Introduction to This Special Issue on Context-Aware Computing

Thomas P. Moran
IBM Almaden Research Center

Paul Dourish
University of California, Irvine

To be published in *Human-Computer Interaction*, Volume 16, Numbers 2-3, 2001

> Pervasive Computing is a term for the strongly emerging trend toward:
>
> - Numerous, casually accessible, often invisible computing devices
> - Frequently mobile or embedded in the environment
> - Connected to an increasingly ubiquitous network infrastructure composed of a wired core and wireless edges
>
> [from the call for a conference on Pervasive Computing[1]]

For most of us, computing has been done during the last two decades on personal computer workstations and laptops. Interacting with computational artifacts and networked information has been largely a "desk experience." This is now changing in a big way …

Ubiquitous/Pervasive Computing

Computation is now packaged in a variety of devices. Smaller and lighter laptop/notebooks, as powerful as conventional personal computers, free us from the confines of the single desk. Specialized devices such as handheld personal organizers are portable enough to be with us all the time. Wireless technology allows devices to be fully interconnected with the electronic world. Cameras and VCRs are being supplanted by digital equivalents, while we increasingly listen to music on devices that are digital and solid-state. Cell phones are really networked computers. The distinction between communication and computational is blurring, not only in the devices, but also in the

[1] http://www.nist.gov/pc2001/.

variety of ways computation allows us to communicate, from email to chat to voice to video.

On a different scale, computation is also moving beyond personal devices. Wall-sized displays allow us to get and interact with information in an inherently social manner. This is the beginning of being able to support collaborative work among people in shared physical locations. Interconnected computing devices, large and small, along with various sensing technologies, from simple motion sensors to electronic tags to video cameras, are being used to make physical rooms and buildings "intelligent." Interaction with computation can soon be an "environmental" and communal experience rather than a just a virtual and private one. Through these developments, computation is invading the fabric of our personal and social activities and environments.

We are being carried in this direction by several related strands of research, beginning with Weiser's (1991) vision of "ubiquitous computing" (now often called "pervasive computing"). New research communities and programs have formed around the notions of "augmented reality" (Mackay et al., 1993) "tangible interfaces" (Ishii, 1997), "wearable computers" (Bass et al., 1997) "cooperative buildings" (Streitz, Konomi, and Burkhardt, 1998), and so on. An annual series of Ubiquitous Computing Conferences is being organized,[2] and the National Institute of Standards and Technology has sponsored conferences on Pervasive Computing.[3]

What these technologies have in common is that they move the site and style of interaction beyond the desktop[4] and into the larger real world where we live and act. For us as users, this is a boon. For us as designers, however, it presents many challenges. The desktop is a well-understood, well-controlled environment. The real world, however, is complex and dynamic. The design challenge, then, is to make computation useful in the myriad various situations that can be encountered in the real world – the ever-changing context of use.

Context-Aware Computing

Context is a powerful, and longstanding, concept in human-computer interaction. Interaction with computation is by explicit acts of communication (e.g., pointing to a menu item), and the context is implicit (e.g., default settings). Context can be used to interpret explicit acts, making communication much more efficient. Thus, by carefully embedding computing into the context of our lived activities, it can serve us with minimal effort on our part. Communication can be not only effortless, but also naturally fit in with our ongoing activities. Pushing this further, the actions we take are not even felt to be communication acts at all; rather, we just engaged in normal activities; and the computation becomes invisible (e.g., Norman, 1998).

[2] http://www.ubicomp.org/.
[3] http://www.nist.gov/pc2001/.
[4] We mean this in both senses: beyond the virtual desktop, i.e., the graphical user interface with its "desktop metaphor," and beyond the physical desktop where computing devices have been confined.

Several years ago there was a special issue of *Human-Computer Interaction* on *Context in Design* (Moran, 1994). It discussed the notion that the design of computing artifacts must take into account how people draw on and evolve social contexts to make the artifacts understandable, useful, and meaningful.

The notion of context is much more widely appreciated today. The term "context-aware computing" is commonly understood by those working in ubiquitous/pervasive computing, where it is felt that context is key in their efforts to disperse and enmesh computation into our lives. *Context* refers to the physical and social situation in which computational devices are embedded. One goal of context-aware computing is to acquire and utilize information about the context of a device to provide services that are appropriate to the particular people, place, time, events, etc. For example, a cell phone will always vibrate and never beep in a concert, if the system can know the location of the cell phone and the concert schedule. However, this is more than simply a question of gathering more and more contextual information about complex situations. More information is not necessarily more helpful. Further, gathering information about our activities intrudes on our privacy. Context information is useful only when it can be usefully interpreted, and it must be treated with sensitivity.

Context-awareness is fine in theory. The research issue is figuring out how to get it to work in practice. The problems for human-computer interaction, in particular, are significant ones. Context-aware computing completely redefines the basic notions of interface and interaction. Research questions abound: What role does context play in our everyday experience? How can this be extended to a technological domain? What can the computation really do for us? How can we interact with as an invisible presence and yet maintain adequate control? How can we feel both served and safe?

Structure of this Special Issue

This special issue began in mid 1999 in discussions with Gregory Abowd. He, Daniel Salber and Anind Dey had just published their first paper on the Context Toolkit. (Salber, Dey, and Abowd, 1999), and were contemplating a more complete treatment of their work in context-aware computing. We encouraged this paper as representing a significant milestone in the long program of work in ubiquitous computing at Georgia Tech.

The resulting article by **Dey, Abowd, and Salber** is the *anchor article* of this special issue. Dey et al. observe that there have been numerous context-aware efforts, which are mostly demonstrations of various technologies for sensing, capturing, and presenting, and interacting with information in the physical context of people's work activities and demonstrations of various applications on user mobility and location-dependent information. Their "conference assistant" scenario illustrates a mobile context-aware device that allows one, by physically moving from room to room, to connect to the information being presented in various conference rooms, to annotate the events as one attends them, and to keep track of where one's colleagues are and what events they are finding interesting. After the conference, one can use their activities at the conference as indices into the online conference recordings to create a trip report. The thrust of their work is to systemize the complex task of designing and developing such applications.

They first define context in a very general way as any information that describes the setting of the users' activities, with emphasis on the physical attributes: time, place, people, physical artifacts, and computational objects. They identify the difficulties of handling context due to the myriad of sensors and devices involved, its distributed nature, the need to interpret the data into useful abstractions, and the need for persistent storage and access of context information. They address these issues by a conceptual framework consisting of context "widgets" (for capture and interaction), interpreters and aggregators, services, and resource discoverers. This framework is embodied in their Context Toolkit. They then illustrate several applications and how the toolkit supports their implementation, and, in one case, how the framework can be the basis of a systematic software design process.

One of the main motivations for Dey et al.'s work is the need is to explore realistic applications, and they provide a toolkit to make that easier. However, even more than building an application, building a toolkit involves taking a stand on critical issues that go beyond technical design. Dey et al.'s article, then, does more than simply present a technology design; it opens up a conversation about the questions of context in general. What aspects of context matter, and what sorts of representational approaches will we use? How does context enter into the computational world, and how can computers incorporate contextual information? How do computational context and human context relate to each other?

In view of the broader questions raised by the anchor paper, we sought constructive input from a range of researchers and practitioners who, in their work, have been thinking about context and the issues that it raises. Context is a broad topic, and so the range of approaches represented by these authors is similarly broad. Contributors come from a wide range of disciplinary backgrounds – computer science, social science, human-computer interaction, various domains of design, cognitive science, and beyond. We invited them to use the anchor paper as a launching point for essays, outlining their perspective on context, context-awareness, and computation.

Roadmap Through the Essays

Our goal was to invite essayists who would, collectively, explore the landscape of issues surrounding context-aware computing. We did not set out explicit themes or areas for essays. Each essay represents a unique point of view. Thus, they occur in alphabetical order in this special issue. However, putting a little structure onto the collection of essays helps us better see the landscape. So, we here outline the set of essays as falling into six broad topic areas.

Software Architectures. Architecture is one of the primary topics of the anchor article, and this is reflected in a number of the essays that also address architectural themes. **Hong and Landay** explore the idea of context-aware computing from the perspective of a service infrastructure, a pervasive middleware approach in which much of the work of collecting and processing context information can be decoupled from the application itself. They argue that this approach provides a number of benefits that will be needed if context-aware computing is to be adopted on a broad scale. **Winograd**

compares different architectural approaches for building context-aware systems, and argues that a blackboard-based approach may have more flexibility than the traditional widget-based approach that the anchor paper develops. **Benerecetti, Bouquet, and Bonifacio** consider the consequences of applying context-aware computing more broadly, not simply to people interacting with their environment, but also to distributed agents interacting across a network. Like ubiquitous computing applications, distributed agent interactions depend take place in a rich context of previous actions, individual beliefs, invisible states and different perspectives about which agents must reason in order to interpret the settings in which they find themselves. By extending our notion of context-aware computing beyond the physical world, Benerecetti et al. attempt to move towards an interactionally-defined notion of context.

Foundations. In their essays, Dourish and Svanaes both appeal to phenomenology in order to develop foundational understandings of context-awareness. **Dourish** highlights the relationship between recent work in context-aware computing and the research conducted over the last ten years or so into social aspects of interactive technology, and shows how phenomenological understandings of human activity underpin both endeavors. **Svanaes** shows how these ideas can be used in an experimental framework to explore the experience of context-aware computing as encountered directly by users.

Design Principles. A number of essayists address themselves primarily to design audiences, and explore guidelines for effective context-based design solutions. They address different parts of the problem. **Bellotti and Edwards** focus in particular on the balance of agency and accountability between context-aware systems and their human users. **Greenberg** attempts to link practical design considerations to theoretical accounts of situated and context-based action. **Shafer, Brumitt, and Cadiz** use the idea of context-based computing in home environments to lay out some of the dimensions of the design space, and show how these issues are addressed in their current system development efforts.

The Built Environment. For a number of our essayists, the physical environment in which human activity takes place is especially important, because it is simultaneously the outcome of a design process (suggesting important lessons for the design of context-aware technologies) and a reflection of social norms (which will also affect new technologies). **Agre** puts forward a conceptual framework for understanding context which links the built environment, practice and institutional arrangements, and considers the role that new technologies will play on all three levels. **McCullough** draws on the role of type and typology in architectural design practice to illustrate the range of problems that context-aware computing will need to tackle, and to turn our attention away from computing per se, and towards the other activities in which computing may play a role. Finally, **Kirsh** draws on the distributed cognition framework to call our attention to the complex structural coupling between malleable environments and everyday activities, and considers the consequences for technologies that attempt to capture the richness of everyday interactions.

Domain Context. The architectural model that Dey et al. present is intentionally generic, intended to serve as the basis for a wide range of possible applications. Some of

the essays consider the particular demands of specific domain areas for context-aware computing. Exploring these demands serves two purposes – it richens our understanding of the role that context can play in interaction, and also tests generic design approaches with examples of specific models that they must capture. **Fischer** considers the domain of design environments and the role that tangible interfaces can play in giving design stakeholders a physically shared context to ground their activities. **McGee, Pavel, and Cohen** use a mixed-reality system in a military command setting to explore how different interaction modalities set a context for each other that can aid computational systems in disambiguating ambiguous input.

Captured Content of Context. Finally, some essays broadly tackle the issues of decontextualizing information. One problem in capturing contextual information is that it is typically *de*-contextualized in the process. When stored on a disk, information about activities is inherently less rich than the activities that it describes. This decontextualization leads to a number of problems concerning the control of information and over its interpretations. **Grudin** explores this issue directly. He discusses the mechanisms by which people maintain control over information and its contexts of interpretation, and shows how decontextualization has affected the adoption of other technologies, which may offer lessons for new developments. **Ackerman, Darrell, and Weitzner** consider some of the privacy implications of this kind of decontextualization, and discuss potential mechanisms for giving users information about and control over the system's disclosure of information about their actions. **Lucas** focuses in particular on the problems of identity, and highlights some problems in the way that current context-aware systems construe the issues of device and human identity.

Summary. These are broad themes, and many of the essays deal with many other points besides. Our intention in this introduction is simply to suggest some initial themes across the essays. The broad issues are still in the process of being developed, articulated, and understood. Context-aware computing offers us the opportunity to see these issues emerge and begin to tackle the theoretical and practical problems that result from these new technologies. We cannot be sure what solutions will emerge as we explore the issues. What we can be sure about, however, is that context-aware computing will increasingly become part of our lives and a fundamental feature of our interactive experience. The integration of computational into the everyday world is as important a topic for human-computer interaction researchers and practitioners as the spread of interactive computing and the globalization of networked information have been – and the problems are even tougher. This special issue seeks to bring the questions to our attention, and show both the range of current thinking about the issues.

NOTES

Authors' Present Addresses. Thomas Moran, IBM Almaden Research Center, 650 Harry Road, San Jose, CA, 95120. Email: moran@acm.org. Paul Dourish, Department of Information and Computer Science, University of California, Irvine, CA 92697-3425. Email: jpd@ics.uci.edu.

REFERENCES

Bass, L., Kasabach, C., Martin, R., Siewoirek, D., Smailagic, A., & Stivorik, J. (1997). The Design of a Wearable Computer. *Proceedings of the CHI'97 Conference on Human Factors in Computing Systems*, 139-146.

Ishii, H. & Ullmer, B. (1997). Tangible bits: Towards seamless interfaces between people, bits, and atoms. *Proceedings of the CHI'97 Conference on Human Factors in Computing Systems*, 234-241.

Mackay, W.E., Velay, G., Carter, K., Ma, C., & Pagani, D. (1993). Augmenting Reality: Adding Computational Dimensions to Paper. *Communications of the ACM*, 36, 7, 96-97.

Moran, T. P. (Ed.) (1994). Special Issue on context in design. *Human-Computer Interaction*, 9, 1-149.

Norman, D. A. (1998). *The invisible computer*. Cambridge, MA: MIT Press.

Salber D., Dey, A. K. & Abowd, G. D. (1999). The Context Toolkit: Aiding the development of context-enabled applications. *Proceedings of the CHI '99 Conference on Human Factors in Computer Systems*, 434-441.

Streitz, N. A., Konomi, S., Burkhardt, H-J. (Eds.) (1998). *Cooperative Buildings: Integrating information, organization, and architecture*. Berlin: Springer.

Weiser, M. (1991). The computer for the 21st century. *Scientific American*, *265* (September), 94-104.

ANCHOR ARTICLE FOR THIS SPECIAL ISSUE

Dey, A. K., D., Abowd, G. D., & Salber, D. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, *16*, xxx-xxx.

ESSAYS IN THIS SPECIAL ISSUE

Ackerman, M., Darrell, T., & Weitzner, D. J. (2001). Privacy in context. *Human-Computer Interaction*, *16*, xxx-xxx.

Agre, P. E. (2001). Changing places: Contexts of awareness in computing. *Human-Computer Interaction*, *16*, xxx-xxx.

Bellotti, V., & Edwards, K. (2001). Intelligibility and accountability: Human considerations in context aware systems. *Human-Computer Interaction*, *16*, xxx-xxx.

Benerecetti, M., Bouquet, P., & Bonifacio, M. (2001). Distributed context-aware systems. *Human-Computer Interaction*, *16*, xxx-xxx.

Dourish, P. (2001). Seeking a foundation for context-aware computing. *Human-Computer Interaction*, *16*, xxx-xxx.

Fischer, G. (2001). Articulating the task at hand and making information relevant to it. *Human-Computer Interaction*, *16*, xxx-xxx.

Greenberg, S. (2001). Context as a dynamic construct. *Human-Computer Interaction*, *16*, xxx-xxx.

Grudin, J. (2001). Desituated action: Digital representation of context. *Human-Computer Interaction*, *16*, xxx-xxx.

Hong, J. I., & Landay, J. A. (2001). An infrastructure approach to context-aware computing. *Human-Computer Interaction*, *16*, xxx-xxx.

Kirsh, D. (2001). The context of work. *Human-Computer Interaction*, *16*, xxx-xxx.

Lucas, P. (2001). Mobile devices and mobile data—Issues of identity and reference. *Human-Computer Interaction*, *16*, xxx-xxx.

McCullough, M. (2001). On typologies of situated interaction. *Human-Computer Interaction*, *16*, xxx-xxx.

McGee, D. R., Pavel, M., & Cohen, P. R. (2001). Context shifts: Extending the meanings of physical objects with language. *Human-Computer Interaction*, *16*, xxx-xxx.

Shafer, S. A. N., Brumitt, B., & Cadiz, JJ. (2001). Interaction issues in context-aware intelligent environments. *Human-Computer Interaction*, *16*, xxx-xxx.

Svanaes, D. (2001). Context-aware technology: A phenomenological perspective. *Human-Computer Interaction*, *16*, xxx-xxx.

Winograd, T. (2001). Architectures for context. *Human-Computer Interaction*, *16*, xxx-xxx.