

Software Infrastructure and Design Challenges for Ubiquitous Computing Applications

Striving to integrate computing into everyday activities in a seamless manner.

IN TRADITIONAL COMPUTING ENVIRONMENTS, USERS *actively choose to interact with computers. Ubiquitous computing applications are likely to be different—they will be embedded in the users' physical environments and integrate seamlessly with their everyday tasks. This vision leads to a set of defining characteristics, requirements, and research challenges for ubiquitous applications. This article identifies some of the key characteristics via a possible real-world scenario and derives the important application design and software infrastructure challenges that must be addressed by the computing research community.*

Consider the following scenario: Jane is the CIO of an organization that relies on computer services and is attending an important meeting in her organization's headquarters in New York City. She is in a conference room with three coworkers and two remote participants who have joined via a computer-based videoconferencing system. The system allows her team to see and hear the remote participants as well use a shared whiteboard space to jointly edit documents and explore data. Unfortunately, Jane has to leave the meeting early, as she has to catch a plane to visit a supplier in Paris. Luckily, she can continue participating in

the meeting via her smart PDA. As soon as it detects that she left the conference room, it routes the audio part of the meeting to her cell phone. As she enters the limousine that is taking her to the airport, the screen built into the back of the driver's seat displays the video stream of the meeting, including the people as well as the shared whiteboard space. Furthermore, the audio portion of the meeting is transferred to the limousine's in-car speakerphone system. As the car starts to move, the video adapts itself automatically to changing network quality by changing its resolution.

As the meeting progresses, an instant message pops up on

the screen informing her of a fire at the organization's data center in Colorado Springs. She taps the message on the touch-sensitive screen in order to dial the sender's phone number. The sender, the data center manager, reports that the fire at the data center severely damaged some important servers, but fortunately the backups are safely located off site. Jane decides to change her plans and fly to Colorado Springs immediately to assist with the repairs. She asks the data center manager to send her a detailed damage report in order to plan the repair. She activates her software personal agent using the in-car keyboard and tells the agent to change her flight reservations from Paris to Colorado Springs. She also requests that all the files regarding the data center configuration be transferred to her PDA as soon as the required network bandwidth is available. The agent rebooks the flights. In addition, it infers that she will not be able to fulfill her appointments in Paris, so it cancels the hotel room reservation using the hotel's online services and the meetings with the supplier by sending the appropriate email notification. As Jane leaves the limo, the personal agent "hops" from the car's computer to the PDA and scans the area for a better network connection. As she enters the airport terminal, the agent detects a wireless LAN to which it authenticates itself and starts downloading the requested data about the data center.

While waiting at the terminal, Jane calls the data center manager to get the latest update. Once she boards the plane, her PDA again scans for possible services in her environment and detects that it can connect to the screen and keyboard built into the plane's seat using an in-airplane wireless network. It displays Jane's desktop environment on the screen and highlights the data center information. After the plane takes off, Jane carefully plans the necessary repairs, puts in express orders for the needed parts, and enters requests for freelancers with the appropriate skills to relieve her own employees who are working around the

clock. The PDA's personal agent automatically prioritizes her requests and transfers some files over the slower airplane network. As she lands in Colorado Springs, it can transfer the remaining files using the terminal's wireless LAN. By the time she arrives at the data center, the first bids by freelancers have already arrived, and the computer equipment providers have received the orders for the replacement parts.

This scenario illustrates many aspects of ubiquitous computing, three key characteristics of which we highlight here.

Task Dynamism. Ubiquitous computing applications, by virtue of being available everywhere at all times, will have to adapt to the dynamism of users' environments and the resulting uncertainties. In these environments, users may serendipitously change their goals or adapt their actions to a changing environment [7]. In our scenario, for example, new information about the data center fire arrived unexpectedly, thus changing the actions Jane wanted to take, such as choosing an alternative destination. This requires programs that dynamically adapt to changes in either the goals or the plan structure by which those goals were to be achieved [3]. Sometimes the user might actively reconfigure the system to adapt to the new task settings; at other times the system might have to infer from its sensory input that the user changed his or her mind. Applications will, furthermore, have to be able to explain why they inferred those task changes and learn from their right and wrong inferences.

Device Heterogeneity and Resource Constraints. The omnipresence of ubiquitous applications is typically achieved by either making the technological artifacts (devices) move with the user or by having the applications move between devices tracking the user. In both cases, applications have to adapt to changing technological capabilities in their environment.

If the device itself is mobile (following the user or being carried around by him/her) then it usually has some con-

Once instantiated, applications may have to move seamlessly from one device to another and from one environment to another based on the user's activity.

straints in terms of physical dimensions. These physical constraints limit resources, such as battery power, screen size, networking bandwidth, and so forth. A PDA, for example, has relatively little usable screen area and limited battery power; a cell phone has an even smaller screen size but typically a longer battery life and is at least connected to a network. Furthermore, applications might also experience variability in the availability of resources, as demonstrated in our scenario. Obviously those limitations influence the development of applications and their capabilities.

The second approach to mobility is having the application follow the user and move seamlessly between devices. Applications will thus have to adapt to changing hardware capabilities (different types of pointing devices, keyboards, network types, and so on) and variability in the available software services. In our example scenario, Jane moved from the office (with a PC) to the car (with a more limited handheld and in-car computer), along with different user interfaces and networking capabilities in these two locations. Some resources might even become completely unavailable, for example, because the limo might not have a printer. The meeting software will have to dynamically adapt to changing technological capabilities by adapting the compression rate and size of transmitted pictures or determining an alternative output to a printout.

Computing in a Social Environment. Another major characteristic of ubiquitous computing technology is that it has a significant impact on the social environments in which it is used. Any introduction of a ubiquitous computing environment implies the introduction of sensors, which irrevocably have an impact on the social structure, no matter how unobtrusive they seem to be.

Imagine, for example, that your residence is outfitted with all kinds of sensors to provide information to a ubiquitous computing system. Would you want your neighborhood police station to be able to monitor which room you currently occupy (as indicated by the alarm system's motion detectors) and how much alcohol you are consuming (as inferred from your food inventory system)?

There are also policy questions: Who owns the

data from a ubiquitous computing system? How can we avoid making people feel like they are in an information panopticon [9]? Can one subpoena the data collected by ubiquitous computing systems? Since the answer is probably yes, there might be demand for ubiquitous computing systems in which the raw sensor data cannot be accessed at all, but processed inferences from the data, such as "burglar entry," can.

Research Challenges

Given these characteristics, we identify a number of challenges for the computing research community, organized into four broad categories.

Semantic Modeling. A fundamental necessity for an adaptable and composable computing environment is the ability to describe the preferences of users and the relevant characteristics of computing components using a high-level semantic model. For example, if we have information that a user prefers aisle seats on an airplane and that the user's device supports voice interaction, then a travel application can ask whether the user would like to have an aisle seat as usual. Ontologies [4] can be used to describe users' task environments, as well as their goals, to enable reasoning about a user's needs and therefore to dynamically adapt to changes. Furthermore, descriptions of the device capabilities and their appropriate use will allow applications to reason about how to best support users in any given context. For example, the system could down-sample pictures when network bandwidth decreases on a device. Or, the system could determine how to most appropriately interact with users given a social context by, for example, deciding not to produce an audible ring on the arrival of a new message during an opera but only to vibrate.

The research challenges in semantic modeling include developing a modeling language to express the rich and complex nature of ontologies, developing and validating ontologies for various domains of user activity, and finally, agreeing on shared ontology parts or translations for each application domain [2].

Building the Software Infrastructure. An effective software infrastructure for running ubiquitous applications must be capable of finding, adapting,

and delivering the appropriate applications to the user's computing environment based on the user's context. Semantic models for users, applications, and tasks will be used for this purpose. Such an infrastructure can support task dynamism as well as heterogeneous and resource-constrained devices.

The first challenge here is for the system to determine which user tasks are most relevant to a user in a particular context. They may be determined based on history, on preferences, or other knowledge of the user's behavior, as well as the environmental conditions. Once the user has selected a task from the list of relevant tasks, an application must be synthesized to suit the task, by finding and composing the appropriate components and services. The application should then be instantiated by potentially partitioning application components among various devices to take advantage of the relative strengths.

Once instantiated, applications may have to move seamlessly from one device to another and from one environment to another based on the user's activity, as described in our scenario. Applications must provide reasonable functionality, even when network connectivity is intermittent or unavailable, and must recover gracefully from failures. Finally, the software infrastructure must be scalable to support the large number of devices and applications that ubiquitous computing will enable.

Developing and Configuring Applications.

The current trend in the software industry is for service providers to provide reusable functions via application components called services. These services are described using a standard description language, and in the future, using standard ontologies. Such semantic descriptions could enable automatic composition of services, which in turn enables an infrastructure that dynamically adapts to tasks. Building applications will involve specifying the right composition of services, building a user inter-

face, and orchestrating the data flow among the various components.

The requirements of pervasive computing impose a significant shift in the developer's mindset while building pervasive applications [1]. Applications are not pieces of software targeted to a particular device

Project	Vision/Goals	Challenges Addressed
Aware home (Georgia Tech) www.cc.gatech.edu/fce/ahri	To develop the requisite technologies to create a home environment that can both perceive and assist its occupants.	User experience enhancement and validation
Aura (CMU) www.cs.cmu.edu/~aura/	To fundamentally rethink system design to address the problem of reduced user effectiveness due to explicit and implicit distraction from computers. To provide each user with an invisible halo of computing and information services that persists regardless of location.	Software infrastructure; Service construction and composition; and User experience validation
Cooltown (HP) www.cooltown.hp.com	A technology future where people, places, and things are first-class citizens of the connected world, wired and wireless—a place where e-services meet the physical world, where humans are mobile, devices and services are federated and context-aware, and everything has a Web presence.	Service construction and composition; User experience enhancement and validation
Jini (Sun) www.sun.com/jini	An open architecture that enables developers to create network-centric services—whether implemented in hardware or software—that are highly adaptive to change.	Context-based adaptation
Oxygen (MIT) oxygen.lcs.mit.edu/	To bring an abundance of computation and communication to users through natural spoken and visual interfaces, making it easy for them to collaborate, access knowledge, and automate repetitive tasks.	User experience enhancement and validation
PIMA (IBM Research) www.research.ibm.com/PIMA/	An application model and tools for developing applications in a platform-independent manner and for adapting them to multiple heterogeneous device platforms.	Application development and deployment; Context-based adaptation
Portolano (University of Washington) portolano.cs.washington.edu/	To create a testbed for investigation into ubiquitous task-specific computing devices, which are so highly optimized to particular tasks they blend into the world and require little technical knowledge on the part of their users.	Software infrastructure; Service construction and composition; User experience enhancement and validation
Semantic Web (W3C) www.semanticweb.org	An extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.	Semantic modeling

Ongoing projects addressing challenges.

or a particular environment but rather, high-level descriptions of the task a user needs to perform. The challenge is to be able to specify the interaction logic at an “intent-level,” and the application's requirements on data and computation. This kind of specification allows the infrastructure to adapt the application to heterogeneous devices with constrained resources.

Building reusable services will also be different for

ubiquitous environments. Not only will services encapsulate business logic and data but also “information interfaces” to physical artifacts, such as sensors and actuators. Finally, configuring services and applications so they can be easily and reliably reused by other developers and composed into larger applications will be a major challenge.

Validating the User Experience. As argued previously, pervasive computing has the potential to fundamentally change the way people use computing devices to perform their tasks. The utility of some computing advancements cannot be evaluated without performing significant user studies and in some cases, widely deploying it. Consequently, the development of effective methods for testing and evaluating the usage scenarios enabled by pervasive applications is an important area that needs more attention from researchers. The dynamic nature of some of these usage situations, the device heterogeneity with its varying resource availability, and the widely varying social environments in which mobile computing solutions are being used are, however, complicating the development of such evaluation methods (see the article “Beyond Prototypes” in [6]). Traditional laboratory experiments might not work in many cases, since they don’t capture the rich nature of the usage environments. This will force researchers to use more complicated approaches,

such as field-based quasi-experiments or ethnographies. There are several ongoing projects that deal with various challenges discussed here—a sampling is shown in the accompanying table.

Conclusion

As Mark Weiser described in his seminal article [8], ubiquitous computing is about interconnected hardware and software that are so ubiquitous that no one notices their presence. This will enable people to focus on their tasks and on interacting with other people. This far-reaching vision is still far from our reach [6], and will require fundamental advances in semantic modeling, context-aware software infrastructure, application modeling and tools, and user experience validation. Going back to our scenario, in order to help Jane manage the fire, we need major advances in each of the challenge areas we identified (as well as in others we have not mentioned). Most important, all these advances must be integrated in a seamless manner into Jane’s life so she can use them without constantly worrying about either the underlying mechanics or social appropriateness. 

REFERENCES

1. Banavarf, G. et al. Challenges: An application model for pervasive computing. In *Proceedings of the ACM Conference on Mobile Computing and Communications (Mobicom)*, Boston, August 2000.
2. Berners-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. *Scientific American* (May 2001).
3. Bernstein, A. How can cooperative work tools support dynamic group processes? Bridging the specificity frontier. In *Proceedings of the Computer Supported Cooperative Work Conference*. ACM Press, 2000.
4. Gruber, T. R. A translation approach to portable ontologies. *Knowledge Acquisition* 5 2 (1993), 199–220.
5. Mobicom Challenges 1999/2000/2001. In *Proceedings of the ACM Conference on Mobile Computing and Communications (Mobicom)*, 1999, 2000, 2001.
6. Reaching for Weiser’s Vision. *IEEE Pervasive Computing Magazine* 1 (Jan–Mar. 2002)
7. Suchman, L.A. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, Cambridge, UK, 1987.
8. Weiser, M. The computer for the twenty-first century. *Scientific American* (Sept. 1991).
9. Zuboff, S. *In the Age of the Smart Machine: The Future of Work and Power*. Basic Books, New York, 1988.

GURUDUTH BANAVAR (banavar@us.ibm.com) is a research staff member manager at the IBM T.J. Watson Research Center in Hawthorne, NY.

ABRAHAM BERNSTEIN (avi@acm.org) is an assistant professor at New York University.

This article expands on the work of a working group, which also included: Mark Ackerman (University of Michigan), Mark Adler (Nokia Research), Prasert Kanawatanchai, Kalle Lyytinen, Sean McGann, Frank Merat, and Lin Zhao (all from Case Western Reserve University).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.