Pre-Print for IEEE Computer Society

# Teaching Pervasive Computing: A Report and a Look Ahead From a Dagstuhl Seminar

Andrew L. Kun
University of New Hampshire

Anne Roudaut
University of Bristol

Audrey Girouard
Carleton University

Orit Shaer
Wellesley College

IN A RECENT Education & Training column, we addressed three central questions regarding teaching pervasive computing: Why is it important to teach pervasive computing, What should we teach students, and How should we do it?[1] Subsequently, we organized a Dagstuhl Seminar entitled "Ubiquitous Computing Education: Why, What, and How" to explore these questions in more detail.[2] The workshop gathered 26 faculty members and one undergraduate student[3] to discuss the current state of ubiquitous computing education and to provide ideas for how to improve on our current practices. In this column, we discuss, and expand upon, the work of the seminar, including a detailed overview of the challenges of teaching pervasive computing, proposing a curriculum for students with different backgrounds, and exploring innovative active learning methods for pervasive computing.

## INTRODUCING DAGSTUHL: A SEMINAR SERIES IN A GERMAN CASTLE

Before we report on the results of the seminar, let us first introduce Dagstuhl. Schloss Dagstuhl—Leibniz Center for Informatics is the German organization that hosts the highly prestigious Dagstuhl seminar series. Organizers propose seminars, and if the scientific committee accepts the proposal, Dagstuhl takes on the logistics of the event and provides financial support such that participants can attend at a low cost. The location is idyllic: it is a castle (Figure 1) located in the south-west of Germany, where meeting rooms combine modern amenities with old-world charm. The grounds offer many miles of scenic paths to walk, jog, or bike, e.g., the ruins of a medieval fort are only a few minutes from the meeting rooms. Participants share meals in the castle restaurant, and usually chat late into the night in the pool room or the cellar. This place is made for engaging in deep discussions with your colleagues.

**Figure 1.** Dagstuhl Castle is the location of tens of computer science seminars each year.

## ON TEACHING PERVASIVE COMPUTING

We organized our workshop around the central themes we proposed in our recent Education & Training column: Why is it important to teach pervasive computing, What should constitute the curriculum, and How should we teach the curriculum? On day one, following a round of introductions, we explored the question “why?” and we also discussed the various challenges of teaching pervasive computing. Participants brainstormed ideas around grand challenges. On day two, we explored the question of what we should teach about pervasive computing. Participants organized small groups and worked to create a curriculum for ubicomp education for students of various backgrounds (e.g., technical versus humanities, and university versus industry), different degree levels (undergraduate, graduate, or a short training program). We devoted days three and four to how we should teach pervasive computing. Participants generated a list of their current active learning methods or tools and exchanged them in a speed networking fashion with each other. Participants also developed and experienced new active learning pedagogies on ubiquitous computing topics. We also discussed pedagogies for academic ubiquitous computing (ubicomp) programs. Finally, on day five, we wrapped up the seminar and formulated plans for concrete collaborative actions.

## Why?

Perhaps our most animated discussion was about the definition of pervasive (ubiquitous) computing. What exactly is ubicomp? Clearly, ubicomp brings together multiple disciplines. So, which disciplines, or subdisciplines, are at the core of any definition of the field? Is it human– computer interaction? Networking? Embedded electronics? Our discussion made it clear that there is no broad agreement on this topic, and there would be value in a detailed description of the elements of this field.

While not everyone agreed on a single definition of ubicomp, we all shared the Weiserinspired broad working definition that ubicomp is a field that explores connected computing devices that are embedded in the fabric of our lives.[4] From this starting point, we identified a number of issues that can help us determine why we should teach ubicomp. One line of reasoning revolved around the idea that research and ubicomp are intimately connected. This is because ubicomp remains a new and rapidly developing area. Thus, teaching ubicomp can be a vehicle to instruct students in the general tools of creating new knowledge: the tools of the scientific method. This could be especially valuable in academic programs that historically focus on specific tools for generating new knowledge (such as the tools of math, physics, and engineering).

Another interesting topic was the relationship between industry and academia. Where is the balance between industry motivating the need for teaching ubicomp, and academia providing possible paths forward in research? What are the roles of research publications and products? Which is important, and when? How do the answers to all these questions change over time? We did not reach conclusions, but our discussions clarified that the role of industry in education and training is a hot topic that deserves a great deal of attention.

Another topic discussed in the seminar is the need for diversity of perspectives and disciplinary backgrounds in the workforce. We agreed that one reason to teach ubicomp is that our students need to become experts who are comfortable solving problems and innovating in a multidisciplinary environment.

## What?

So, what is "ubicomp education done right?" One aspect of this question is what content we should teach. We first proposed that the topics to teach depend on the student audience as follows.

1. Are we designing for students in a standalone course (e.g., a one-semester undergraduate course for computer science students), or a program (e.g., a graduate certificate program in ubicomp)?

2. What is the target education level? Are we designing for undergraduate students, graduate students, employees of a tech company, or some other group?

3. What are the students' backgrounds? Do they all have a technical background in a single discipline (for example, electrical engineering)? Or do the students form a multidisciplinary group, for example, with some having a technical background and others with a social science background?

Next, we introduced a list of topics (grouped into themes) that would be of interest to various students—either because these topics are already part of some courses related to ubicomp, or because we feel they should be in the future. We then collaboratively created a matrix of student characteristics versus topics.[2] This matrix can act as an inspiration for those designing various teaching and training curricula for ubicomp. Here, we want to point out two issues related to the matrix.

First, the list reflects those who created it. The majority of workshop participants had a technical background and so most of the topics in the matrix are technical. The topics are multidisciplinary, in the sense that they cover different technical topics from electrical engineering, to computer engineering, to computer science, to mathematics. One important nontechnical topic that is included in our matrix is ethics. We recognize that ubicomp technology and products impact individuals, communities, and society at large, and seek for our students to be able to carefully consider the intended and unintended consequences of the products they envision and develop. But we expect that students will benefit from curricula that go beyond a brief discussion of ethics, and include a variety of both technical and nontechnical topics. For example, an exposure to topics in psychology, sociology, and anthropology will benefit technically oriented students just as exposure to coding and electronics prototyping will benefit those in the social sciences.

Second, it is instructive to see how topics of interest to ubicomp change over time. Our matrix includes prototyping and fabrication techniques. In contrast to a decade ago, today an instructor can include these topics in ubicomp training because the tools for prototyping and fabrication are widely available at affordable prices. We expect that two technologies that will soon be similarly widely available at reasonable prices are mobile eye tracking and augmented reality. Today, devices in these two categories are available, but the cost is still several thousand dollars per device, in contrast to tens of dollars for a prototyping board.

### How?

Next, we focused on pedagogy—How should we teach students? Our approach was first to list and discuss the issues that are directly or indirectly related to pedagogy, and second, to generate new educational material that can complement existing educational resources.

One observation that emerged from our discussions is that instructors must make sure that students can manage the workload of an interdisciplinary ubicomp course. Many ubicomp courses present material related to a number of disciplines. Furthermore, many of these courses have a project component, where students might need to be familiar with topics from a variety of disciplines. Instructors in these courses could inadvertently overwhelm students with the sheer number of topics that the course covers.

A related problem is that of deciding how much a student needs to learn about a topic. When are we satisfied with students only achieving shallow understanding, and when do we aim for them to have deeper insights? Furthermore, how does this vary from one student to another in the same course? For example, if a ubicomp course project brings together a psychologist and a computer scientist, how much do we want the psychologist to learn about sensing, and how much do we want the computer scientist to learn about cognitive load?

And, once we have a satisfactory answer to the questions above, how do we assess student progress? Does this require multiple instructors with different backgrounds? Finally, how do we scale this approach to larger groups of students, and how do we adapt it to online learning?

Clearly, we do not yet have the answers to the above questions. In order to start finding the answers, workshop participants worked in small groups to create new educational material for topics of interest. This was a valuable exercise because it further helped sharpen the questions in our minds. The small groups tested the new material on each other, allowing for further in depth exchange of ideas.

## OUTLOOK

Ubicomp education is complex. The field has a broad set of stakeholders: students, educators, industry, government, as well as the general public. Each of the stakeholders has multiple communities with different backgrounds, needs, and expectations. In five busy and exciting days at Dagstuhl, we identified a number of issues to consider in ubicomp education and training, and we identified a number of noteworthy approaches that are currently used at institutions around the world. We also found that large unresolved issues remain. One such issue is the very definition of what ubicomp is and what it is not. Another is that we still do not quite know how best to address (and take advantage of) the heterogeneity of our students. Furthermore, we do not yet know how best to align educational approaches with the goals of various stakeholders. On the other hand, the workshop also made it clear that there are a variety of well thought-out and successful ubicomp educational approaches that work well for students. Perhaps most importantly, our community embraces multidisciplinarity, and more broadly it embraces diversity: we realize that the workforce of the future should be diverse in education, as well as along a host of other dimensions. Such a diverse workforce is needed to support the needs of our society.

Also, technological advances have provided educators with tools that can make a difference in ubicomp education. We can train students to create devices, to collect measurements, and to process data because the technology to do this has become accessible to most, both in terms of cost and in terms of ease-of-use. And when students need training, they can take advantage of a broad range of resources, from tutorials to full blown online programs, from which they can find the help that they need.

So, what is next for ubicomp education? There are many exciting avenues to pursue. We will wrap up this article with three such avenues—each of them connects teaching ever advancing technologies with helping students understand the needs of users:

1. **Doing good.** The last few decades brought us unprecedented technological advances that make it easier to implement our ubicomp ideas than it has ever been before. But which ideas should we pursue? How to apply ethical lenses to evaluating new ideas? Which ideas are the good ideas— those that will ultimately help us do something beneficial in the world? We can think of these ideas as those that are simultaneously useful, ethical, sustainable, and profitable. How do we teach our students to find these good ideas? Furthermore, how do we teach them not to pursue ideas that have harmful consequences?

2. **AI and ubicomp.** Advances in artificial intelligence (AI) have the potential to transform the world.[5] But AI systems will not only be confined to situations where they interact with individuals one-on-one (such as intelligent investment assistants or robotic nurses), but will also be combined with pervasive computing technologies and have world-wide implications. How can we prepare our students to create such systems and to make sure that they implement good ideas, as in point one above?

3. **Learning how to generate new knowledge.** All human knowledge is advancing at an increasingly rapid pace. We must help all students, and this includes students preparing for careers related to ubicomp, to participate in the complex process of generating new knowledge. For this, they will have to acquire many specific tools—these are the tools we discussed in the section entitled “What?”. But, crucially, they will also have to acquire the general tools of generating new knowledge: understanding how to ask questions that have not been answered before, how to propose hypotheses, how to test them, and how to turn what was learned into products that impact us all.

Interested readers can find further information on the topic of ubicomp education in our Dagstuhl report.[2] This includes insightful reflection statements by all of the participants, a list of suggested reading, and a summary of possible active learning approaches for ubicomp.

## ACKNOWLEDGMENTS

The authors would like to acknowledge the support of Schloss Dagstuhl—Leibniz Center for Informatics in organizing the seminar. The authors would also like to acknowledge all of the participants of Dagstuhl seminar 19232 for contributing in myriad ways to the success of the event. The work of Andrew Kun and Orit Shaer was supported in part by NSF grant OISE-1658594. The work of Anne Roudaut was supported by the Bristol Institute For Learning and Teaching.

## REFERENCES

1. A. Girouard, A. L. Kun, A. Roudaut, and O. Shaer, "Pervasive computing education," IEEE Pervasive Comput., vol. 17, no. 4, pp. 9–12, Oct.–Dec. 2018.

2. A. Girouard, A. L. Kun, A. Roudat, and O. Shaer. "Ubiquitous computing education: Why, what, and how (Dagstuhl Seminar 19232)," Dagstuhl Rep., vol. 9, no. 6, pp. 26–54, 2019.

3. A. McLeod, "Ubiquitous computing education," IEEE Pervasive Comput., vol. 18, no. 3, pp. 59–62, Jul. 2019.

4. M. Weiser, "The computer for the 21st century," IEEE Pervasive Comput., vol. 1, no. 1, pp. 19–25, Jan. 2002.

5. J. Brockman, ed., Possible Minds: Twenty-five Ways of Looking at AI. Baltimore, MD, USA: Penguin Press, 2019.

## AUTHORS

**Andrew L. Kun** is currently a Professor of electrical and computer engineering with the University of New Hampshire, Durham, NH, USA. Contact him at andrew.kun@unh.edu.

**Audrey Girouard** is currently an Associate Professor with the School of Information Technology, Carleton University, Ottawa, ON, Canada. Contact her at audrey.girouard@carleton.ca.

**Anne Roudaut** is currently an Associate Professor with the Department of Computer Science, University of Bristol, Bristol, U.K. Contact her at anne.roudaut@bristol.ac.uk.

**Orit Shaer** is currently the Class of 1966 Associate Professor of Computer Science and the Co-Director of the Media Arts and Sciences Program, Wellesley College, Wellesley, MA, USA. Contact her at oshaer@wellesley.edu.