications.

The World Wide Web

The Web's emergence has fundamentally changed the way many people interact with computers. It has also created a culture that is substantially more amenable to the deployment of ubiquitous computing environments than that which existed when Weiser first articulated his vision.

Most obviously, the Web has created a nearly ubiquitous information and communications infrastructure. We can now access a huge wealth of knowledge and services from almost any computer, including low-power mobile devices such as smart phones and PDAs. However, the Web has had other, more subtle effects on our culture.

First, the increased use of computers as portals to

1536-1268/02/$17.00 © 2002 IEEE

the Web has reduced many users' sense of attachment to a single computing device. In his article on *calm technology*,[2] Weiser drew parallels between personal computers and automobiles, saying that both are special, relatively expensive items with which users form relationships. One example of this relationship between users and their personal computers is the common practice of anthropomorphizing a computer—for example, by naming it. While such relationships are likely to persist for some period of time, many users now relate not to their computer but rather to their point of presence within the digital world—typically, their homepage, portal, or email service. So, for users who extensively use Web services and information, the computer that they use to access these things has become largely irrelevant. Many users commonly access the same point in digital space from several different devices (office or home PC, cell phone, PDA, and so forth) throughout the course of a typical day. Consequently, for most users, computers themselves are becoming increasingly unimportant—what matters is the view a particular machine provides of the digital world. In this sense, we are well on the way to computers "disappearing" and users being free to focus beyond them.

The Web has also accelerated the development of social and legal constructs for dealing with computationally rich environments. In particular, Web users have had to contend with significant challenges to their privacy, primarily in the form of logging technologies that can generate substantial amounts of detail about a given user's Web activities. This has raised significant concerns among many Web users. Commercial organizations, legislators, and privacy groups are struggling to come to terms with new technology's implications for an individual's privacy.

Mobile communications for the masses

In addition to creating a ubiquitous information infrastructure, the last decade also witnessed the widespread deployment and adoption of digital mobile communications, primarily in the form of the global system for mobile communications.[3] The Internet's growth, particularly in the US, sometimes overshadows the huge impact that technologies such as GSM have had on many consumers worldwide. There are currently an estimated 800 million subscribers to mobile phone services, of which 65 percent use GSM-based networks (see www.gsmworld.com/news/statistics/index.shtml). During September 2001, these users sent over 23 billion SMS (short message service) messages—part of a steep upward trend as usage heads toward one billion messages per day (see www.gsmworld.com/news/statistics/index.shtml). Furthermore, modern handsets offer far more capabilities than early ubiquitous computing devices such as ParcTabs in roughly the same form factor. A typical phone today might include simple PDA applications such as a calendar and to-do lists, games, text-messaging facilities, voice communications (of course), Web access, and even simple voice recognition.

Modern mobile phones have another important property that Weiser associated with ubiquitous computing devices: many users view them as a commodity that they can find and use anywhere they travel. An important factor in forming this view is the price. For most users, handsets are essentially free and are replaced relatively frequently. Weiser suggested that users can use any ParcTab as if it were their own. Separating the handset from the subscriber identity module card found in GSM systems approximates this model of operation. By inserting their SIM card into a handset, subscribers can automatically use the handset, placing and receiving calls as if it were their own phone. Of course, this represents only a partial solution because users typically own only one SIM card and hence can't use multiple devices simultaneously. Moreover, users must consciously insert their SIM card into a handset—they can't just pick up and use any handset. Similarly, although phones are perceived as cheap, they are not usually left lying around for casual use in quite the way Weiser described. Despite these shortcomings, using SIM cards demonstrates once again that, for many users, the end-system is becoming less important than the access it provides to the digital world.

Why the whole is more than the sum of its parts

Weiser's vision still seems like science fiction, primarily due to a lack of integration in existing systems. Consider the scenario of Sal's world that Weiser outlined in his article (see page 24 in this issue); clearly, the technologies required are either already deployed or could be deployed relatively trivially.

For example, to realize the foreview mirror Weiser describes would require

- Equipping Sal's car with a satellite navigation and information system
- Including some mechanism for detecting the availability of parking spaces near her office building
- Including a detection or notification mechanism that can highlight new shops opening in her area

These things are already deployed in many environments. Satellite navigation and information systems are standard in many high-end cars; several different means exist for detecting parking availability (for example, we could tap into the parking lot's video surveillance camera and run an algorithm for detecting spaces); and we can probably already identify new shops using a neighborhood Web page or store guide. However, analyzing but a single scenario in Sal's world uncovers a host of issues associated with deploying ubiquitous computing systems. The components from which to construct such systems might already be available, but they are typically conceived and operated independently, in the context of their own restricted view of the world.

In the following discussion, we assume that the "simple" problems associated with large-scale ad hoc software integration—such as naming, interface specification, fault tolerance, and configuration management—have all been solved. We focus on the problems of creating integrated ubiquitous computing systems.

Technical challenges

The first difficulty for anyone attempting to realize Weiser's vision of the foreview system is that it clearly has detailed knowledge of Sal's current task. In particular, it knows that Sal is driving to work and hence will be interested in viewing parking spaces at her office. Assuming that Sal has not explicitly told the system about her current task, it must deduce this information, perhaps based on the time, day, or direction the car is traveling. Clearly, the possibility of making erroneous deductions is high: the problems associated with modern office packages that attempt to predict user intentions illustrate the task's difficulty, even when the domain is extremely restricted.

Assuming that the system has determined that Sal is interested in parking availability at her office, the next challenge is to make the correct associations between the various components involved in providing this information. This process is relatively easy for humans but extremely difficult in software. For example, we quickly determined that we might locate available parking using surveillance cameras, but how could this ability to create new associations between components be realized in a computer system? Moreover, suppose that there are many parking lots near the office, and many cameras in each lot. What criteria does the foreview system use to decide which parking lot to investigate (closest, cheapest, or most frequently used), and how does it determine which cameras apply to the task at hand (especially if the cameras are not static)?

Once the foreview system has found a video image of a parking lot that appears to contain a space, the next challenge is to translate this information into driving directions or a map. This would require determining the parking space's precise geographical coordinates—something that would probably imply knowing the positioning of the parking lot's cameras. Furthermore, if the cameras describe their field of view using a different model of location than that used in the car navigation system (such as a symbolic model versus a geographic model), the system must address the problem of mapping between different location models. This area has seen some promising research,[4] but the problems are by no means solved.

Social and legal issues

Even if we assume that realizing the foreview application is technically feasible, numerous problems would still inhibit the system's deployment. For example, surveillance data can obviously contain sensitive information such as the identity of people at the scene, either directly through recognition or indirectly through vehicle identification. Consequently, data protection legislation applies (in European countries, protection is decided on the basis of Directive 95/46/ec[5]). So, the video surveillance component could only deliver images if it sought explicit consent from the people who might be identified—which clearly is technically and socially infeasible.[6]

Developing effective business models for ubiquitous computing systems will clearly be crucial to their success, yet at best, system designers poorly understand this issue.

The obvious alternative would be for the video surveillance component to return specific information on parking availability—removing the possibility of other components accessing sensitive information but reducing the component's general usefulness in a ubiquitous computing environment. However, designing such components would still require care to avoid personal data spills into public services. For example, a service that reports reserved spaces as being possible parking spaces would provide implicit information on the movement of those spaces' owners.

The reduction in generality also has significant consequences for ubiquitous computing system designers. For example, one often-cited application of a ubiquitous computing environment is to help users locate objects such as missing wallets or keys. It is obvious to a designer how the components that were used to find a parking space could also help locate a user's missing items if the user thinks he might have lost them when he parked his car. In this case, the benefit of a ubiquitous system operating as an integrated whole as opposed to a set of separate components is clear. However, such generalization would be impossible if the surveillance component offered only an interface to locate parking spaces. Of course, it is important to note that it is not just infrastructure components that require careful design to protect user privacy. For example, if the foreview system's store-finder feature were implemented by making requests to a Web server providing local information, the foreview system would be providing detailed information about Sal's movements to the server.

Economic concerns

Finally, we observe that while the parking-space application delivers value to Sal—and she might well be prepared to pay for such a service—the cost of providing the application is distributed among numerous components. Such components might include the video surveillance system and its operators and the communications provider that enables the information to reach Sal's car. How do these service providers recover their costs? Can one parking lot offer the information for free, hoping to recover the costs through increased use? How would we ensure fair competition between neighboring parking lots? How would we bill Sal, and how much would she pay for the information? Developing effective business models for ubiquitous computing systems will clearly be crucial to their success, yet at best, system designers poorly understand this issue.

Experiences in deploying ubiquitous systems

Many systems highlight development trends that we generally associate with the drive toward ubiquitous computing, such as the provision of continuous service, contextualized use, and ad hoc collaboration of components. Here, we review some of these systems—ones that, in one sense or another, have been deployed in environments beyond strictly controlled lab settings. First, we discuss the pioneering work at Xerox Parc and at AT&T Labs in Cambridge,[7] both of which emphasize deploying infrastructures to facilitate ubiquitous computing applications. In contrast, we then discuss Guide and Cooltown,[8] which focus more on creating actual ubiquitous computing applications and deploying them in large user communities (a similar project is Classroom 2000;[9] see "The Human Experience" in this issue for more information on that project). Finally, we examine the MediaCup[10] project at Karlsruhe, which explores the issue of deployment in everyday artifacts.

Xerox Parc's ubiquitous computing experiment

Weiser's articles on ubiquitous computing usually presented not just ideas but descriptions of systems designed, developed, and deployed as part of Xerox Parc's Ubicomp project. Obviously, these systems were bound by the technology of the early 1990s. Because the Ubicomp project could not push computing and networking into everyday objects to the extent envisioned, it instead approached the diversity of devices in a future world by designing and implementing computers representing different scales—from inch- through yard-sized devices.

Devices with characteristics similar to those prototyped have since become common: clear parallels exist between inch-scale ParcTabs and phone and PDA products; foot-scale Pads[1] and reading appliances and tablet PCs; and the yard-scale Liveboard and interactive white board products. However, 10 years ago, developing ParcTabs, Pads, and Liveboards required tremendous effort, because researchers had to construct the entire system (network, communication, user interface, and applications) from scratch. Not surprisingly, the technical challenges of developing the individual devices meant that the overall project fell short of addressing integration issues, and developments such as the ParcTab and the Liveboard, which were tremendously influential in their own right, remained mostly separate.

Among the systems explored in the Ubicomp project, the ParcTab particularly influenced ubiquitous computing research because it introduced two major features that represent a departure from desktop computing: continuous service and contextualized use. The device's lightweight design enabled continuous service in combination with permanent connectivity to remote applications, using a wireless infrared network. The network's cell-based nature supported localization of a ParcTab device, enabling context-aware applications.[11] The entire ParcTab system was deployed in a large office environment, with first 25 and later 50 infrared cells, to study the use and evolution of applications with a community of more than 40 people.[12] The experiment's scale facilitated an understanding of design and user issues that demonstrations or simulations could not have achieved.

Of the many ParcTab observations reported, we relate just one here to underline the value of deploying working systems. Researchers identified email access early on as a particularly compelling ParcTab application. It was instantly appealing in demonstrations and popular with users when first introduced to the system. However, in the deployed system, wherever a ParcTab infrared cell existed, there was also a powerful workstation that made a much better platform for reading email. The only exceptions to this rule were conference rooms; in these, all the conference participants (typically about 20 people) would try to use their ParcTabs simultaneously over the limited bandwidth of the shared cell, making them too slow for communication.[13] The lesson learned is that understanding a new artifacts utility requires serious deployment; experience gathered in demonstrations and lab trials can be misleading.

From the Active Badge to sentient computing

The Active Badge project, conducted in the early 1990s at the Olivetti Research Lab (now AT&T Laboratories) in Cambridge, England, heavily affected ubiquitous computing research because it implemented the first significant indoor positioning system. The system used wearable badges to emit beacons and networked sensors to detect badges and locate their wearers.[14,15] Interestingly, its development was not driven by a long-term vision such as ubiquitous computing but by a very practical concern: how to locate people in an office complex for the purpose of routing phone calls. Addressing a clearly perceived problem, the Active Badge brought a new level of utility to users that facilitated widespread deployment (various universities and research labs throughout Europe and the US deployed over 1,500 badges and 2,000 sensors, with the largest single system at Cambridge University having 200 badges and 300 sensors in daily use). Once deployed on this scale, the Active Badge system helped new applications emerge, including compelling examples of the departure from personal desktop-bound computing toward ubiquitous computing.

One such example was Teleporting, which supported mobile users by transporting the interface of their remote applications to the nearest available terminal.[16] The teleporting concept supports the vision that services become directly attached to people rather than to a particular machine, rendering the service ubiquitous and making access to any particular machine less relevant. This links back to the effect ubiquitous Web access has on the tie between the user and personal computer. For teleporting, X Windows provided the underlying machine independence, but the principles remain the same.

Teleporting and other early experiments with context-aware systems used room-scale information generated by Active Badges, but it soon became obvious that many applications require finer-grained 3D location and orientation information. This corresponds to Roy Want and his colleagues' analysis of in-building location as a key problem in ubiquitous computing.[17] The challenge is to achieve a degree of spatial resolution much closer to the human perception of space, so we can treat the

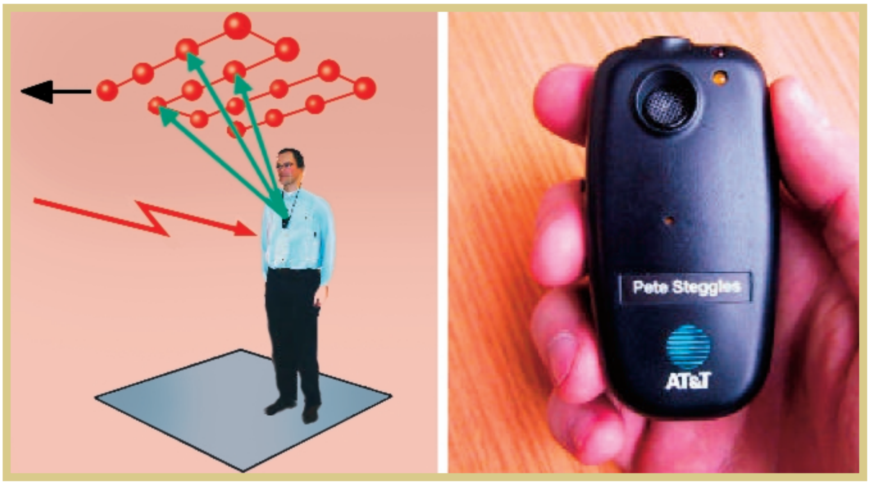

Figure 1. (a) The AT&T Active Bat system and (b) the Bat wireless tag device. Figure courtesy of M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper.[7]

physical environment's current state as common ground between computers and their users.

In work that has evolved from their earlier location-aware systems, researchers at AT&T Laboratories Cambridge have developed the Active Bat, an ultrasonic location system that achieves fine-grained 3D positioning (see Figure 1).[18] The Bat is deployed throughout the lab's three-floor office building, with all 50 staff using it continuously. With 95 percent of the sensor readings being correct to within 3 cm in 3D, the Bat is by far the most accurate deployed indoor positioning system. The Bat system serves as infrastructure for further research into sentient computing, exploring applications that become possible when computers have a fine-grained model of the physical environment in which they are used.[7] The sentient computing system maintains a software model of objects in the real world that contains up-to-date information about location and state. The system further supports a programming model in which developers can specify spaces to monitor and operations to execute based on spatial relationships.

Although the work of the Cambridge-based lab continues to be at the forefront of enabling location as a resource for ubiquitous computing applications, it has generated public controversy. In particular, introducing the Active Badge triggered Orwellian interpretations of future work environments, and tracking people continues to be a sensitive topic. This reaction to location technologies is an important reminder that understanding user concerns will be critical to obtaining sufficient acceptance to ensure successful deployment of ubiquitous computing systems.

Lancaster's Guide system

The pioneering work at Parc and at AT&T Labs required developing homegrown infrastructures; hence, the projects were primarily bound to research environments. However, the increasing availability of readily deployable mobile computing infrastructures and affordable outdoor location technology components (such as GPS) led to a series of systems that investigated ubiquitous computing beyond the office setting. In 1996, systems began to emerge that studied mobility, continuous service provision, and contextualized use in application domains such as tour guides and navigation. Examples include Cyberguide,[19] the Touring Machine,[20] and Lancaster's Guide.[21] The Lancaster Guide project was a comparatively small and focused research effort, but it involved a large field trial with real users, providing insight into issues concerning deploying ubiquitous computing systems to the wider public.[22]

The research team at Lancaster designed the Guide system to provide visitors to the city of Lancaster with the type of information normally found in a tour guide, but contextualized on the basis of the visitor's interest and movement around the city. They based the system on a distributed, dynamic information model, augmenting standard hypertext with geographic information and navigation elements, as well as context-sensitive active components. Users interact with the system using tablet PCs as end-systems, connected to information servers through an 802.11 network deployed around the historic city's major attractions. This network covers most but not all of the inner city. Its cell structure also provides end-systems with coarse-grained (approximately 200 m accuracy) location information based on their current cell identifier. Despite the coarse granularity, integrating location nonetheless supports an interesting information navigation model. Users effectively navigate the information space both explicitly through the Guide user interface and implicitly through their movements in the real world.

In 1999, the Lancaster team deployed Guide and evaluated it in a field trial for approximately four weeks. Sixty tourists visiting Lancaster used the system—people of all age groups, mostly without any previous experience with information navigation systems such as the Web.

The design and use experience gained from the Guide project uncovered various issues that related to the interplay of infrastructure and user interaction. For example, in the user interface, the team had to consider the fact that the positioning system is not available everywhere a tourist might go. They chose to involve the user in disambiguation of the system state shared between the human and computer (see Figure 2). This relates closely to the issue of dealing with ambiguity in prediction-based user interfaces investigated at Georgia Tech.[23] The general lesson to be learned from systems such as Guide is that user interaction tends to be governed by the capabilities of the infrastructure, which, in ubiquitous computing environments, is more prone to changes in composition and availability than in traditional application settings.

The Cooltown project

Although the Web infrastructure was never conceived as a general distributed systems platform, its ubiquity and versatility have made it an attractive choice for large-scale application deployment. Not surprisingly, researchers have investigated combining the Web infrastructure with ubiquitous computing concepts. For example, Accenture's CStar group has developed various demonstrators for augmented commerce, integrating Web-based e-commerce services with context-aware client technology and tagged real-world environments. (One example is the Pocket Bargain Finder, which integrates a barcode scanner and wireless Internet access to retrieve online quotes for products browsed in real-world shops.[24]) In a more general way, Hewlett-Packard Labs' Cooltown explores opportunities arising from the convergence of Web technologies, wireless networking, and new kinds of devices on the client and server sides.[25]

The Cooltown project is developing an infrastructure to support Web presence for "people, places, and things," extending the Web's ad hoc nature and mechanisms for locating and linking resources into the real world of physical entities.[25] Cooltown technology addresses physical integration with embedded server technology, virtual representations of physical places (called place managers), real-world embedded URLs (for example, emitted by beacons), and new interaction techniques such as e-squirt, a real world equivalent to drag-and-drop. HP internally deployed the technology to create office environments in which mobile users could use resources (printers, projectors, and so forth) served by place managers (maintaining Web portals to locations) and into which users can "squirt" URLs as handles to any kind of information. Interestingly, users have not changed the way they do things (for example, they don't bring URLs rather than document copies into meetings), even when the benefits seem obvious. HP has also externally deployed components of the Cooltown system in a science museum, enabling the combination of physical and virtual exploration.

From an overall systems perspective, Cooltown presents an interesting counterpoint to many other ubiquitous computing systems by placing the user in the control loop. For example, other proposed platforms suggest that dynamically requested resources should be discovered automatically, using service discovery protocols. In contrast, Cooltown involves the user in resource discovery—for example, using ad hoc compilation of descriptions and links to available resources in the Web page of a particular location (each place manager maintains a page for a given location). The argument for user involvement is that it avoids having to design systems that can carry out task analysis or form complex associations between components—processes that we have already observed are extremely difficult to achieve in computer systems.

The MediaCup experience

To cite Mark Weiser and John Seely Brown, "Ubiquitous computing is fundamentally characterized by the connection of things in the world with computation."[26] In many deployed ubiquitous computing systems, this connection is established indirectly through the approximate location of physical objects. A more direct approach, applied in Cooltown and many other projects, is to attach pointers from everyday objects to entities in the computational world, using, for example, RFID tags.[27] Taking this approach one step further, computation might be completely embedded in everyday objects to turn them into permanently connected residents in a ubiquitous computing world. The MediaCup project at the University of Karlsruhe is an experimental deployment of everyday objects activated in this sense.[10]

The guiding principle in the MediaCup project was to augment objects with a digital presence while preserving their original appearance, purpose, and use. The first objects prototyped were coffee cups equipped with a low-power microcontroller, embedded sensors, and wireless communications (see Figure 3). The embedded technology lets the cups sense their physical state and map sensor readings autonomously to a domain-specific model

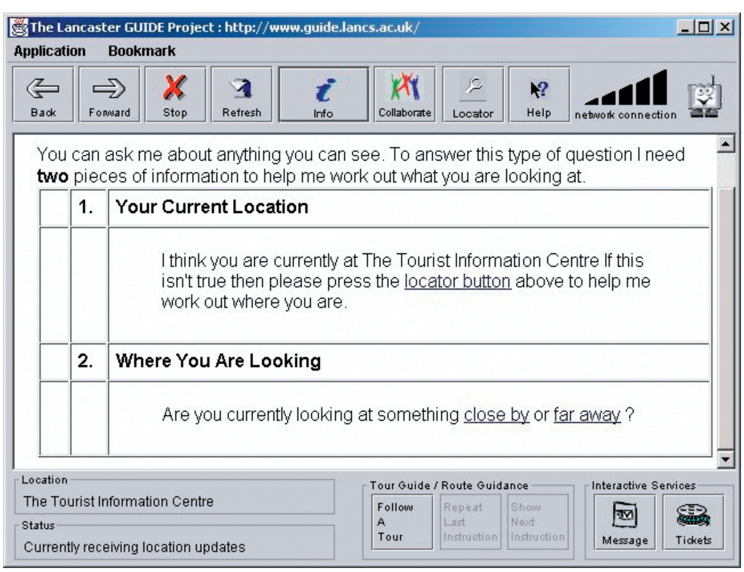

Figure 2. Involving the user in disambiguating the shared state in the Guide system.

of the cup. This object model is broadcasted at regular intervals over the wireless link to establish the object's digital presence. Karlsruhe first deployed the augmented coffee cups in September 1999 in an office environment. Since then, a small population of cups, typically five to six at a time, has been in everyday use. The cups are not personalized but are simply a shared resource in the lab's kitchen. While there are a few staff who are dedicated users, the cups are mostly used occasionally by lab members and visitors.

The MediaCup deployment is a long-term experiment aimed at understanding opportunities that might arise from the digital presence of mundane objects. Even though the MediaCup was designed without any specific use or application in mind, as the project evolves, applications that use MediaCup information have emerged. An initial application simply visualized cups and their state on a 2D map of the office environment. This was followed by more embedded applications such as door plates that used the aggregation of hot cups in a room to infer and indicate meetings, and wrist-worn PCs that issue warnings if the user picks up a cup with coffee or tea that's still too hot to drink.

Figure 3. The MediaCup.

Analysis

From a technical perspective, many of these projects have faced common problems in developing deployable prototypes. For example, energy concerns heavily influenced designs for the Active Badge, Active Bat, ParcTab, and MediaCup. Moreover, these concerns affected many aspects of the systems' designs, highlighting the integrated nature of ubiquitous computing implementations (for example, decisions regarding the temporal accuracy of sensor readings were influenced by energy considerations but have a significant impact on the types of application that can be supported). A further example of the close coupling between aspects of ubiquitous computing system design can be seen in the way in which projects such as Guide support variations in networking connectivity in both the underlying protocols and the user interface. For the research community, this implies that we will need forums to discuss projects and results that cut across various computer science topics.

In terms of deployment, only through experience can we understand the cost-versus-benefit issues in ubiquitous computing systems. For example, the Active Badge system implies a cost for the badge wearer (some inconvenience, loss of privacy), so the badge must offer a benefit (and not just a communal benefit). For example, to boost the number of badge wearers for an active badge system deployed at Lancaster, applications were designed that specifically benefited the wearer. Specifically, we offered a collection of user-sensitive public monitors that provided tailored information to users as they approached the screen (though we eventually had to remove the monitors from the building's corridors because they did not comply with safety regulations).

Probably the most significant lesson, however, is the importance of deploying prototypes—not just to evaluate their utility but also to explore ideas, discover new viewpoints, and unearth unexpected issues. Design and usage data is crucial to advancing our understanding of future ubiquitous computing systems. However, obtaining such experience is nontrivial. To achieve the levels of deployment required for realistic user trials often involves researchers addressing design tradeoffs in areas such as cost, functionality, and robustness that are more commonly the focus of product development teams.

Research challenges

Deploying ubiquitous computing systems provides insight into their design and use, but deploying systems beyond the limited existing prototypes will require significant progress toward integration. Here, we consider major research challenges we must overcome before achieving this goal. This list is not exhaustive; it simply provides illustrative examples of the requirements that supporting integration places on components of ubiquitous computing systems.

Component interaction

To gain leverage from the substantial work carried out in the distributed systems community, future components of ubiquitous computing systems must clearly follow the same basic principles as open distributed systems. They should be designed and implemented in an open and extensible manner, letting us combine components to form applications unforeseen at the time of their deployment.

Technically, this implies obvious features such as open interfaces and support for intercomponent communication.[28,29] However, deployed ubiquitous computing systems will require assurances from their components in terms of metrics such as performance, security, and reliability. Furthermore, the characteristics of ubiquitous computing environments place demands on existing platforms that the platforms have not been designed to address. For example, many existing platforms perform poorly when applied to the type of saturated computing environments Weiser described.[30] ("System Software for Ubiquitous Computing" in this issue discusses designing software components for integrated ubiquitous computing systems.)

Adaptation and contextual sensitivity

The environment in which a ubiquitous

computing component functions is subject to change.[31] Such changes might be prompted by variations in resource availability as a result of failures or the deployment of new services or by variations in patterns of usage or mobility. The importance of adaptation is well understood in the field of mobile computing.[32] However, it is significantly more complicated in ubiquitous computing systems, where there is a need to respond to a much larger set of contextual triggers. At a component level, components will be required to adapt internally. More importantly, we might also have to substantially reconfigure applications involving multiple components.

The need to manage these configuration changes in ad hoc ubiquitous environments poses significant problems. One approach to addressing these problems—one that has received considerable attention of late—is to use reflective middleware platforms.[33] Although researchers have generally focused on using reflection to support adaptation in mobile environments, its application to ubiquitous computing systems is an obvious extension.

Appropriate management mechanisms and policies

As the number of deployed components increases, system management will likely become increasingly problematic. While we want zero-configuration, low-maintenance systems, the reality is that substantial system management will still likely be required. We might expect future components to support standardized management interfaces enabling, for example, tasks such as configuration management over a wide range of components. A particular challenge for ubiquitous computing systems is that this management is unlikely to occur within the context of a single administrative domain. Indeed, for many components, the administrative domain might change dynamically—for example, depending on the proximity of different users or devices. The combination of requirements for low (or zero) administration, multidomain management, and support for rapid reconfiguration will likely raise new challenges for system management.

Component association and task analysis

In studying Weiser's scenarios of a ubiquitous computing world, we clearly get the sense of some form of intelligence working on the user's behalf to coordinate the actions of components in the infrastructure. Two areas in which this is particularly evident are the system's ability to accurately determine a user's task and intention and its ability to develop associations between components to assist the user in these activities. Achieving these objectives in anything other than extremely limited domains is an unsolved problem.

Viable economic models and supporting infrastructure

One often-cited reason for the relative lack of success in deploying ubiquitous computing systems is that none of the application scenarios seem likely to generate significant revenue. Users might pay to live in a ubiquitous computing world, but it is difficult to imagine them paying a lot of money for any one application or feature (this of course assumes that the search for a single "killer" application is unsuccessful). Consequently, the cost of deploying and operating a given component might need to be recovered in the form of many small contributions from applications that use the component. This will require support from components in terms of billing and accounting at a level previously unseen in widespread distributed systems.

User interface integration

As the number of applications operating in a ubiquitous computing environment increases, we'll need coordination between these applications to ensure we can provide a reasonable user interface.[18] This coordination might range from traditional areas such as arbitrating screen usage to new challenges such as deciding which application may use the intensity of the light in a room to communicate with the user. While user interface designs for ubiquitous computing systems are covered elsewhere in this issue, it is important to stress the lessons learned from the Guide system: the infrastructure's capabilities directly affect the user interface. So, support of the user interface becomes an issue for all components, at whatever level of the system they operate.

Social, legal, and technical solutions to privacy and security concerns

Support for adequately handling personal data challenges both legislators and ubiquitous systems developers. Traditionally, data protection legislation has tended to prohibit any capture and storage of person-related data and has only allowed exceptions bounded by a clearly defined purpose, at the end of which data records had to be deleted. This approach to privacy protection is inadequate for a modern, open information society. For example, to demand that sensitive data be deleted after its use is clearly out of sync with the Internet. Most important, legislation must acknowledge that person-related data has become a currency in the information economy.

Here lies a core problem for developers who need to create systems that better address privacy issues. Currently, users don't fully understand how the electronic trails they create can be used, so they cannot understand their personal data's value. A key challenge for future ubiquitous system designers is to empower users to evaluate the tradeoff between protection of privacy and access to improved service. Meanwhile, legislation must contribute by defining the boundaries within which such trade-offs may occur.

The challenges we've described focus on integration. However, although integration is key to future ubiquitous systems, we might also design such systems with the notion of interference in mind. By way of analogy, consider the current situation regarding electromagnetic interference from devices. An appropriate authority must certify most electronic equipment to confirm that it does not cause undue interference to neighboring electronic systems. This certification lets consumers purchase products without worrying about negative side effects on their existing equipment.

We expect the same concerns will arise when consumers start purchasing and deploying components of a ubiquitous computing environment. However, in this case, we'll need to extend the concept of interference beyond the notion of physical interference to include interference at the logical (software) level. For example, if two ubiquitous computing applications have conflicting requirements in terms of infrastructure services, they will need to resolve this conflict, ideally without user intervention. It seems reasonable to assume that future ubiquitous computing components will need to be both integration-friendly and interference-free for wide-scale deployment. How this capability is tested will likely be a significant research challenge in its own right.

ACKNOWLEDGMENTS

We thank Keith Cheverst, Tim Kindberg, Marc Langheinrich, and all the reviewers for their help in preparing this article.

REFERENCES

1. M. Weiser, " The Computer of the 21st Century," *Scientific American*, vol. 265, no. 3, Sept. 1991, pp. 66–75 (reprinted in this issue, see pp. 19–25).

2. M. Weiser and J. Brown, "Designing Calm Technology," *PowerGrid J.*, v 1.01, July 1996; also see "The Coming Age of Calm Technology," *Beyond Calculation: The Next Fifty Years of Computing*, Springer-Verlag, New York.

3. M. Rahnema, "Overview of the GSM System and Protocol Architecture," *IEEE Comm.*, Apr. 1993, pp. 92–100.

4. U. Leonhardt, *Supporting Location-Awareness in Open Distributed Systems*, doctoral dissertation, Dept. of Computing, Imperial College London, 1998.

5. European Commission, Directive 95/46/ec, *Official J. L 281,* 23 Oct. 1995, pp. 31–50; http://europa.eu.int/eur-lex/en/lif/dat/1995/en_395L0046.html.

6. M. Langheinrich, "Privacy by Design—Principles for Privacy-Aware Ubiquitous Systems," *Ubicomp 2001: Ubiquitous Computing*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2001, pp. 273–291.

7. M. Addlesee et al., "Implementing a Sentient Computing System," *Computer*, vol. 34, no. 8, Aug. 2001, pp. 50–56.

8. K. Cheverst et al., "Experiences of Developing and Deploying a Context-Aware Tourist Guide: The Lancaster GUIDE Project," *Proc. ACM MobiCom 00*, ACM Press, New York, 2000. pp. 20–31.

9. G.D. Abowd, "Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment," *IBM Systems J.*, vol. 38, no. 4, 1999, pp. 508–530.

10. M. Beigl, H.W. Gellersen, and A. Schmidt, "Mediacups: Experience with Design and Use of Computer-Augmented Everyday Objects," *Computer Networks*, vol. 35, no. 4, Mar. 2001, pp. 401–409.

11. B.N. Schilit, N.I. Adams, and R. Want. "Context-Aware Computing Applications," *Proc. IEEE Workshop on Mobile Computer Systems and Applications*, IEEE CS Press, Los Alamitos, Calif., 1994, pp. 85–90.

12. R. Want et al., "An Overview of the ParcTab Ubiquitous Computing Experiment," *IEEE Personal Comm.*, vol. 2, no. 6, Dec. 1995, pp. 28–43.

13. R. Want, "Ten Lessons Learned about Ubiquitous Computing," PowerPoint presentation from invited keynote (at the Dagstuhl Seminar on Ubiquitous Computing), Sept. 2001; www.inf.ethz.ch/vs/events/dag2001/slides/roy-lessons.pdf.

14. R. Want et al., "The Active Badge Location System," *ACM Trans. Information Systems*, vol. 10, no. 1, Jan. 1992, pp. 91–102.

15. A. Harter and A. Hopper, "A Distributed Location System for the Active Office," *IEEE Network*, vol. 8, no. 1, Jan./Feb. 1994, pp. 62–70.

16. F. Bennett, T. Richardson, and A. Harter, "Teleporting—Making Applications Mobile," *Proc. IEEE Workshop on Mobile Computer Systems and Applications*, IEEE CS Press, Los Alamitos, Calif., 1994, pp. 82–84.

17. R. Want, M. Weiser, and E. Mynatt, "Activating Everyday Objects," *DARPA/NIST Workshop on Smart Spaces*, 1998; sandbox.xerox.com/want/papers/aeo-nist-jul98.pdf.

18. A. Harter et al., "The Anatomy of a Context-Aware Application," *Proc. 5th Ann. ACM/IEEE Int'l Conf. Mobile Computing and Networking* (MobiCom 99), ACM Press, New York, 1999, pp. 59–68.

19. S. Long et al., "Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study," *Proc. 2nd Intl. Conf. Mobile Computing and Networking* (MobiCom 96), ACM Press, New York, 1996, pp. 97–107.

20. S. Feiner et al., "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment," *Personal Technologies*, vol. 1, no. 4, Dec. 1997, pp. 208–217.

21. N. Davies et al., "Caches in the Air: Disseminating Information in the Guide System," *Proc. 2nd IEEE Workshop Mobile Computing Systems and Applications* (WMCSA 99), IEEE Press, Piscataway, N.J., 1999, pp. 11–19.

22. K. Cheverst et al., "Developing a Context-Aware Electronic Tourist Guide: Some Issues and Experiences," *Proc. 2000 Conf. Human Factors in Computing Systems* (CHI 2000), ACM Press, New York, 2000, pp. 17–24.

23. J. Mankoff, S.E. Hudson, and G.D. Abowd, "Interaction Techniques for Ambiguity Resolution in Recognition-Based Interfaces," *Proc. UIST, 13th Ann. ACM Symp. User Interface Software and Technology*, ACM Press, New York, 2000, pp. 11–20.

24. A.V. Gershman, J.F. McCarthy, and A.E. Fano, "Situated Computing: Bridging the Gap between Intention and Action," *Proc. 3rd Int'l Symp. Wearable Computing* (ISWC 99), IEEE Press, Piscataway, N.J., 1999, pp. 3–9.

25. T. Kindberg et al., "People, Places, Things: Web Presence for the Real World," *3rd Int'l IEEE Workshop Mobile Computing Systems and Applications* IEEE CS Press, Los Alamitos, Calif., 2000.

26. M. Weiser and J. Brown, "Designing Calm Technology," *PowerGrid J.*, vol. 1.01, July 1996.

27. R. Want et al., "Bridging Real and Virtual Worlds with Electronic Tags," *Proc. ACM SIGCHI*, ACM Press, New York, 1999, pp. 370–377.

28. *The Common Object Request Broker: Architecture and Specification*, Revision 2.5., The Object Management Group, Needham, Mass., Sept. 2001.

29. *Jini Architectural Overview*, white paper, Sun Microsystems, Palo Alto, Calif., 1999.

30. A. Friday, N. Davies, and E. Catterall, "Supporting Service Discovery, Querying and Interaction in Ubiquitous Computing Environments," *Proc. 2nd ACM Int'l Workshop Data Engineering for Wireless and Mobile Access* (MobiDE 2001), ACM Press, New York, 2001.

31. M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *Personal Comm.*, vol. 8, no. 4, Aug. 2001, pp. 10–17.

32. R. Katz, "Adaptation and Mobility in Wireless Information Systems," *IEEE Personal Comm.*, vol. 1, no. 1, 1st Quarter 1994, pp. 6–17.

33. M. Roman, F. Kon, and R.H. Campbell, "Reflective Middleware: From Your Desk to Your Hand," *IEEE Distributed Systems Online*, vol. 2, no. 5, 2001; http://computer.org/dsonline/0105/features/rom0105_print.htm.

For more information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

the AUTHORS

Nigel Davies is an associate professor at the University of Arizona and a professor of computer science at Lancaster University. He has participated actively in the mobile computing research community. He has managed numerous projects at Lancaster, and he helped create its Mobile Computing group. In recognition of his work in establishing Lancaster as a major research center for mobile computing, he was awarded a personal chair in the Computing Department in 2000. He holds a BSc and PhD in computer science, both from Lancaster University, UK. He is an associate editor in chief of *IEEE Pervasive Computing* and an associate editor of *IEEE Transactions on Mobile Computing*. Contact him at the Dept. of Computer Science, Univ. of Arizona, Gould-Simpson 721, PO Box 210077, Tucson, AZ 85721-0077; nigel@cs.arizona.edu.

Hans-Werner Gellersen is professor for interactive systems at Lancaster University. He obtained an MSc and PhD in computer science from the University of Karlsruhe. His research interest is in ubiquitous computing and context-aware systems. He is actively involved in the formation of a ubiquitous computing research community, has initiated the HUC/Ubicomp conference series, and leads European research collaborations in the Commission's program on future and emerging technologies. He is editor of *Personal and Ubiquitous Computing*. Contact him at the Computing Department, Lancaster University, Bailrigg, Lancaster LA1 4YR, UK; hwg@comp.lancs.ac.uk.