Smart-Its: An Embedded Platform for Smart Objects

Michael Beigl

Telecooperation Office (TecO)
University of Karslruhe, Germany
michael@teco.edu

Hans Gellersen

Computing Department
Lancaster University, U.K.
hwg@comp.lancs.ac.uk

Abstract

Computer and communication hardware has become so small and inexpensive to consider their embedding in everyday objects. As a consequence it is expected that networked smart objects will give rise to new types of application and in particular such that are more tightly coupled with activity in the physical world. Recent years have seen many design examples for smart objects but for lack of a suitable platform it has remained difficult to study applications that involve networked operation of a diversity of such objects. In the Smart-Its project we are addressing this concern with development of embedded technology designed for post hoc augmentation of everyday objects. This paper describes our technology concept and the design of the 1G Smart-Its hardware/software platform.

1. Introduction

The transition from mainframe to personal computing was marked by 'human integration', considering human users no longer as peripheral but as integral in computer applications. Similarly, the current transition from personal to ubiquitous computing is marked by 'physical integration', considering the physical world around computer and user as integral part of the overall system. A far-reaching approach to achieve such physical integration is to embed computing into the objects and artefacts that are subject of everyday activity. A hindrance though is the lack of a suitable platform that would support such augmentation of everyday objects.

Artefacts are commonly defined as 'something created by human for a practical purpose' and it is compelling to build on these familiar purposes while enabling new applications on the basis of embedded computing and communication. Artefacts thus augmented become smart objects that can be tied directly into software processes to overcome the media break between physical flow of activity and related flow of information. The possibilities appear to be open-ended. For instance in an enterprise environment, smart objects may provide continuous access to their physical state and context, or even embody autonomous behaviour in support of more decentralized enterprise management. Likewise in emerging interactive environments, smart objects may embody physical I/O to be enabled as tangible user interface objects that facilitate richer interactions between people and their environments.

Miniaturization of components is presently reaching a stage at which it becomes practical and affordable to embed processing, networking and physical interaction into even the most mundane objects [1]. This has inspired a range of design examples built over the last years to explore application opportunities and technology design challenges, for instance the Mediacup (a coffee-cup that autonomously computes its use context from embedded sensors, and serves itself to potential applications in the local environment [2]), the Strata Drawer (a chest of drawers that tracks its physical contents to provide new forms of user interaction [3]), and the Pin&Play noticeboard (a board that has smart pushpins autonomously asserting priorities to visually alert users [4]). These examples are generally one-off prototypes and can only provide very limited insight into applications and challenges that may emerge with more pervasive networking of smart objects. Investigation of applications that involve a larger number and diversity of smart objects has so far been hindered by the lack of a suitable hardware/software platform. The generic technologies available for augmentation of common objects are presently limited to passive provision of digital identity [5,6].

In this paper we describe the development of a new platform specifically designed for augmentation of everyday objects. At the core of our contribution are *Smart-Its*. These are small computing devices ultimately envisioned to have a form factor of a label that can be attached post hoc and unobtrusively to arbitrary physical objects, to empower these with processing, context-awareness and communication. We approach this vision with development of a family of small embedded devices that enable prototyping with different configurations of wireless network, sensors and actuators, and that support customization to particular physical objects through software abstractions and development tools. The different types of Smart-Its developed to date include a first generation (1G) optimized for low energy-consumption and deployment in longitudinal studies, a 2G device developed for integration with the Bluetooth world[1], and a "do-it-yourself" device for very rapid prototyping and ease of modification [7]. In this paper we focus on the development of the 1G platform. We briefly reflect on the general Smart-Its architecture and requirements followed by description of the 1G hardware, system software and application development support.

2. The Smart-Its Concept and Architecture

2.1. Technology Concept and Architecture

Smart-Its are generic computing devices however it is fundamental to our concept that we consider them not stand-alone but always as secondary to a physical object of primary interest in the application world. This follows the philosophy of the "Disappearing Computer" which places computing in the background of people's interaction with their physical and social environment [8].

[1] Developed by project partners at ETH Zurich

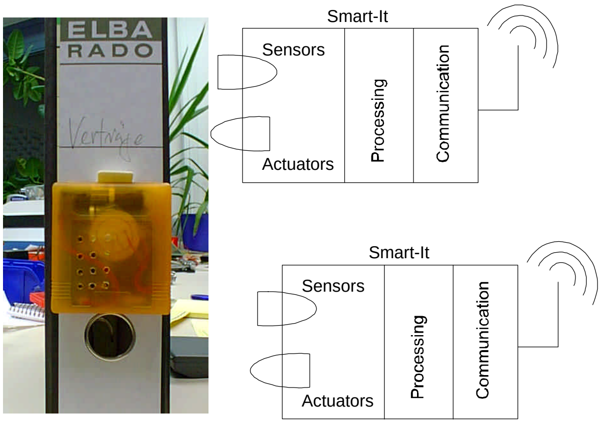


Figure 1: The Smart-Its architecture is based on small devices that integrate processing, physical interaction and wireless peer-to-peer communication. The left illustrates Smart-Its use for augmentation of a lever arch file.

A second defining element of our concept is that we seek to support post hoc attachment of smart technology to common objects. This means objects can be augmented at any time in their lifecycle, by attachment of a suitably configured and adapted Smart-It device. A further distinct element of our concept is to enable active objects with tight coupling between their physical context and digital state, and with a large degree of autonomy and control over their digital self.

The Smart-Its architecture to implement these concepts is shown figure 1. At the core are the Smart-Its units that integrate three basic capabilities: processing, interaction with their physical environment, and communication with peers.

2.2. Platform Requirements

The Smart-Its architecture implies that a hardware/software platform has to integrate physical I/O, a processing environment, and wireless networking. Constraints are imposed by the anticipated unobtrusive embedding of the technology – this has implications for physical dimensions and energy management of Smart-Its.

A basic objective of Smart-its is to enable customization of sensors, perception, context-awareness as well as physical output to specific types of object. This requires flexible configuration, control, and exchange of sensors, as well as support for different sensor modalities and their combination.

Sensors provide objects with awareness of physical context. Communication is required to allow objects to promote a digital presence and to become part of networked applications. Communication requirements include support for local broadcast to disseminate context information to artefacts within certain proximity, and fast reaction to network changes without any form of user interaction.

Smart-Its are targeted at augmentation of very mundane objects. These are often not very big and not mains connected. As a consequence, the devices have to be small, lightweight, and highly energy-efficient. The size constraints rule out integration of large battery packs, while the Smart-Its should nonetheless support long run times without user intervention.

3. Hardware design of the 1G Smart-Its

In the 1G Smart-Its design, the three core functionalities are mapped onto two hardware modules, one for communication and the other for physical I/O, with processors on either module (fig. 2). The modules are interconnected by an I2C data bus and a power bus. The central module is the communication board (also: core board). It is responsible for communicating with other Smart-Its, or with backbone services. Attached to the communication board are one ore more sensor boards, integrating diverse sensors and additional interfaces to retrieve parameters from the environment. The boards measure 50x50x10mm, and weigh 10g resp. 18g.

The main components of the communication board (fig. 3a) are the microprocessor unit (20 MHz RISC Arizona Microchip PIC Processor 16F87x), the transceiver for wireless communication (RFM transceiver module TR 1001 at 868.35 MHz), the power supply (supporting different types of battery) and the software-controlled field-strength regulation unit. The board further contains basic physical I/O (movement switch, LED, buzzer) and various hardware interfaces to connect boards, sensors and serial line.

The sensor boards (fig. 3b) contain their own Microprocessor and RAM enabling on-board computation of context information from sensor observations and execution of application-specific code. The boards integrate sensors for audio (high-linear microphone and amplifier from 50Hz to 20kHz), light level (TSL 250 Light sensor at 880 nm wavelength), acceleration (Analog Devices ADXL 202 2 axis sensor, precision ~5mg), pressure (IEE FSR152 for pressure between 1-100N at low precision) and temperature (Dallas DS1621, >99% accurate between 0-40°). Further main components are flexible power supply, basic actuators (piezo loudspeaker and 3 LEDs), and various interfaces/connectors for integration of additional sensors/actuators.

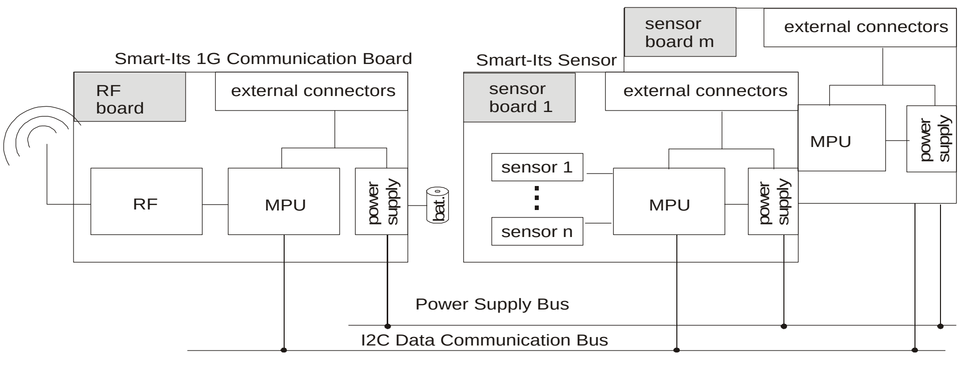


Figure 2: The 1G Smart-Its devices are based on a modular architecture with separate boards for communication and for physical I/O

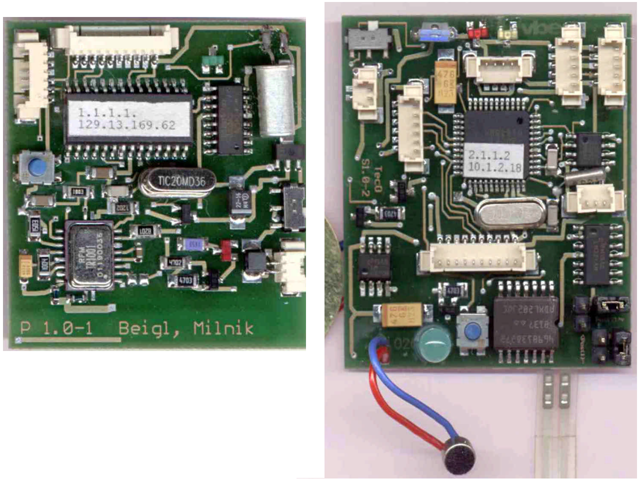


Figure 3: 1G Smart-Its hardware: a) the communication board with RFM transceiver, and b) the sensor board with audio, light, acceleration, pressure and temperature sensors, LEDs and piezo loudspeaker, and a variety of connectors for additional sensors and actuators.

4. System Software

4.1. System software architecture

Smart-Its devices are designed for embedded and highly applied use. The technology concept foresees that the software executed on the device is task-specific in the sense that is customized for a particular physical object. This involves management of subtasks for physical I/O, communication and processing of events, but it does not involve support for concurrent user-level applications that would compete over system resources. The required system software is provided in libraries and becomes compiled into applications in the development process.

As shown in fig. 4, the system software interface is based on three libraries with device drivers for each hardware component. These provide hardware abstraction but also support fine-grained control, for instance over individual sensors. In addition to libraries for the physical I/O and communication subsystems, basic core functions are provided for coordination of access to resources. The processor platform does not support context switching but separate cooperative tasks are supported with semaphores. Core functionality also includes real time clock and calendar, as well as notification of timing violations.

4.2. Physical I/O management

Rich physical interaction is key to the Smart-Its concept for a close coupling between embedded device and its physical context. However, sensor and actuator are the major power drains on the comparatively low-powered Smart-Its devices. To support energy-efficiency, all sensors and actuators can be powered on and off individually. This is provided in two modes, system-controlled and application-controlled ('power-on-demand').

The sensor/actuator library is generally designed to support configurability and extensibility on the basis of device drivers. The drivers encapsulate hardware access and the translation of raw signal values to application-level formats.

4.3. Communication subsystem

Smart-Its communicate via short range radio frequency. They are designed to enable communication among spontaneously grouped smart objects. The emphasis is therefore not on routing and sensor data diffusion as investigated in large-scale wireless sensor networks, but on fast network discovery and exchange of context (as opposed to low-level sensor data).

Data transmission is based on a stateless peer-to-peer protocol. It implements physical access to the transceiver including the coding on the medium, the data link layer (DLL), and an "Abstract Communication Layer" (ACL, Layer 3). The ACL is responsible for sending and receiving of context data and other types of information.

On the physical layer, a time division multiplex collision avoidance protocol is used, based on fixed slots and strict synchronisation of all Smart-Its within communication range. Thereby a wake-up on traffic procedure is avoided, as it is problematic on a network channel that is also used by various consumer devices that inevitably create noise.

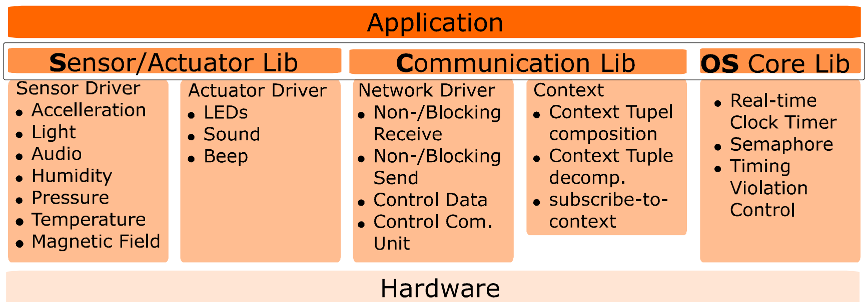
5. Application Development

5.1. Development support

Smart-Its applications are largely based on processing of sensor input, computation and communication of context, and reaction to particular events. The development is supported by a variety of tools, including a standard C compiler for the embedded processor and a wireless programming tool ('AirProg') both integrated into an IDE-like environment, and debugging tools. The AirProg tool lets an application developer download compiled code onto a Smart-It without the need for a wired connection. This means specifically that already deployed Smart-Its can be reprogrammed in situ.

The steps involved in application programming are:

1) write code using the software APIs
2) compile



3) download code to the Smart-Its wirelessly
4) debug with the SmartSpy tool: watch sensor output and incoming communication data

The following is a typical piece of code for processing of sensor events, in this case computing and broadcasting the sound level measured on a Smart-It:

```
if (get_audiosamples(20))    // 200ms sampling
{
   l=calc_audio_volume();    //  calculate  avg.
                             volume from values
   if (l >10) RED_LED_ON;    //  sound  detected?
                                   ->Red light
   else  RED_LED_OFF;
   AUDIO_ADD(l);             // add this context
                             to outgoing packet
   ACLsendPacket(50);        // and send it non-
                               blocking
```

5.2. Application experience

A batch of 160 Smart-Its devices are currently in use for research into networked smart objects conducted by various research labs across Europe. The platform has successfully enabled application research by ourselves as well as by others. For instance, Smart-Its devices have been used to demonstrate a new interaction technique for connecting of objects by subjecting them to the same movement (e.g. by shaking them together) [9]. In another research effort they have been used to build smart furniture that guides the assembly process, demonstrating closure between plans existing in the virtual world and action carried out in the real world [10].

6. Related Work and Summary

The Smart-Its technology concept is distinct in providing an embedded computing and communication platform for the augmentation of mundane objects. Related work on augmentation of everyday objects has either been in the form of one-off prototypes such as the Mediacup [2], or limited in functionality to passive provision of digital identity [5]. Our work in contrast can be viewed as generalizing the technology build into the Mediacup, and as enabling objects to be active and contextually aware nodes in digital networks.

The Smart-Its platform shares similarities with the Berkeley Motes which have come into widespread use for research into wireless ad hoc sensor networks [11]. However Motes are targeted at sensor data collection in large-scale networks and hence place more emphasis on networking, communication and data propagation, and less on support for sensor-to-context processing. Motes are viewed as primary devices, and networks of motes are homogeneous. Smart-Its in contrast are designed for adaptation to particular objects, for networking of very diverse smart objects. Smart-Its also put more emphasis on minimizing energy consumption, in comparison trading of more computing and communication power to enable deployment with smaller footprint and longer lifetime.

There is further work related to ours on certain aspects. For instance, the Sentient Computing project also uses pervasive sensors to give the physical world a digital presence, however primarily based on location tracking and backend processing [12]. In a different way related is the Phidgets work which like ours seeks to provide a platform to lower the hurdle for prototyping with interactive physical components, however focussed on explicit tangible user interfaces rather than unobtrusively embedded systems [13].

To summarize our contribution, Smart-Its introduces a distinct technology concept in which computing is decentralized and placed in the background of physical artefacts. The concept is fully implemented in a platform that has become deployed in a number of different research efforts and groups for investigation of smart object applications.

Acknowledgements

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).

7. References

[1] H.-W. Gellersen, A. Schmidt, M. Beigl. Multi-Sensor Context-Awareness in Mobile Devices and Smart Artifacts, *Mobile Networks and Applications (MONET)*, Kluwer, Oct 2002.

[2] Beigl M., Gellersen H., Schmidt A. MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects, *Computer Networks*, Vol. 35, No. 4, March 2001, Elsevier, pp 401-409

[3] Siio, T. Digital Decor. http://siio.ele.eng.tamagawa.ac.jp/projects/decor/index.html

[4] Van Laerhoven, K., Schmidt, A. and Gellersen, H. Pin&Play: Networking Objects through Pins. *Proc. Ubicomp 2002*, Gothenburg, Sept 2002, pp. 219-229.

[5] Brock, D., Sarma, S., and Ashton, K. The Networked Physical World – Proposals for Engineering The Next Generation of Computing, Commerce & Automatic Identification. Technical Report MIT-AUTOID-WH-001, MIT Auto-ID Center, 2000.

[6] Römer, K., Schoch, T., Mattern, F. and Dübendorfer, T. Smart Identification Frameworks for Ubiquitous Computing Applications**.** *Proc. PerCom 2003*, Fort Worth, March 2003, IEEE Press.

[7] The Smart-Its Project. http://www.smart-its.org/

[8] Weiser, M. and Seely-Brown, J. The Coming Age of Calm Technology. In *Denning, P.J. and Metclafe, R.M. (eds.) Beyond Calculation: The Next Fifty Years of Computing*, Copernicus, Heidelberg, Germany, 1998.

[9] Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M. and Gellersen, H. Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts. *Proc. Ubicomp 2001*, Atlanta, USA, October 2001.

[10] Antifakos, S., Michahelles, F. and Schiele, B. Proactive Instructions for Furniture Assembly. *Proc. Ubicomp 2002*, Gothenburg, Sweden, Sept. 2002.

[11] Estrin, D., Culler, D., Pister, K. and Sukhatme, G. Connecting the Physical World with Pervasive Networks. *IEEE Pervasive Comnputing*, Vol. 1, No. 1, pp. 59-69.

[12] Addlesee, M. et al Implementing a Sentient Computing System, *IEEE Computer*, vol. 34, no. 8, Aug. 2001, pp. 50-56.

[13] Greenberg, S. and Boyle, M. Customizable physical interfaces for interacting with conventional applications. *Proc UIST 2002*, Paris, Oct 2002, ACM Press.