

Teaching a practical Ubicomp course with Smart-Its

Michael Beigl, Philip Robinson, Tobias Zimmer, Christian Decker

TecO, University of Karlsruhe

Karlsruhe, Germany

Email {michael, philip, zimmer, cdecker}@teco.edu

Abstract

We propose the Smart-Its platform as a platform for teaching Ubicomp in practical student courses. We present four experiments that were undertaken during summer term 2002 along with the outcome. Applications, context, sensor and Wireless ad-hoc communication were the key issues in these experiments. We also introduced the use of the Smart-Its hardware and development platform as tools for such teaching activities.

Keywords

Teaching, smart devices, context, sensors, communication.

INTRODUCTION

Teaching Ubiquitous computing topics in a practical student course requires a platform that is easily used yet provides a realistic environment for students to solve real-world problems, while experimenting. This paper therefore presents Smart-Its as a platform for conducting such experiments in student courses. Smart-Its [1] are small devices (1.5x4 cm) containing sensing (various), computing (5 MIPS) and communication (via RF) capabilities. These Smart-Its are intended to be attached to everyday objects so as to enhance the functionality of the object, deliver information (collected through sensors) to other devices and to produce acoustic or visual output. The Smart-Its platform also includes a software library and development support that is simple to program when implementing Ubiquitous Computing scenarios. The platform is powerful enough to allow programmers to implement algorithms and rather complex context detection. Smart-Its provide access to functionality like sensors and communication through libraries and operating system functions. For our practical course we identified four main topics where Smart-Its are helpful. Smart-Its are used in the experiments as **Smart-Its nodes** for sensing data in the environment, for computing context out of these sensor data, for communicating these data to other nodes, and for building and testing applications. For the experiments, Smart-Its nodes had to be attached to everyday objects by students and then had to be programmed. The paper briefly describes the content of the course held first in summer term 2002, experience collected and also the development environment and technical parameters.

Related Work

Up to now there are only a few platforms available that can serve as a basis for a practical course. One of them is the Berkley Mots [2] system and the TinyOS. In contrast to the Mots Smart-Its provide some better performance values (e.g. faster communication) and also a development environment including wireless programming with systems in place over the RF channel. Smart-its also support analysis of context and sensor data via backend systems. First ideas for tools for building ubiquitous computing systems were presented in the first workshop on Ubicomp Tools at Ubicomp 2001 [3]. However, only a minority is available for use now, e.g. the CoolBase or the Phidget system. The drawback of

these systems is their reliance on a wired or simple IR-base communication architecture, which restricts their application range.

Teaching a practical course in Ubicomp

In an example practical course we identified four main topics to be taught; the premier topic was based on building and testing applications in the area of ubiquitous computing, while the remaining three topics demonstrated Ubicomp-enabling technologies. The second dealt with context and sensor technology, the third with context communication between different nodes in a network and the fourth with basic communication and energy issues. The students conducted general experiments for topics 2-4, followed by a more detailed experiment based on topic 1. The motivation to use the Smart-Its platform and the expected outcome of these topics are described in the paragraphs below:

Ad-hoc application experiments

This part of the course was dedicated to collecting practical experiences with applying technology into the everyday environment. In the experiment students were faced with a complex and distributed application, where multiple nodes worked together to fulfill a task. The major outcomes were that students learnt to develop distributed applications, how to address security and privacy issues and where to implement Smart-Its nodes. The student also had to solve practical problems associated with attaching Smart-its nodes to objects and collecting information. A definite advantage of using Smart-Its is the short implementation time for small and medium-sized application settings. Such settings require nodes that are not fixed to wired sensor structures (with long set-up times) and are not reliant on a complicated infrastructure. Furthermore, as Smart-Its computation is processed on-board, applications and experiments are not confined to the lab, plus they can be implemented within a restricted time frame, having no previous knowledge of the technology.

Context and sensors

In this experiment students learnt about the available sensors, which sensors or combination of sensors can be used to retrieve a context and what algorithms are appropriate for use in small sensor nodes. They also became familiar with the characteristics of sensors, e.g. their required sampling rate, the energy consumption and their error rate. The Smart-Its sensor board provides a general set of sensors that can be used for sensor-based applications and for experiments on deriving context from sensor data. Such experiments always require information retrieval yet, due to the data transmission deficiencies of wireless networks, they often require processing of data at the sensors' location. Smart-its support this by providing a complete computing and sensor infrastructure integrated into the boards.

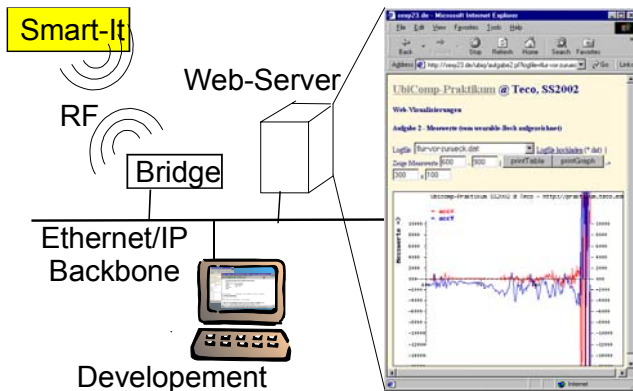
Abstract context aware communication

The outcome of this experiment was a general understanding of how to express context information for use in other applications, how to filter and combine them into new (richer) types of contexts and how to communicate them between applications. Smart-Its

support this through a context protocol layer and an application programming interface (API). This layer is intended to be extended by new types of contexts that may result from previous work on contexts retrieval with Smart-Its.

Network and energy issues

As most of the energy in a node is spent on communication, another consideration in the experiment was the implementation of energy saving and energy aware protocol extensions. A major outcome of the experiment was the understanding of communication issues for ubiquitous computing networks, their characteristics and solution strategies. Smart-Its support access to different layers of communication; In Smart-Its the network layer can be accessed through a simple API providing all functionality to implement various types of algorithms including energy aware protocol enhancements.



Smart-Its Development and Technology

The Smart-Its hardware consists of a generic RF communication board connected over I²C to a sensor board (total size: 4x1.5x1 cm), on which application development is done. Wireless broadcast communication facilitates interact with other Smart-Its in their vicinity (at 125kbit/s), and optionally with servers over a RF-Ethernet/IP bridge. The backend is used for analyzing sensor data through a tool running on a Web-Server (figure 1) while the experimentation process is running. The programming environment (also connected via Internet) consists of the Smart-Its Software Development Toolkit (SDK), compiler and program download utility. The latter one allows you to download compiled programs directly onto a Smart-Its from the development PC.

Smart-Its provide sensors for collecting light, pressure, temperature, acceleration and audio information. The freely programmable microprocessor on the Smart-Its provides about 12kbyte space for programs and about 8kbyte for data. Sensor values can be accessed by simple functions using the SDK. To share information with other Smart-Its, programs can communicate and express sensor values and contexts via APIs. A subscription mechanism of the communication subsystem allows one to watch for selected or all information on the communication channel.

Experiences with Experiments and outlook

During the practical course 4 experiments were carried out. In “Night&Day” students automatically adjusted the time-of-day setting of the Smart-It by observing the environment through the sensors and deriving the context. In “F1 Skates” Smart-Its were attached to inline skates and a kickboard; context had to be re-

trieved and communicated to a second Smart-It attached to the body of the user for output status (Figure 2). In a third experiment a protocol enhancement for energy aware communication of context had to be implemented. The fourth and last experiment, spaces of trust, was about using context collected in a room to generate areas of trust and keys for exchanging documents among those present.

For many of the technologies in UbiComp students had no previous knowledge. Although we thought this would be a major problem, we found out it was not. E.g. the programming language for Smart-Its is plain C. None of the students were familiar with C, yet all were able to run programs successfully after 2 weeks. The provided example code and code-parts were enough to have a quick start into the environment. Also, although there was no previous knowledge of dealing with hardware, the students managed to acquire this skill, e.g. built cables for connecting sensors and batteries to Smart-Its, soldered additional piezos for acoustic output etc. Instead, the major problems appeared when implementing Smart-Its in the physical world. Cable deterioration and inappropriate housings without damping were found as the major sources of error during the run of the course.



Figure 2. Practical Course Experiment

Before the next course held in 2003 we will especially improve these parts of the system. We will also look into using Smart-Its as a platform for other UbiComp topics like network routing and HCI studies.

ACKNOWLEDGMENTS

The Smart-Its project is funded by the Commission of the European Union as part of the research initiative “The Disappearing Computer” (contract IST-2000-25428). General information about the Smart-Its platform is available at <http://smart-its.teco.edu>.

REFERENCES

- [1] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl and H.-W. Gellersen. Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts, UbiComp, pp. 116-122, 2001
- [2] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.E.. Culler, K. S. J. Pister. System Architecture Directions for Networked Sensors in Architectural Support for Programming Languages and Operating Systems, pp 93-104, 2000
- [3] Ubitools workshop
<<http://choices.cs.uiuc.edu/UbiTools01/cfp.html>>

