Report from the Ubicomp Education Workshop

Gregory D. Abowd, Gaetano Borriello, and Gerd Kortuem

EDITOR'S INTRODUCTION

This issue's Education & Training column departs somewhat from the norm. Rather than presenting information about a specific class, the authors discuss, more broadly, the status and future directions for pervasive and ubiquitous computing education. The column summarizes the Ubicomp Education Workshop, which the authors chaired, held in conjunction with UbiComp 2003. We hope the column will serve as a foundation for future discussions on pervasive and ubiquitous computing education, such as at the Workshop on Pervasive Computing Education held in conjunction with PerCom 2004 (www.percom.org). Please let me know your comments and suggestions for future columns. *—Scott Midkiff*

As interest and activity in ubiquitous computing continues to grow, we need to reflect as a community on how we're codifying this subject's intellectual core. At the Ubicomp Education Workshop (see the related sidebar), a small number of researchers and educators gathered to discuss how ubiquitous computing is taught at universities and whether we can define common themes and resources to influence the development of future educational materials. We aimed to uncover common themes, gather advice for best practices, and seek ways to produce resources to share with the larger ubiquitous computing community.

The one-day workshop was conducted in a fairly informal fashion, beginning with each individual describing his or her experience in teaching ubiquitous computing courses. The remainder of the workshop involved collective brainstorming on common themes and practices of ubiquitous computing education. We noted a considerable variety in the kinds of courses being offered, including one-week block courses, reading seminars, traditional lecture courses, and open-ended project courses. Despite this variety, some common themes emerged during the discussions, which we'll briefly summarize here.

HOW TO TEACH AND WHEN?

The topics of interest in ubiquitous computing go beyond or extend many of the core topics in computer science, including distributed systems, networking, databases, architecture, programming languages, human-computer interaction, software engineering, and artificial intelligence. Ubiquitous computing also raises many interesting challenges that lie outside of traditional computer science, such as socioeconomic impact, privacy, and legal issues. We must ask whether to teach ubiquitous computing as a collection of specialized modules that you can teach within the context of existing core classes or as one or more additional courses specifically designed to explore the subject's breadth and depth. Most universities teach ubiquitous computing as self-contained courses, either as senior-level undergraduate specialization courses or as introductory graduate courses.

Although computing or computer science departments typically offer these courses, the background of students interested in ubiquitous computing varies substantially, particularly at the graduate level, where specialized master's courses can attract students with diverse technical and design backgrounds. While this enhances the educational experience, particularly with group project activity, it also increases the challenge of developing educational materials that will reach the entire audience.

Workshop attendees described how they've taught ubiquitous computing in the past:

- Brief, intense one-week courses for graduate students or industry, covering general lecture or tutorial material as well as hands-on design and implementation projects
- Traditional lecture courses of 10 to 16 weeks, usually accompanied by group projects
- Open-ended project courses, in which an instructor facilitates small groups throughout a lifecycle of concept development to prototype deployment with ubiquitous computing technologies
- Technology-centric development

Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/04/$20.00 © 2004 IEEE

courses, in which elements of ubiquitous computing technologies are explained and integrated in a well-planned, hands-on case study
- Reading seminars that discuss selected literature in the area

THE INTELLECTUAL CORE

One of the first challenges anyone who wishes to teach a ubiquitous computing course faces is how to organize a syllabus of topics to cover. Although many courses assign Mark Weiser's seminal *Scientific American* article (reprinted in the January–March 2002 issue of *IEEE Pervasive Computing*) as the first reading, the path after that is less clear. A major outcome of this workshop is a proposed set of five themes for an introductory course in ubiquitous computing.

Theme 1: Physical and virtual integration

Much of ubiquitous computing's intrigue deals with merging the physical and virtual worlds. Important subtopics include

- *Sensing and actuation*. Students should be introduced to ways to gather information about the physical world and represent it in the virtual world, as well as how to take decisions made in the virtual world and make them apparent in the physical world.
- *Awareness and perception*. Students should learn how to extend the notion of sensing and how to create higher-level knowledge of the physical world.
- *Ambient displays*. Students should see how to use the physical world's features to reflect information from the virtual world.
- *World modeling*. Students should learn to create representations and ontologies of the physical world that they can represent virtually.

Theme 2: Interaction models

The human experience with ubiquitous computing is important for students to understand and design. This topic deals with formative ways to produce the right human experience. Important subtopics include

- *Situated computing*. Ubiquitous computing is woven into everyday activities, so it's important for students to understand the relationship between technologies and everyday life.
- *Multimodality and natural interaction*. Ubiquitous computing encourages numerous modes of interaction, so students should understand both explicit interaction, such as speech, gestures, and writing, and implicit interaction, such as location and user activity.
- *Multidevice*. We can view the age of ubiquitous computing as one in which users interact with many devices, both shared and personal. Students should be exposed to the many form factors of devices that provide the ways to interact.
- *Invisibility*. The essence of Mark Weiser's vision was that the proliferation of computing results in a decrease in the cognitive demand on the individual. Students must understand that the invisibility doesn't mean the technology shrinks but that its pull on our consciousness decreases.
- *Disambiguation*. Sensing and perception aren't perfect, so students should learn techniques to represent and reconcile imperfect knowledge for consumption by both humans and software.
- *Proactivity*. The profusion of computational devices envisioned by ubiquitous computing raises the question of effective user control. Students should be familiar with both traditional models of user-initiated interaction—that is, direct manipulation—as well as proactive or mixed-initiative interfaces.
- *Models of cognition*. Models of individual cognition drive much of traditional human-computer interaction practice. To design for ubiquitous computing, students must understand extensions to traditional models that consider behaviors influenced by the physical world.

Theme 3: System components

Emerging technology provides the building blocks for creating ubiquitous computing systems. These components represent a system's hardware and software aspects. Important subtopics include

- *Platforms*. Students must learn about devices or platforms with small form factors that are highly mobile and attached to an individual, as well as those that you can easily embed into the environment.
- *Sensors and actuators*. Students should understand the importance of interpreting the physical world (sens-

ABOUT THE WORKSHOP

The workshop on "Ubicomp Education: Current Status and Future Directions" was held on 12 October 2003 in conjunction with the International Conference on Ubiquitous Computing (UbiComp 2003) in Seattle (http://ubicomp.lancs.ac.uk/workshops/education03). The participants were

- Gregory Abowd, Georgia Tech
- Gaetano Borriello, University of Washington and Intel Research Seattle
- Nigel Davies, Lancaster University and University of Arizona
- Alois Ferscha, University of Linz
- Hans Gellersen, Lancaster University
- Gerd Kortuem, Lancaster University
- Peter Ljungstrand, Chalmers University
- Cristina Videira Lopes, University of California, Irvine
- Bill Plymale, Virginia Tech
- Larry Rudolph, Massachusetts Institute of Technology

EDUCATION & TRAINING

CASE STUDIES DISCUSSED AT THE WORKSHOP

Active Badge (Cambridge Univ.)—www.uk.research.att.com/ab.html
Active Campus (UC San Diego)—http://activecampus.ucsd.edu
ActivityCompass (Univ. of Washington)—www.cs.washington.edu/homes/djp3/AI/AssistedCognition/ActivityCompass
Aware Home (Georgia Tech)—www.cc.gatech.edu/fce/ahri
Cooltown (HP)—http://cooltown.hp.com/cooltownhome
Digital desk (Univ. of Toronto)
eClass/Classroom2000 (Georgia Tech)—www.cc.gatech.edu/fce/eclass
Equator projects, including Uncle Roy All Around You, Can You See Me Now?, and Ambient Wood Guide (various universities)—www.equator.ac.uk
iRoom (Stanford Univ.)—http://iwork.stanford.edu
Labscape (Intel/Univ. of Washington)—http://labscape.cs.washington.edu
MediaCup (TecO/Univ. of Karlsruhe)—http://mediacup.teco.edu
MusicFX (Accenture)—www.accenture.com
NeuralNetwork Home (Univ. of Colorado)
PARCTab (Xerox Parc)—www.ubiq.com/parctab
PlantCare (Intel)—http://seattleweb.intel-research.net/projects/plantcare
Remembrance Agent (MIT Media Lab)—www.remem.org
Smart-Its (Lancaster Univ.)—www.comp.lancs.ac.uk/eis
Smart Kindergarten (UCLA)—http://mediacup.teco.edu

ing) and affecting it (actuation). The actual devices that perform both sensing and actuation are this subtopic's focus.

- *Power awareness*. The proliferation of wireless and highly mobile computational platforms brings an increased concern for delivery of power to these devices. In addition to advances in battery technology, students should also learn about the design of power-aware computation.
- *Communications technologies and connectivity*. Highly distributed computational platforms are more useful when they can communicate. Courses in ubiquitous computing should cover advances in both wired and wireless communications technologies.
- *Software architectures and technologies*. The wide variety of computing platforms is matched by the many software technologies used to create ubiquitous computing applications. As is often the case for complex system development, the complexity lies not in the individual components but in their integration into a functioning system. Ubiquitous computing courses must address many software challenges, including architectural approaches, naming schemes, resource discovery, and event notification. Many of these are software engineering challenges for ubiquitous computing.

Theme 4: Deployment

Putting applications into the real world raises many challenges associated with the summative understanding of ubiquitous computing. Important subtopics include

- *Scalability*. Students should understand that ubiquitous computing is characterized by applications that address scale in terms of space, people, and devices.
- *Reliability*. Students should learn to address reliability issues, because applications in the real world must work, even when constituent parts fail.
- *Maintenance*. Students should learn principles of maintenance because in areas where technologies advance rapidly, it's particularly important to evolve applications as underlying technologies change.
- *Evaluation*. Traditional forms of evaluation from the field of human-computer interaction often fail when applied to ubiquitous computing systems. Students should learn techniques applied "in the wild" and not just controlled laboratory user studies.
- *Practicality*. Students need to understand various practical concerns for daily system operation, including cost and convenience.

Theme 5: Social concerns

Ubiquitous computing promises to impact even more of our everyday lives than personal computing, and this brings consequent concerns about how the technology and services best serve social needs. Technology itself is only one factor that influences or controls the relationship between people and ubiquitous computing. Important subtopics include

- *Socioeconomics*. Students should appreciate that market forces can influence system deployment and sustainability. Students should appreciate that the way in which a technology is perceived to impact social interactions can greatly influence adoption and long-term use.
- *Legal issues*. Concerns that suitable social, market, and technological forces cannot address are often left

to legal bodies. It's important for students to understand how national and international laws influence and are influenced by technology.

- *Privacy*—Integrating the physical and virtual worlds has huge implications on personal privacy, security, and autonomy. Students must understand how to respect the individual and see how ubiquitous computing influences an individual's power.

Although the workshop identified the core areas just discussed, the participants didn't determine any particular ordering of the topics, nor did they identify specific important results and readings in the different areas. Future examinations of ubiquitous computing education should identify seminal readings.

LEARNING THROUGH EXPERIENCE

Most workshop participants agreed that the best understanding of ubiquitous computing comes through experience. In the educational setting, this means exploring specific examples from the past, so-called case studies, or the hands-on development of ubiquitous computing applications done through individual or group projects.

Design projects

Most courses in ubiquitous computing involve hands-on experience with some of the technologies and applications. Teaching students how to structure projects to ensure successful learning outcomes has been a concern. With critical technologies—such as wireless connectivity, handheld computing, sensors and actuators, and recognition technologies—becoming more mainstream, it's easier to provide a relevant infrastructure for students to build projects.

The extent to which classes ask students to design as well as implement varies. Some courses consider the user-centered design and prototyping as a critical project component, even more important than building a real system. This kind of project encourages creative application of technologies, even when those technologies aren't available. Other courses emphasize construction with ubiquitous computing technologies, to the extent of ignoring user-centered design. Although these projects often might not provide compelling uses of technology, they teach students the importance of integration. The workshop didn't judge which is right or wrong. Both approaches have strengths and limitations, and the kind of project chosen will vary based on the students' backgrounds.

We also recognize that the available building block technologies often influence the applications that students can explore. Many courses leverage easily available commodity platforms, such as handheld computers. Computer science

EDUCATION & TRAINING

has a long tradition of sharing software technologies. Increasingly, we're seeing research labs produce hardware technologies that other researchers can use for exploration. Examples include Phidgets from the University of Calgary; Motes from the University of California, Berkeley, and Crossbow; and European Smart-Its.

Case studies

For case studies, a ubiquitous computing course could use the research community's growing list of interesting investigations of ubiquitous computing applications. The extent of documentation marks an important feature of a good case study. This includes retrospective articles on the project's history and lessons learned as well as artifacts that students can explore. It's also important to discuss case studies in terms of the core intellectual topics outlined earlier. Although it's unlikely that any single case study will exhibit important features from every intellectual topic, you might choose a case study on the basis of its coverage of those topics (see the related sidebar for case studies discussed at the workshop).

Building shared resources

Although it might be premature to suggest writing a textbook for this field, it's appropriate as a community for us to develop a means to share educational materials. Those who teach ubiquitous computing courses produce online repositories of readings, lectures, projects, and case studies. One recommendation from the workshop is to develop a single online repository or clearinghouse for gathering ubiquitous computing educational material. The most likely location for such a clearinghouse is ubicomp.org, the community's formal site for conferences and discussion forums.

The Ubicomp Education Workshop was intentionally informal because it was one of the first meetings to discuss education in ubiquitous computing. Several national and international forums for ubiquitous and pervasive computing now exist, such as the Workshop on Pervasive Computing Education (www.irean.vt.edu/PerEd2004) in March 2004. We strongly recommend that future forums take on the specific challenges we've raised to continue to advance the educational agenda for this important emerging topic.

Gregory D. Abowd is an associate professor in the College of Computing at the Georgia Institute of Technology. Contact him at abowd@cc.gatech.edu.

Gaetano Borriello is a professor of computer science and engineering at the University of Washington and a member of research staff at Intel Research Seattle. Contact him at gaetano@cs.washington.edu.

Gerd Kortuem is a lecturer in the Computing Department at Lancaster University. Contact him at kortuem@comp.lancs.ac.uk.