Education & Training

A Pervasive Computing Curriculum for Engineering and Science Students

Fei Hu and Ankur Teredesai

EDITOR'S INTRODUCTION

In this issue, Fei Hu and Ankur Teredesai of the Rochester Institute of Technology describe a set of courses on pervasive computing topics and a supporting laboratory designed for both computer engineering and computer science students. Please send me your comments and suggestions for future columns.
—Scott F. Midkiff

Today, there's an urgent need for well-trained pervasive computing engineers and scientists. Pervasive computing has evolved into a critical field, influencing a broad range of industries from healthcare to manufacturing to defense. However, most computer engineering and computer science programs pay marginal attention, at best, to the field.

At the Rochester Institute of Technology, we aim to enhance our students' pervasive computing skills. With the support of the US National Science Foundation and Cisco, we implemented a multidisciplinary Pervasive Computing Lab and have developed relevant courses for both CE and CS students.

PERVASIVE COMPUTING LABORATORY

In our pervasive computing curriculum, we established a multidisciplinary laboratory that offers educational elements to CE and CS programs (see figure 1). This includes a middleware framework that supports some typical applications such as sensor database management and sensor data mining. Computer engineering students can also learn about the integration of typical pervasive networks, including high-speed wireless local area networks (WLANs), mobile ad hoc networks (MANETs), RFID, and wireless sensor networks (WSNs). We also emphasize security issues in pervasive environments.

As Leonard Kleinrock pointed out in an interview conducted by James Kurose and Keith Ross in their book *Computer Networking: A Top-Down Approach Featuring the Internet* (Addison Wesley, 2004), pervasive computing builds on three components: nomadic computing, smart spaces, and mobile computing data management. Various wireless network devices in our lab give students hands-on experience with these components.

Nomadic computing

Wireless-technology advances have enabled portable devices to support many important nomadic, or mobile, applications. Our lab gives students access to Compaq iPAQ PDAs to set up ad hoc networks and Cisco WLAN devices such as base stations.

Sensor-based smart spaces

Smart spaces refer to environment-monitoring and control systems achieved by deploying tiny sensors and actuators. Applications include detecting fires, earthquakes, water pollution, and homeland security intrusions. Our curriculum uses sensor networks as the educational platform of smart spaces. The Crossbow classroom-oriented motes, with their pro-

QUICK FACTS

Courses: Data Management in Pervasive Computing (CECS 400), Principles of Wireless and Mobile Networks (CECS 500), and Pervasive Computing Architecture and Design (CECS 550)
Units: Computer Engineering and Computer Science Departments, Pervasive Computing Lab
Institution: Rochester Institute of Technology
Instructors: Fei Hu and Ankur Teredesai
Level: Upper-division undergraduate and graduate
URL: http://mycourses.rit.edu (please contact the authors for access)

Published by the IEEE Computer Society ■ 1536-1268/07/$25.00 © 2007 IEEE

grammable routing and MAC (media access control) stacks, help us provide smart spaces training. We also use readers (such as the SkyeRead M1-Mini) and tags for RFID teaching.

Mobile computing data management

For many years CS education focused on data management in traditional centralized databases. However, with the advent of pervasive computing, an ever-increasing amount of information is collected by distributed nomadic networks, and we believe students need instruction about them as well. For this component, we use Crossbow motes that we can program to run distributed databases. We also developed a series of lab assignments on WSN data organization and mining.

COURSES

We developed our courses (see the "Course Topics" sidebar) to correspond with Kleinrock's three pervasive computing components. We aimed to make them a cohesive sequence, cover the benchmark courses' core topics, adhere to RIT's 10-week quarter system (we also developed 15-week course materials in case semester-based schools use our materials), and target upper-level undergraduate CE and CS majors (we focus on building solid foundations as well as on advanced topics):

- Data Management in Pervasive Computing (CECS 400) concentrates on data-mining issues and distributed database management in pervasive computing environments.
- Principles of Wireless and Mobile Networks (CECS 500) examines the principles of current nomadic-computing systems, including cellular networks and CDMA (code division multiple access), MANETs, GPS, and WLANs. The lab exercises include physical and MAC design using OPNET.
- Pervasive Computing Architecture and Design (CECS 550) discusses the system architecture, hardware

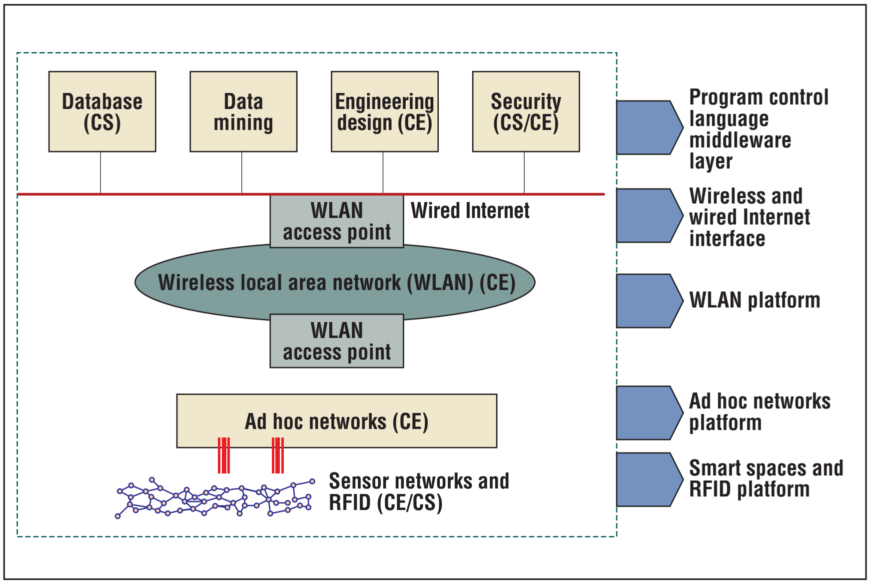

Figure 1. The Pervasive Computing Lab architecture.

COURSE TOPICS

Data Management in Pervasive Computing

- Primitive data types for sensor networks
- Effect of networking issues on data collection
- Data dissemination, quality of service, and data reliability
- Sensor databases: query processing and probabilistic queries
- Event-based data management and event detection in sensor networks
- Distributed data storage and indexing, and indexing implementations
- Database indexing in peer-to-peer networks versus sensor networks

Principles of Wireless and Mobile Networks

- Radio basics: path loss model, fading, and modulation and coding
- Wireless LAN: architecture and protocol, media access control protocol design, and security
- Ad hoc networks: routing, TCP (Transmission Control Protocol) improvement in MANETS (mobile ad hoc networks), and mobility
- Wireless sensor networks (WSNs): self-organization, topology, coverage, and hardware
- Cellular networks: call admission control, handoff, and CDMA (code division multiple access)
- Satellite networks and GPS
- Bluetooth: scatternets, frequency hopping, and products

Pervasive Computing Architecture and Design

- Digital signal processing for wireless communications: DSP review, wireless signal transmission, and receiving architecture
- Hardware for wireless communications: antenna design and diversity receivers
- Mobile-network architecture: system design of cellular networks and ad hoc networks, and Bluetooth and WLAN hardware
- Wireless security: applied cryptography and WSN and MANET security
- Smart spaces: sensor network and RFID hardware design
- Pervasive computing: software engineering principles

EDUCATION & TRAINING

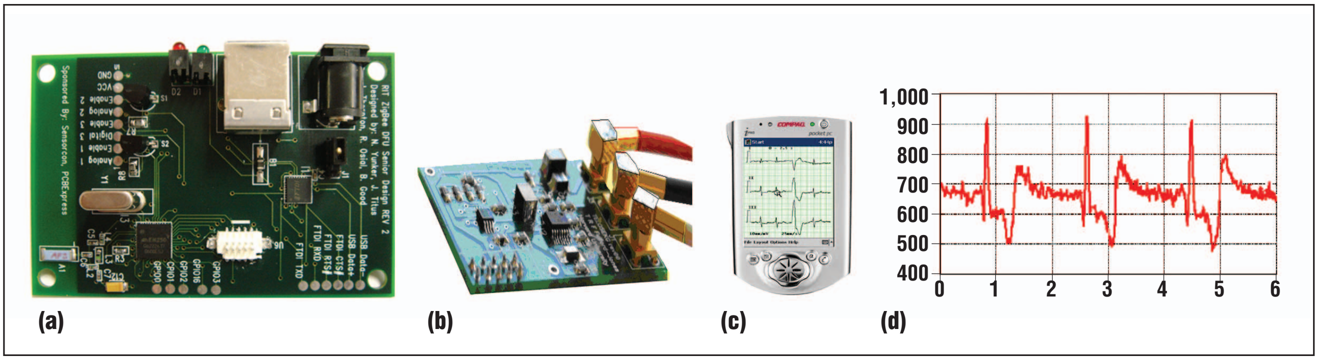

Figure 2. A medical-sensor-network senior project: (a) RF communication board, (b) echocardiogram sensor(created with help from the Harvard CodeBlue team), (c) PDA software to collect data from the sensors, and (d) ECG signals collected from the sensors.

components, smart spaces, network security, and software applications in pervasive computing platforms.

The target audience is CE and CS third-, fourth-, and fifth-year undergraduate students and master's students and PhD candidates. We offer these courses each year in separate quarters. Every year at least one class section of CE and CS undergraduate students (class size is approximately 40 students) take each of these courses. Each class has two lab sections (approximately 20 students) managed with the help of two to four lab teaching assistants. The lab TAs receive training before they begin helping students and must keep in close contact with instructors.

From our teaching experience with these courses, we learned these important lessons:

- Don't rely on PowerPoint slides. Students like interactive methods, such as using a whiteboard, to demonstrate their understanding of a topic. They digest material much better when actively taking notes from step-to-step demonstrations given by instructors than by sitting and watching slides.
- Always explain "why." For instance, tell them why we should use on-demand routing in highly mobile ad hoc networks instead of just teaching them AODV (ad hoc on-demand distance vector) details.
- Multiple, short quizzes work better than one or two long tests. Pervasive computing is a new field, and radio communication concepts (such as CDMA, OFDM [orthogonal frequency-division multiplexing], and coding) are abstract to beginners. We can push students to review materials more frequently and efficiently by giving more frequent, short quizzes.
- For master's and PhD students, we assign team-based hardware and software projects rather than term papers. We find term papers difficult to evaluate, and they don't motivate students to invest much effort in writing them because it's easy to find papers with similar topics already available.

CLASS LABS AND PROJECTS

A pervasive computing education requires hands-on experience because you can't give in-depth explanations of many abstract wireless network concepts without practical hardware and software demonstrations. For instance, students might not understand the importance of hop-to-hop data relay in sensor networks until they see the high packet loss rates in a single-hop scenario—that is, direct source-destination communication. We use three types of assignments to enhance students' learning.

Weekly class lab assignments

We assign regular labs for two courses. In Principles of Wireless and Mobile Networks, we ask our students to measure radio interference, packet loss rate, and so on and to build a simple ad hoc network routing scheme. In Data Management in Pervasive Computing, we offer a series of labs focusing on sensor network data collection, analysis, and mining using Crossbow motes and RFID devices.

10-week class projects

In Pervasive Computing Architecture and Design, we organize students into teams to build wireless projects and to make presentations and demonstrations in the course's final week.

Senior projects

It's important to involve senior-project students in pervasive computing topics because they'll soon be entering the workforce. Several senior projects have produced some promising results. For example, figure 2 shows a senior project created by Laura Celentano, Paul Tilghman, Stephen Mokey, and Andrew Sackett in the spring quarter of 2006. They built a medical sensor network similar to CodeBlue (www.eecs.harvard.edu/~mdw/proj/codeblue) but built their own RF communication boards and PDA software to receive the sensed data.

EDUCATION EVALUATION

To evaluate the teaching performance in our courses, we invited two wireless experts (D. Raychaudhuri and Ivan Seskar) from Cornell and Rutgers to freely respond to these questions:

- Are the course and lab material

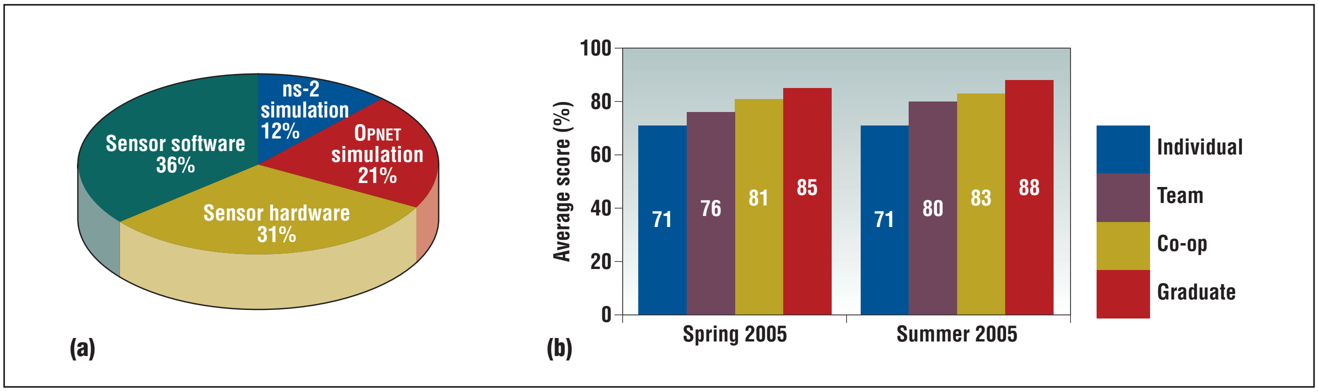

Figure 3. Student outcomes: (a) learning interests and (b) average lab grades.

appropriate and relevant to the profession?

- Is the material being presented logically and cohesively?
- Are the time and emphasis placed on each topic appropriate?
- Are the experiments, projects, and lab equipment appropriate to students without any wireless foundations?

They provided positive evaluations and some important ideas on how to integrate sensor hardware and software education for CE and CS students. For instance, they suggested the use of cognitive radio field-programmable gate array boards (instead of just software simulations) to teach senior students recent technologies in advanced wireless networks. Engineering students prefer to work hands-on in a real environment.

We also asked pervasive computing experts from the local IEEE Communications Society and Sigma Xi chapters to evaluate the PCL. They also provided very positive feedback on our course and lab development.

We use surveys and tests at the beginning and end of each course to evaluate student learning outcomes. One of the surveys measures the students' interest in pervasive computing, and a self-assessment asks them to what degree they feel they achieved the course outcomes.

Our investigation of 60 students' project interests (see Figure 3a) revealed that about one-third of the students (most from CS) enjoy building different software applications for sensors such as Mica2 and Mica2Dot. Thirty-one percent of the students (most from CE) like making sensor hardware from off-the-shelf components (such as microcontrollers, memory units, radios and antennas, sensor chips, and power units). Twenty-one percent of the students like our OPNET software labs for simulating wireless networks. (OPNET is difficult to use because it requires complicated finite-state-machine programming.) Only 12 percent of the students like using the ns-2 simulation tool. This is because the tool requires them to use a standalone Unix machine to write code, and ns-2 has some bugs that aren't easy to fix.

We also investigated the lab grades of four groups of students (individual, team, co-op, and graduate students), from the spring 2005 and summer 2005 courses. Figure 3b shows that students in the later quarter have higher grades. This could come from improved teaching performance because we learned from our previous experiences.

Among the groups of students, the graduate students have the highest average score, which might be because of their solid networking foundations gained in their undergraduate studies. The undergraduate students with industrial co-op experiences perform better in the wireless-networking labs better than individuals without co-op experiences. (RIT requires students to obtain between four to five co-ops during their five-year undergraduate study. The co-ops help students get hands-on experience in their disciplines.) We also allowed the students to form teams (two to three students) to carry out the wireless-networking labs. We find that this lets students share their experiences and submit better results.

ACKNOWLEDGMENTS

The US National Science Foundation CCLI-DUE program (# 0511098) supported the development of the Rochester Institute of Technology's Pervasive Computing Laboratory. We also thank the Cisco University Research Program for supporting the development of sensor network security projects for our senior classes.

Fei Hu is an assistant professor in the Kate Gleason College of Engineering's Department of Computer Engineering at the Rochester Institute of Technology. Contact him at fei.hu@rit.edu.

Ankur Teredesai is an assistant professor in the Golisano College of Computing and Information Sciences' Department of Computer Science at the Rochester Institute of Technology. Contact him at amt@cs.rit.edu.