A Pervasive Computing Workshop for Pre-Collegiate Students

William R. Hazlewood, Dennis P. Groth

Human-Computer Interaction / Design
School of Informatics
Indiana University at Bloomington
901 E. 10th Street
Bloomington, IN 47408 USA
{whazlewo, dgroth}@indiana.edu

ABSTRACT

There are several difficulties that arise when teaching students about the design and implementation of pervasive computing systems. Some of these issues are caused by the tension between design and implementation. In particular, students may limit their design based on what they perceive they are capable of implementing. In this paper we describe our first attempt at creating a method for separating design from implementation for pervasive computing. We tested the method by having a group of non-experts participate in the design of a pervasive technology.

Keywords

Pervasive computing, Methodologies

INTRODUCTION

There are several pervasive computing development platforms that are powerful and simple to use. Examples of these platforms include GroupLab's Phidgets [2], Parallax's BASIC Stamps [5], and MakingThings' Teleo [4]. These platforms offer students access to a wide array of sensors and controllers that can be used in pervasive computing projects. With only a few lines of code a student can have motion sensors controlling audio levels, humidity sensors controlling fans, or make it so that all of their household appliances are enabled from a single control panel. Because of the relative simplicity involved in using these platforms, students are allowed to focus more on *what* they want to build rather than worrying about *how* it is going to be built. However, because of the nature of pervasive computing, it can still be difficult to shift focus between the implementation issues and the goals of the overall design.

In order to teach all of the concepts necessary for a well-rounded pervasive computing class, one must take into account both the design and implementation of a pervasive computing system, but even with the existence of several easy-to-implement pervasive computing platforms, there are issues that have to be considered. Specifically, how does one let students explore the vast possibilities of pervasive computing without confusing them with the challenges of implementation?

PERVASIVE PEDAGOGY

When a teacher wishes to conduct a class on pervasive computing, typically it involves a project where the students focus on some sort of implementation. For example, a group of students may wish to implement a system in which sensors are placed around a room in order to predict the individual behavior of people as they enter and exit. To adequately facilitate a project such as this, the teacher must have access to a suitable lab space that is stocked with a wide variety of the necessary components and tools. These include computers, sensors, controllers, soldering irons, cables, breadboards, nuts, bolts, and several other raw materials. Unless they have prior knowledge, a class in pervasive computing must inform the students how to use this specialized equipment as well as how to construct the programs that drive them. Learning all of the necessary technology leaves little time for the students to think on the human-centered aspects of pervasive computing. Consequently, the students end up spending the majority of their time focusing on *how* to build a system rather than *what* the ideal system might be.

A PRE-COLLEGETE SUMMER WORKSHOP

In order to explore methods of teaching pervasive computing concepts without focusing on hardware or programming languages, we taught a session on the topic to high school students who attended a summer workshop at the School of Informatics. The majority of these students had no previous programming or hardware experience, so it would have been impossible to cover all the information necessary to do a relevant implementation in the 3-hour session we had planned. One suggestion we considered was to design a lab where the students could follow along, pasting small code segments into a mostly pre-written program that controlled a sensor and servo motor, but we felt that the outcome would not give the students a full understanding of the possibilities or implications of pervasive technologies.

We decided to try an approach that did not involve an actual implementation. The goal was to get the students thinking abstractly how they would like to solve a problem using pervasive technology. After a short lecture the students were shown several videos that detailed current pervasive technology initiatives, including Hewlett-Packard's *CoolTown* [3], and the European Union's *DAIDALOS* [1]. These videos were selected because they show large scale scenarios of the impact pervasive technologies may have on the future.

After the lecture and videos the students were asked to design a solution to some problem of their choosing from a perspective of pervasive computing. Specifically the students were asked to:

1. Select a specific problem that bothers you in your daily life (e.g. getting settled into bed and realizing that you had forgotten to lock the front door and turn off the television).
2. Figure out what sorts of things in your environment could be changed to keep this sort of problem (1) from occurring (e.g. make a television that knows whether someone is actually watching).
3. Think about what you would need to know in order to provide the necessary information for (2) (e.g. the television could have built-in motion or heat detection that would tell it when no one was actually in the room)

These questions encouraged the students to think about their problem without them having to know how to do the actual implementation, but they still needed to have an idea of what sorts of things could be sensed in the environment.

PERVASIVE COMPONENT CARDS

To provide the students with some notion of what was easily sensed with existing technology we provided set of *component cards* that had an image of a controller, sensor, or display device, along with a short simple description. We decided to use this format based on the success of a previous research project where information was placed on cards and used in a complex design task [6]. These cards also had a series of icons showing the necessary inputs and possible outputs of each. We also provide several blank cards incase the students wanted to suggest their own sensors.

OUTCOME AND DISCSSION

The students did very well with the task assigned to them. The designs produced included a shoe that adapts to the kind of activity being done (hiking, football, etc.), a system for alerting people that they have forgotten important items (keys, wallet, etc.) using RFID tags, and an alarm clock that attempts several methods of waking a lazy sleeper (see Fig 1). The cards we provided blended well with the student's designs, but several of the groups used the blank cards to design their own devices, including a set of contact lenses that changed color based on the level of adrenaline in the wearer's blood, and a glove with a galvanic skin response sensor so that one can tell if another is possibly lying when shaking hands. The next phase is to develop a way for these students to have the students critique their own in order for them to separate out what is, or is not, plausible for implementation.

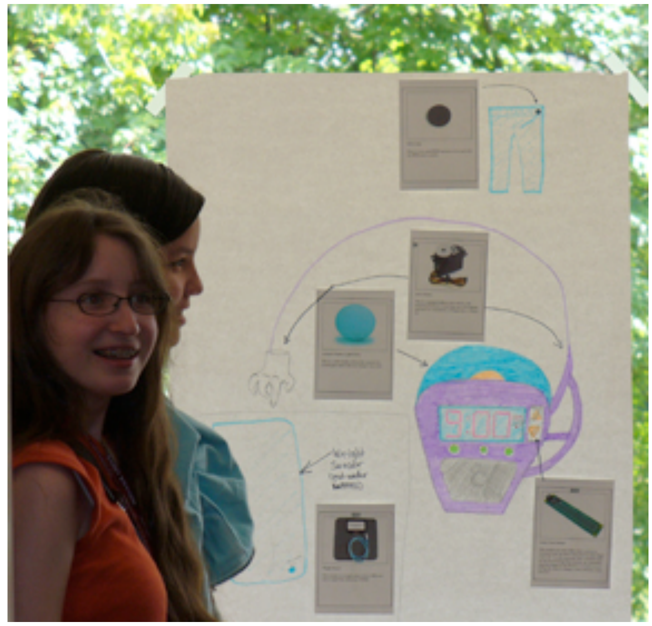

Figure 1: A student design of a sensor enabled alarm clock showing the use of the pervasive component cards.

ACKNOWLEDGMENTS

We would like to thank all of the students and volunteers that participated in the summer workshop.

REFERENCES

1. DAIDALOS -- http://www.ist-daidalos.org June 2006.
2. Greenberg, S. and C. Fitchett, Phidgets: easy development of physical interfaces through physical widgets. *User Interface Software & Technology, CHI Letters*, 2001. 3(2) 209.
3. Hewlett Packard. "A Future Called CoolTown." Available from http://www.hpl.hp.com/news/-cooltown.html June 2006.
4. MakingThings -- Teleo Modules and Sensor Devices http://www.makingthings.com/products/products.htm June 2006.
5. Parallax – Basic STAMP http://www.parallax.com June 2006.
6. Yvonne Rogers, Youn-Kyung Lim, and William R. Hazlewood: Extending Tabletops to Support Flexible Collaborative Interactions. In *Proceedings of TableTop'06*, 71-78.