Education & Training

Editor: Scott F. Midkiff ■ Virginia Tech ■ midkiff@vt.edu

Wearable and Ubiquitous Computing

Tom Martin

EDITOR'S INTRODUCTION

This issue's column continues its coverage of innovative courses in pervasive computing. Tom Martin of Virginia Tech describes a course on wearable and ubiquitous computing that he developed and has taught twice. He describes the course's scope, assignments and grading, and design projects and his experiences with them. Please send me your comments and suggestions for future columns.
—Scott Midkiff

Virginia Tech has offered a course on wearable and ubiquitous computing in the Bradley Department of Electrical and Computer Engineering twice: one at the senior or master's level in spring 2002 (15 students) and another at the master's level in spring 2003 (11 students). The course aims to provide students with an appreciation of current wearable and ubiquitous computing research issues and give them hands-on design experience. The course features a group project (see the related sidebar) that reinforces the readings and lectures. The project also teaches students about the design process in general, including refining a specification, partitioning functionality, creating interfaces between subsystems, working in teams, and planning their work.

COURSE BACKGROUND

The course comprises about 25 lectures covering

- A wearable and ubiquitous computing overview
- Low-power design and power management
- Hardware case studies
- User input/output devices
- Location and context awareness
- Application case studies

The wearable and ubiquitous computing overview begins with Mark Weiser's papers from the early 1990s[1–2] and more recent articles by Mahadev Satyanarayanan[3] and Thad Starner.[4–5] These provide a road map and motivation for topics covered later in the course. The papers also give the students an historical perspective about the evolution of the research issues since Weiser's early work and about advances in hardware and wireless networking. Many of the students are in their early twenties. For most of their lives, cell phones and personal digital assistants have been widely available, and processor clock speeds have been measured in hundreds of megahertz and main memory in hundreds of megabytes. They have heard about Moore's Law, but this is their first concrete lesson in its consequences. Another major lesson from these papers concerns the interdependence of the user's personal device and the infrastructure available in the user's environment. The required infrastructure and thickness of the user's device primarily depends on the relative power consumption and expense of computation, communication, and local storage.

Because wearable and ubiquitous computing systems face significant power consumption challenges, the course spends a good amount of time on low-power design and system-level power management. These topics are also some of my main research areas and let me connect teaching and research. I begin this section of the class with the power consumption mechanisms in digital CMOS (complementary metal-oxide semiconductor) circuits and then move on to higher-level power management problems, such as dynamic CPU speed-setting, low-power compilation and source code modification, and power management state transition strategies. We also spend time studying batteries—particularly characteristics of various battery chemistries and battery life estimation.

QUICK FACTS

Courses: ECE 4984 and ECE 5984, Wearable and Ubiquitous Computing
Department: Electrical and Computer Engineering
Institution: Virginia Tech
Instructor: Tom Martin (tlmartin@vt.edu)
Level: Undergraduate and graduate
URL: www.ece.vt.edu/~tlmartin/ece5984_wearable

Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/03/$17.00 © 2003 IEEE

Case studies of the Itsy[6] and the IBM Linux wristwatch[7] unite many of the low-power issues. Itsy's main design goal was high performance with low power consumption, so a large research community would find it useful; size was a secondary consideration because of the PDA form factor. However, size drove the IBM Linux wristwatch's design because of the form factor. IBM also wanted an intuitive user interface in an aesthetic package. Consequently, the two research prototypes illustrate the trade-offs that designers can make between computational performance, power consumption, and physical size.

Another major problem wearable computing faces is the impracticality of traditional user I/O devices such as screens and keyboards. The lectures cover alternative forms of interaction, including tactile displays, manipulative user interfaces, and movement-aware clothing. I also ask students to use an alternative text input interface, Dasher (www.inference.phy.cam.ac.uk/dasher), and compare it to typing and writing by hand.

Several of the user I/O devices provide a smooth transition into context-awareness because they can be sensitive to user actions that are explicitly meant to be input or to implicit actions that an application can use to infer the user's current context. For example, an explicit action would be a hand signal that tells a stereo to lower its volume, while an implicit action would be walking upstairs. Context-awareness involves knowing the user's location, activity, companions, and nearby resources. Without context-awareness, a wearable computer will only distract the user, and a ubiquitous computing environment will offer little benefit. The course lectures discuss implementations using various types of sensors (for example, accelerometers, magnetometers, omnidirectional video cameras, and microphones) and post-processing. We also discuss location-awareness, particularly methods for determining location indoors.

The remaining lectures comprise application case studies. These case

SAMPLE PROJECT DESCRIPTIONS

The group design project aims to give students hands-on experience designing a wearable and ubiquitous computing system and to teach them about aspects of the design process that will help them with any computing system's design. Here I give some representative sample project descriptions.

Wearable computer front-end for emergency response wireless network

When responding to a disaster, you must keep track of emergency personnel and transmit timely information in the field. In this project, students should create a front-end system such that someone in the command center will have a map showing the locations of all personnel and be able to select an individual on the map to communicate with them or find out about that individual's environment. The individual will have a body or head-mounted video camera, a microphone and headset, and perhaps other sensors (for example, temperature, oxygen, and physiological).

A project by-product should be a product-feature matrix for commercially available wearable computers that will suit this task.

Camera for historical surveys

Before beginning a major construction project such as a road, the construction company must perform a historical survey of the area where the construction will take place. This survey documents any structures and features that might have historical significance. Currently, an individual takes field notes by hand describing the location and structure details and takes photographs of the structure and surroundings. Back in the office, the individual locates the structure on a map, types the notes, and integrates the photographs into the notes. The surveyor then enters this into a GIS (geographic information system) and perhaps creates a computer model of the landscape.

The project should create a historical survey device for a digital camera that records the photograph's location and the direction it was taken from. The device should automatically place the photograph on an electronic map of the area.

Ultrasonic building-mapping garment

This project aims to model and create an electronic-textile garment that maps a building as the user walks through it using ultrasonic emitters and detectors, and perhaps other sensors facing in all directions. The device should be able to estimate the distance to objects in all directions (within a certain range) and create a map of the areas (of a building) that a user has passed through. The system should account for the user's movement (it might be necessary to stand at an entrance to create an initial registration point).

Students should model the garment in Ptolemy (www.ptolemy.eecs.berkeley.edu) in two dimensions. The model should take as input a floor plan and a path through that floor plan, and output the garment's map along the path. The model must account for the ultrasonic sensors' physical properties. Students should use the model to answer questions such as

- What are the algorithms for mapping and feature detection (for example, walls, corners, and openings)?
- How many ultrasonic emitters and detectors do you need? How should you place them on the body? How does the generated map's accuracy vary with the number and placement of emitters and detectors?
- What information do you need about the user's movement (for example, velocity and rates of turning)?

Continued on p. 10

SAMPLE PROJECT DESCRIPTIONS (cont'd)

Continued from p. 9

Electronic buttons for e-textiles

This project aims to devise a method for reliably and cheaply connecting a printed circuit board button to a textile. Soldering alone won't suffice due to mechanical problems and other incompatibilities. Even if you could solder, you'd still have to line up the button's connections with fabric's threads. Students will need to explore other options for attaching the battery, for example, by modifying the connectors that ribbon cables use or modifying regular snap buttons used for clothing.

For this project, I'll give students an audio circuit (microphone plus op amp filter) that they'll have to implement on a button-sized PCB and connect to an e-textile. Other desirable buttons include LED, display, and microcontroller with analog/digital buttons. Students should answer questions including

- What are your options for connecting the button to the fabric—the relative advantages and disadvantages of each?
- What is the right protocol for the button to communicate with the fabric (for example, Firewire or USB)?
- How many connections to the fabric can a button reasonably support?
- How big can a button reasonably be before the mechanical properties (flexibility, strength of attachment, and so on) become unacceptable?

studies cover a range of applications, from wearable computers for UPS warehouses and train maintenance to augmented reality for conferencing and modeling outdoor environments. Some applications were widely deployed, with tens of thousands of units shipped, while others were deployed only for limited field trials. These applications show how the topics previously covered in the course come together in a particular design. They also show how practical issues such as ergonomics impact an implementation and require consideration before ubiquitous computing can become truly ubiquitous. Other applications, particularly in augmented reality, are currently research demonstrations, but they illustrate potentially attractive future applications.

The first time we offered the course, I included sections on wireless networking and privacy issues. But I didn't have enough time to fit in all of the topics. Even without covering networking and privacy, the course's pace is difficult, and many students have commented that they would have preferred more depth on various topics. However, the comments differed on which topics should have received more depth, so the coverage is probably about right for an introductory course.

COURSE DETAILS

I base the student's grade for the course on a group design project (40 percent), an individual research report (25 percent), an examination (20 percent), and homework and reading summaries (15 percent). The individual research report is a survey paper that targets the class as its audience. I encourage students to find a topic related to their own research (such as radio frequency circuitry, wireless networking, and embedded systems), and many students use it to find background material for a thesis.

No textbook adequately covers the course's range of topics, so all reading assignments come from journals and conference proceedings. Students must read three to four papers per week, and I also provide a few supplemental readings. This exposes many students to extensive readings from the research literature for the first time. To help them with their reading, I require them to write a brief summary of each paper, submitted via email at the beginning of the week. I also ask them to submit a list of questions about the readings, which I try to work into the lecture if possible. The class meetings are meant to be conversational, and I encourage students to ask questions and make comments. Consequently, the discussion often follows tangents to the prepared lecture, but they are usually fruitful, informative, and thought provoking. I've even incorporated "tangents" from the course's first offering into the prepared material for the second offering. For example, a question about how CPU speed-setting policies would handle a certain situation led me to prepare an extended example comparing the policies' behavior.

During the last few weeks of the course's first offering, I no longer required reading summaries, to give students more time to focus on the design projects. Unfortunately, this brought on a noticeable decline in the amount of dialogue. Although the students were initially grateful for the lighter workload, one of the most common comments in the course evaluations at the end of the semester was that the summaries should have been required throughout the term. As one student said, "Continue to require weekly reading summaries. I found it harder to keep up [with the readings and class discussion] when they weren't required."

DESIGN PROJECTS

For the design project, students work in groups of two to four people. I base their project grade on weekly written progress reports, an oral presentation, a demonstration, and a final written report.

A few weeks into the course, I hand out descriptions of possible projects. The students have a week to look over the project descriptions before forming teams. No two teams can work on the same project. The teams choose from about six to eight projects, but only those projects (typically four or five) that have enough students interested in them to form a team go forward.

I make the project descriptions intentionally vague for two reasons: First, it's good experience for the students to iterate on a specification with a customer (in this case, me). Second, it gives them considerable leeway in making design decisions. Having too specific descriptions would force students down a design path that they might not choose on their own. The related sidebar shows samples of project descriptions.

The projects tend to relate to research performed on campus. For example, I have based several projects on work by Virginia Tech's electronic textiles group (www.ccm.ece.vt.edu/etextiles), which is currently focusing on e-textiles for wearable computing and a related design tool framework. A few projects have sprung from the research topics of students in the course. Notable projects include

- A Bluetooth-based personal authentication device
- A wearable computer for emergency first responders and field commanders (see Figure 1)
- A model of an e-textile garment for mapping a building as the user walks through it (see Figure 2)
- A model of a garment that can detect its own shape
- Electronic buttons that permit cheap and reliable connection of electronics to e-textiles (see Figure 3)

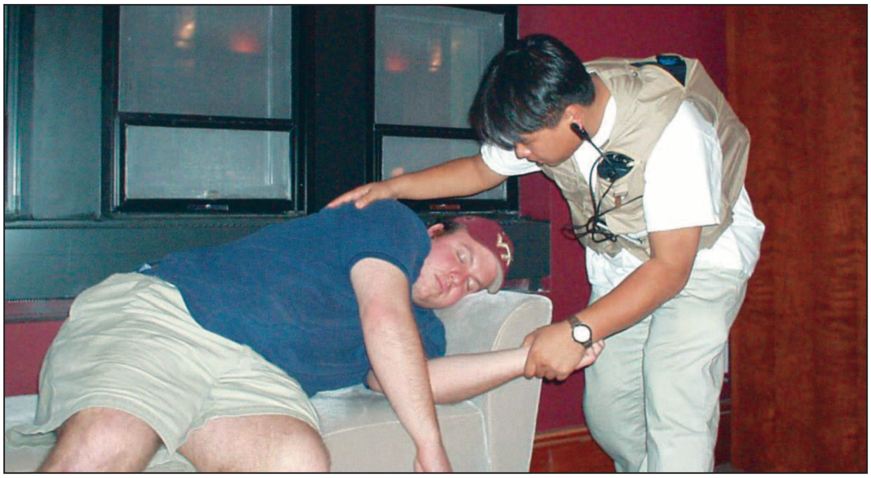

Figure 1. A student wearing a prototype vest-based wearable computer for emergency response personnel, RANGER (Rapidly Accessible Network for Geospecific Emergency Response), while rescuing another from a "sleep emergency" during the final days of their project.

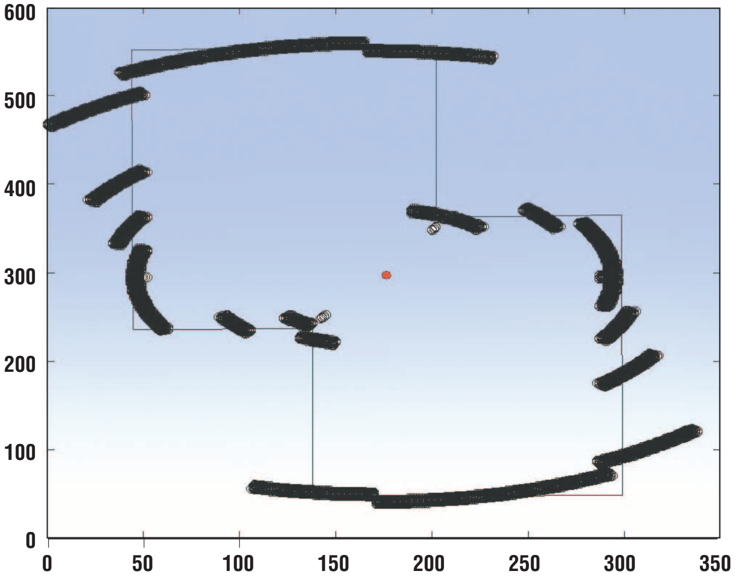

Figure 2. Simulation results for a building-mapping garment showing a room's walls and the garment's current approximation for them.

Each week, students must submit written progress reports (so they don't put work off until the last minute) and part of one class period is spent meeting as a group with me. At the start of the projects, these meetings focus on improving the problem description. Early on, I advise students to break the project into smaller parts that individuals in the group can handle and to carefully delineate the interface between the parts for easy integration later. As the description develops, the weekly meetings begin focusing on specific problems they need to solve for the next week.

With a relatively short design cycle (10–12 weeks), students typically use off-the-shelf components for prototyping. Each group has a small budget and access to my research lab, which includes test and measurement equipment, notebook computers, PDAs, and wireless network cards. Students have to plan for the lead time needed to purchase a component and for what they can do while they're waiting for it to arrive. They often find that components don't work as advertised or have inter-

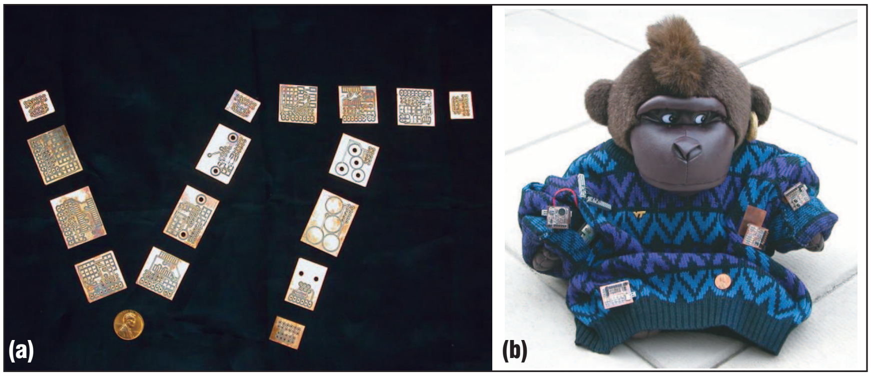

Figure 3. Electronic buttons for e-textiles: (a) unpopulated and (b) finished on an e-textile sweater.

operability problems with the rest of their system or that they overlooked an important detail. Many projects require using state-of-the-art hardware and software that is not well tested, supported, or stable. These practical issues provide valuable lessons for students that they don't learn from reading research papers. It also gives them a greater appreciation for the work behind research papers that describe the design, implementation, and deployment of tens or hundreds of prototypes (such as Itsy[6] and Xerox PARC's tabs, pads, and boards[2]).

Despite widespread gnashing of teeth and sleeplessness in the last few days before project demonstrations, students often feel that the project is one of the best parts of the course. As one student put it: "I liked the idea of working on group projects … as it allowed for some sort of interest that isn't totally present in pre-made engineering projects."

Once students complete their project, they must demonstrate it, make an oral presentation, and submit a final written report. The report has two major pieces, a user's guide and a design document. The design document discusses the project's major design decisions and trade-offs, product-feature matrices for major components (both hardware and software), and test methods and results for both individual subsystems and the overall system. I also ask them to write a section entitled "If I could do it all over again."

I first offered the course at the 4000 level, which is for both undergraduate (seniors) and graduate students. I intended the course to be a mixture of seniors and graduate students, either split evenly or having slightly more graduate students. I ended up, however, with mainly seniors. While seniors for the most part performed well in the course, a number of them had difficulty handling the pace and the required readings. I taught the second course offering at the 5000-level, which is for graduate students only, and we were able to sustain a higher pace of study. Seniors can still take the course if they have a high enough GPA.

To improve the course, I could reduce the group design project's scope and add one or two individual mini projects. These mini projects would permit more depth in certain topic areas. For example, numerous architectural simulation tools exist for estimating software energy consumption that I could use to reinforce concepts on low-power compilation and source code modifications. Similarly, for context-awareness, a mini project might involve having the students use sets of sensors to determine a user's current activity or location.

Another area for improvement would be to make the course more multidisciplinary. Very real issues of industrial design, ergonomics, human-computer interaction, security, and economics come into play in this field. I discuss such issues briefly during the lectures when appropriate, but the students would benefit from more detailed, concentrated treatments. Today's students will be tomorrow's engineers and researchers; they must be ready to tackle all the issues if ubiquitous computing is to fulfill its promise.

REFERENCES

1. M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 265, no. 3, Sept. 1991, pp. 94–104.

2. M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Comm. ACM*, vol. 36, no. 7, July 1993, pp. 75–84.

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

4. T. Starner, "The Challenges of Wearable Computing: Part 1," *IEEE Micro*, vol. 21, no. 4, July/Aug. 2001, pp. 44–52.

5. T. Starner, "The Challenges of Wearable Computing: Part 2," *IEEE Micro*, vol. 21, no. 4, July/Aug. 2001, pp. 54–67.

6. W. Hamburgen et al., "Itsy: Stretching the Bounds of Mobile Computing," *Computer*, vol. 34, no. 4, Apr. 2001, pp. 28–36.

7. C. Narayanaswami et al., "IBM's Linux Watch, the Challenge of Miniaturization," *Computer*, vol. 35, no. 1, Jan. 2002, pp. 33–41.

Tom Martin is an assistant professor in the Bradley Department of Electrical and Computer Engineering at Virginia Tech. Contact him at tlmartin@vt.edu.