Teaching Pervasive Computing for Architects

A simple but powerful building simulator explaining the potential and power of pervasive computing through hands-on exercises

Odilo Schoch
Department of Architecture, Swiss Federal Institute of Technology ETH Zurich
http://www.caad.arch.ethz.ch, schoch@arch.ethz.ch

Pervasive Computing will soon be integrated part of the architectural education. The design of digital services and media enriched architecture is expected to become an important element for the architectural design within near future. This technology has an enormous spatial and creative impact. Pervasive computing is seen as the key technology for communication within mediated spaces.
This paper introduces a successful approach of teaching the creative principles of pervasive computing. This reflects the ubiquitous quality of digital technologies and services in both today's life and building industry. It described the aim, technical solution, scope and result of exercises carried out at ETH Zurich (bachelor program).

Keywords: *pervasive computing; ubicomp; interactive architecture; education.*

Introduction

The aim of the course was to teach future architects the principles of pervasive computing by focusing on the topic's overall logic, on possible services and general problems. Only few technical details were aimed to be taught.

In this context principles of pervasive computing are: design impact, human computer interaction, spatial interfaces, spatial impact, multi user problematic, telepistemology (Goldberg, 2000), mediatecture and sustainability. Most of these aspects are known in research, but seen novel in the architectural education. It is known, that the direct formal impact of pervasive computing on architecture is very little – except of elements such as media facades. But secondary impacts are deep. Therefore hands-on exercises were introduces to let the students directly experience the potentials and problems of this new design material.

Technical Solution

The technical system is a low-cost installation using standard technology. It's modular and scalable design is based on commonly used Internet technologies. A conventional desktop PC performed as server, the clients were made in Macromedia Flash, the light switch an network controllable device using http-protocol and the light-sensor a small wireless device sending its information by UDP broadcast. The technical setup is not new and know by computer scientists for decades, but is was carefully selected as it represents the basics of contemporary and future digital networked building technologies. The system combines the following aspects: simple handling, technical stability,

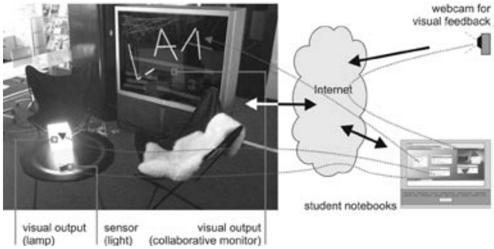

Figure 1
Physical setup at the chair of caad: a remote controllable situation directly attached to the Itnernet.

modular integration, scalability, focus on architectural dimension, graphical programming.

The core of the system is a software named 'echo-server' written in the programming language JAVA™. The software replies incoming XML-based information packages to all connected clients.

Clients are either small devices such as light switches, thumbnail-sized digital sensors or the Flash-clients on the student's notebooks. The software allows the composition of logical dependencies within the network – without programming a single line of code. This software works as a client, it receives all the information sent by another client in the network, both software-clients, server or sensor. The software allows single or networked processing of logical dependencies. Each student had the possibility to create individual dependencies by arranging given templates such as 'if-then-send-variable' or 'if-then-switch light'.

Technical detail of importance is the use of socket connections on port 80 in order to pass most firewalls (Tanenbaum, 2002). Most students are not able to handle unknown technical set-ups on their own computers. Therefore the whole set-up was intensively designed in the way, that creative working was encouraged. A standardized Webcam was installed in order to monitoring the interacted physical space set up within the chair's office.

Students Work

A lecture illustrating applications of digitally networked building- and media technologies in architecture defined the starting point of the exercise. Subsequently the principles, dangers and positive potential of the physical disruption between actor and sensor, space and time of action and result. Examples were for instance the media facades in Ginsha, Tokyo, and the showcase house of the Deutsche Telekom in Berlin, Germany, as partially bad example due to it's massive wiring and omnipresent video displays. A brief introduction of approximate 30 minutes into the systematic of the exercises concluded the setup.

During the introduction, the students were asked to download instantly a simple customized client from the chair's website[1]. As the majority of the students do work with their WLAN enabled notebooks in the lecture hall, the tests could be conducted directly on site. These clients allow only three core activities:

- chat (one to all),
- play sound (one to all),
- switch light in another room on and off.

The students performed first tests, such as messages in the chat, that were instantly projected on the lecture hall's huge screen and collective sound output on all connected computers, wherever the devices are physically located. This illustrated three main aspects of networked : limited privacy in the web, small action causes big output and expect unintentional actions from unknown source (Negroponte, 1995).

Next step was the download of the simple graphical programming interface (Figure 4, right). It is more a configurator with drag&drop abilities than a scripting tool. It's very simple logic allows instant access even for shy or technically afraid students. This client was written by ETH Zurich Chair of Caad for this special exercise and provided in both executables for Mac OSX and Windows®. The software allows conditional comparisons of incoming variables (if-then-else). Incoming means: send by the socket server. For example user SUSI sends the variable A

[1] http://wiki.arch.ethz.ch/asterix/bin/view/ Caad0506ub/SmartHouses: June 2006

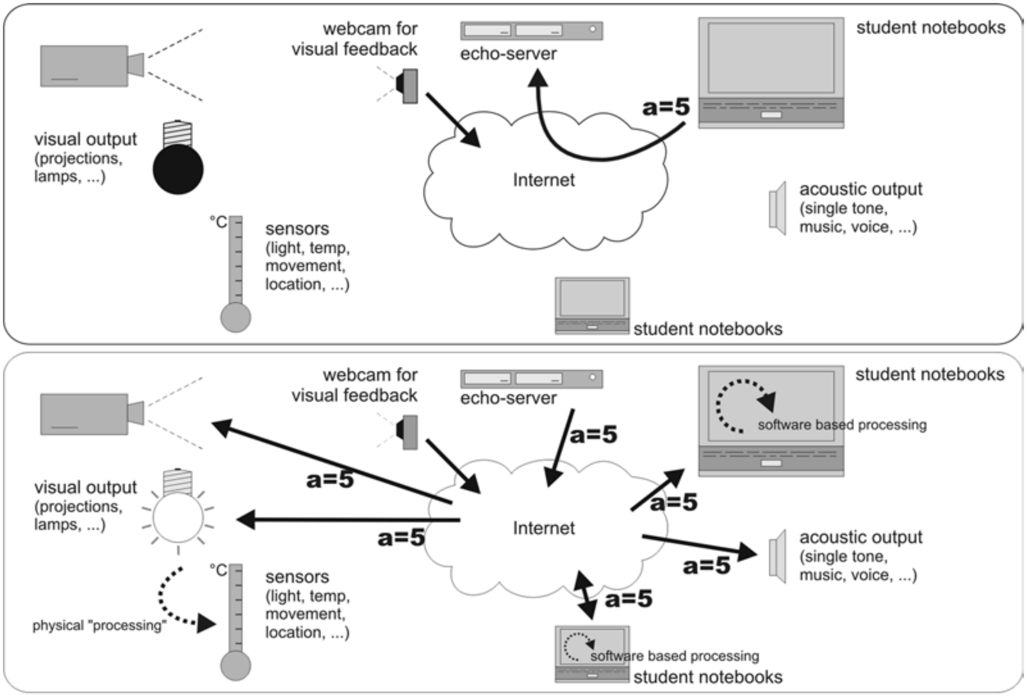

Figure 2
System set-up, virtual vs. physical world and procedure of networked communication using XML packages through socket connections.

with the content 10 to the server. User BEN set up a conditional expression that says "as soon as a variable A comes in and if this is larger than 9, then play the sound 'minor g'". This example will locally play the sound (minor g) whenever somebody sends the variable A with a content larger than 9. The settings were automatically saved on the server under the student's name.

Findings

With little introduction (less than one hour), students were able to understand and experience the intended principles of pervasive computing. The power of networked processing was successfully implemented as students formed groups with each member composing a specific part of the a control circuit on

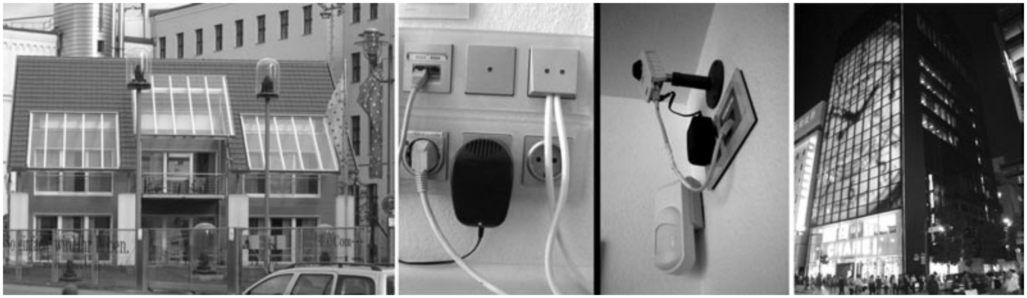

Figure 3
Left and Middle: T-Com smart house in Berlin with questionable installation of analogue and digital motion detectors; Right: Chanel flag ship store in Tokyo, Japan.

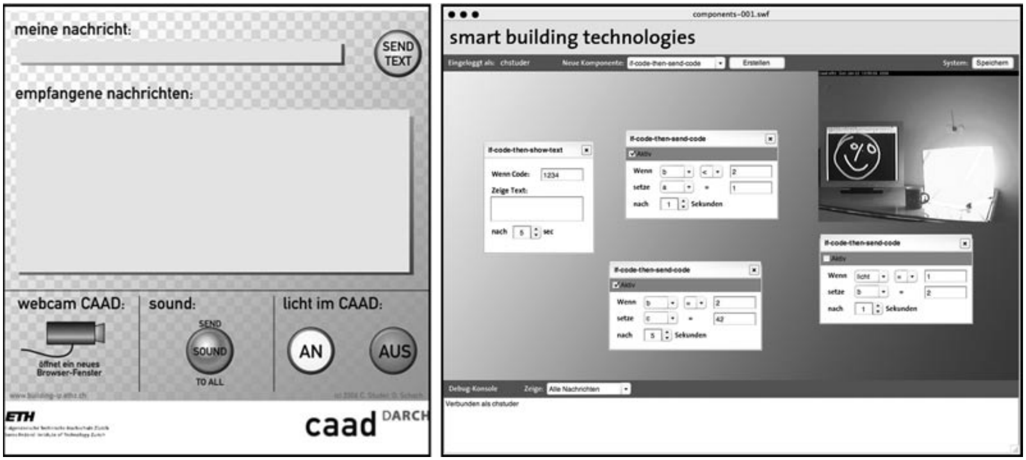

Figure 4
Left: Screenshot of first test client enabling network based chat, light control, webcam-input and collective broadcast of sound. Right: Screenshot of second client enabling graphically oriented "programming" in order to control spatial interactivity.

their own computer.

Students collaborated from different physical locations such as home, café and workplace. They identified the problem of trusting and multi-user activities. Most groups found a smart workaround within the system that allowed their service to absorb disturbances. The students practiced the life in 'digital neighborhood' as described by Negroponte in 1995.

The hands-on exercise was successful as the students could work creatively following their personal scenographic concepts. Most of the students were able to achieve their concept. Technological fundamentals of pervasive computing were learned auto didactically. A brief evaluation of the results by questioning the students showed, that they learned about remote control, differentiation of processing logic (software), networked sensors and actors. The students rarely use the light-sensor as input source. Only one group out of ten did use it. Reason for this might be the prolongated response of the sensor.

Conclusion

The exercise was a success as the students did work out hands on examples. They educated themselves in the logics and workarounds of digitally networked building interaction. In following up exercises, it is recommended to hand out more samples for the faster start and more advanced output.

Acknowledgement

Christoph Studer and Thomas Lenherr for helping to program and run the software.

References

Goldberg, K.: 2000, The Robot in the Garden, MIT Press, Cambridge Massachusetts, p. 3.

Negroponte, N.: 1995, Being Digital, Goldmann/Bertelsmann, p. 13.

Tanenbaum, A.S.: 2002, Computer Networks, Prentice Hall PTR, Indianapolis, 487p.