Lars Erik Holmquist
Viktoria Institute

Hans-Werner Gellersen, Gerd Kortuem, Albrecht Schmidt, and Martin Strohbach
Lancaster University

Stavros Antifakos, Florian Michahelles, and Bernt Schiele
ETH Zurich

Michael Beigl
Universität Karlsruhe

Ramia Mazé
Interactive Institute

Building Intelligent Environments with Smart-Its

Smart-Its are self-contained, stick-on computers that attach to everyday objects. These augmented objects become soft media, enabling dynamic digital relationships with users and each other.

In the Smart-Its project, we are developing technology to realize a vision of computation everywhere, where computer technology seamlessly integrates into everyday life, supporting users in their daily tasks. By embedding sensors, computation, and communication into common artifacts, future computing applications can adapt to human users rather than the other way around. However, it's currently difficult to develop this type of ubiquitous computing[1] because of the lack of toolkits integrating both the required hardware and software. Therefore, we are creating a class of small computers—called Smart-Its (http://www.smart-its.org)—equipped with wireless communication and sensors to make it possible to create smart artifacts with little overhead.

The Smart-Its project is a collaboration between six partners in five countries: ETH Zurich (Switzerland), Interactive Institute (Sweden), Lancaster University (UK), the University of Karlsruhe (Germany), the Viktoria Institute (Sweden), and VTT (Finland). The project ran from 2001 to 2003 as part of the European research initiative called the Disappearing Computer (http://www.disappearing-computer.org). In this proactive research program, the European Union's funding agency for Future and Emerging Technologies supports the exploration of how information technology can be diffused into everyday objects and settings. There are 16 projects in the program, developing everything from computational fibers to computer-augmented office furniture. By supporting a critical mass of innovative and far-reaching projects, the Disappearing Computer program hopes to create synergy effects that could not have been achieved by a smaller number of separate projects.

The Smart-Its vision fits well into the overall goal of the Disappearing Computer initiative. We see a future in which mundane everyday artifacts become augmented as soft media, and thus become able to enter into dynamic digital relationships with users and with each other. The long-term goal of the Smart-Its project is providing a platform that developers and researchers can use to explore future applications, but with much less overhead than currently required. As Smart-Its and other similar physical construction kits stabilize, they should become as easy to use as today's GUI toolkits for desktop computers. When we reach this stage, we believe there will be an explosion of new innovations; just as fast graphics and GUI toolkits have enabled a multitude of new interactive applications for desktop computers. To this end, we are in the process of releasing all necessary information about Smart-Its into the public domain so that others can benefit from the work.

Smart-Its platform

You can think of a Smart-Its as a small, self-contained, stick-on computer that users can attach to objects much like a Post-It note. To ensure flexibility, we used a modular approach when designing the hardware. A Smart-Its consists of a core board with a wireless transceiver to let the device communicate with other Smart-Its, plus a sensor board that gives the Smart-Its data about its surroundings. For more information about the Smart-Its architecture, see the "The Smart-Its Hardware" sidebar. The standard sensor board has five sensors: light, sound, pressure, acceleration, and temperature. For specific purposes, we could add other sensors—such as a gas sensor, a load sensor, or a camera for receiving images. We have also developed several APIs to aid application development. For instance, there is a communication API to facilitate communication between Smart-Its and other devices. Another example is the perception API, which allows the abstraction of low-level sensor data to higher-level concepts, so-called percepts. For more information about programming for the Smart-Its, see the sidebar "The Smart-Its Perception API" on page ??.

The major advantage of the Smart-Its platform is that it allows designers and researchers to construct responsive or intelligent environments with comparably little

Published by the IEEE Computer Society
0272-1716/04/$20.00 © 2004 IEEE
www.manaraa.com

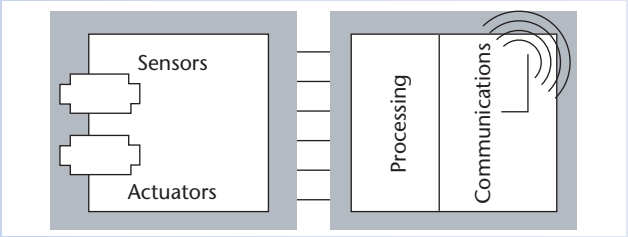
The Smart-Its Hardware

With the Smart-Its hardware (see Figure A), we provide generic devices for sensing, computing, and communication on a large scale. Currently, Smart-Its can be as small as 17 x 25 x 15 mm including sensors and battery, with a weight of about 8 grams (or 20 grams including an AAA battery). Other Smart-Its types are larger and simpler, aiming at ease of modification, customization, and reproduction for use as teaching material or as quick demonstrator projects. Over the course of the project, the Smart-Its hardware has evolved into smaller, more compact iterations, as shown in Figure B. A broad range of sensors and actuators can be integrated on the devices. The generic sensor boards can sense audio, light levels, acceleration, humidity, temperature, and pressure. Outputs include a speaker and LEDs. Additional sensors and actuators are attachable via standard I/O interfaces, allowing us to extend the capabilities with off-the-shelf components.

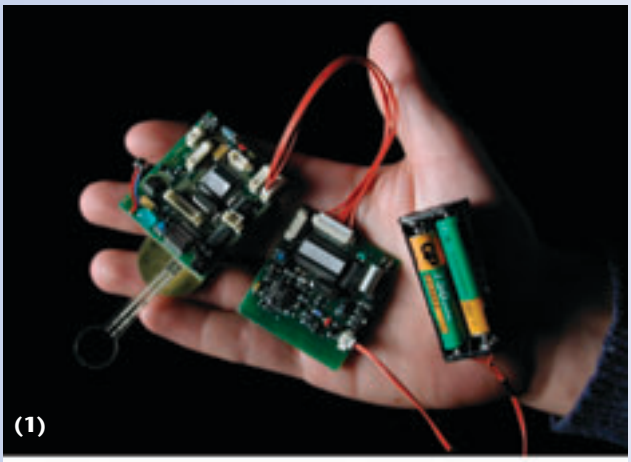
Smart-Its can be powered by a range of sources, from lithium coin cells, standard AAAs, or rechargeable batteries. The choice depends on the target artifact's shape and size, use, and projected lifetime. Smart-Its' on-board computing and storage allows it to process small programs without any assumption of computing infrastructure in the environment. For example, each Smart-It can interpret its own sensor values along with those gathered from other sensor devices. It can then communicate the interpreted values to other systems, such as a PC or other more high-powered processing unit. To minimize the requirements for regular user-based maintenance, Smart-Its introduces a new power-management concept based on sensed situations. The device controls power-safe modes of internal and attached external devices. Depending on the application—sensors used, processing required, and communication needs—a Smart-It's lifetime ranges from several days to one year.

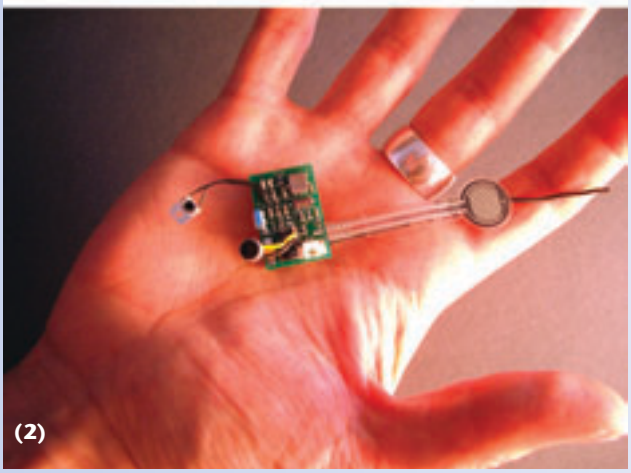
The Smart-Its also features efficient short-range communication through a wireless ad hoc network. Hardware requirements and constraints for the radio link design were size, energy consumption, full control on the physical layer, and midspeed transmission bandwidth (125 kilobits per second). Up to 1,024 nodes can request to be sent at the same time. An arbitrary number of Smart-Its receiving and 256 trying to send at the same time reduces the available network bandwidth by about 10 percent. In typical settings with a dozen Smart-Its, interference is below the noise on the channel and is therefore not measurable. To allow fast reaction on network changes, Smart-Its instantly communicates (average 12.6 ms) even from a switched-off state or when coming to a new environment without any need for configuration. High-precision automatic clock synchronization makes possible real-time reaction on events from other devices, or cooperative behavior of devices. This lets us compare high-speed signals such as audio or acceleration.

Development on Smart-Its is done in C. A wireless program download tool, analyzing tool, and database tool support programming. The download tool connects to the editor and compiler and transfers developed programs to the Smart-Its. The analyzing tool supervises the behavior of Smart-Its and helps debug applications. The database tool allows retrieval of information from all communication and exports the data to statistic tools for long-term evaluation.



A To ensure flexibility, the Smart-Its hardware consists of two separate components: a core board and a sensor board.





B During the course of the project, the Smart-Its hardware has moved through several evolutionary steps from (1) larger hardware to (2) smaller hardware.

overhead.[2] Typically, research projects that develop smart or context-aware objects require building a lot of custom hardware and software from scratch. The Smart-Its project addresses this issue by presenting a standardized hardware solution coupled with communication and sensing APIs. Future interactive systems

www.manaraa.com

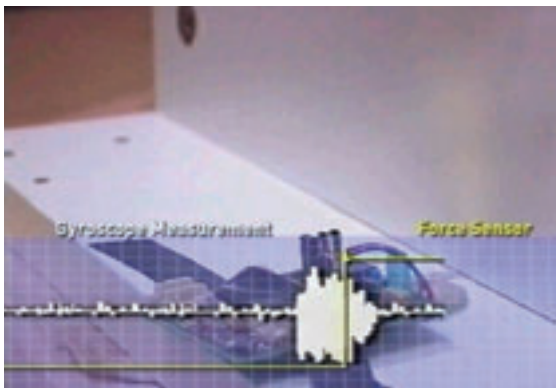
The Smart-Its Perception API

The Perception API (PAPI), which insulates application requests from internal sensor processing, provides uniform access to various kinds of sensors. PAPI addresses single perception among sensors located on the same Smart-Its board, collective perception among distributed Smart-Its boards, and out-band processing for large data streams. Unique universal identifiers distinguish the different boards, sensors, and features of sensor values. Each Smart-Its device is aware of its attached sensors and can share this information with other devices through wireless communication. PAPI offers a query mechanism for discovering available sensors. As a result, PAPI abstracts from internal sensor processing and provides a more convenient access to sensor readings by various methods.

Devices can access local sensor readings, referred to as single perception, via four methods. A single request enables a device to receive only one single sensor value, or allows it to poll a series of values. The condition-triggered notification method offers atomic feature value conditions (<, >, =) that specify when a particular sensor should send an interrupt to the calling device. All conditions are treated as one-time triggers. The continuous-subscription method offers a continuous feature value series on a timely basis. The constant stream method constantly streams feature data by using maximal bandwidth.

Collective perception enables Smart-Its to have remote access to sensors attached to other Smart-Its boards. In such cases, a distance measure can set the sensor discovery range. A value of 0 represents all local sensors on a Smart-Its board, whereas 255 means all accessible boards. The values in between represent corresponding physical distance measures.

Finally, a so-called out-band functionality enables an application to reserve the entire radio band for streaming intensive data, such as video or high-quality audio, to external back-end resources. However, because out-band functionality blocks the entire Smart-Its communication function, its use should be limited to short periods.



1 We attached Smart-Its to the pieces of a flat-pack furniture kit and to some of the assembly tools, letting the system monitor the user's progress.

will require a multitude of sensors and distributed computation. With Smart-Its and other physical prototyping toolkits, we can explore such systems today but without building everything from scratch. Smart-Its is only one of several similar physical prototyping tools that researchers can use to get applications up and running quickly. See the sidebar "Other Physical Prototyping Kits" for more information about prototyping.

A major part of the Smart-Its project has been to develop and integrate the hardware and software. We also developed several sample applications to demonstrate the vision of computation embedded in the world. These applications vary from solving a mundane but intricate task such as furniture assembly, to lifesaving applications such as supporting avalanche rescue teams. However, none of these should be considered the killer app that will accelerate the presence of ubiquitous computing. Instead, they act as a vehicle to explore the possibilities of Smart-Its and similar technology. These application demonstrations have also served to validate the technology itself during the development process.

Proactive furniture assembly

Printed handbooks, instructions, and reference manuals for items such as a laptop, furniture, or VCR all have a common fault: Users rarely read them. It's not simply that people are lazy, overconfident, and believe they have no time to spend bothering with instructions. The quality of the instructions themselves is often poor. Furthermore, there is a divide between the instructions on the printed page and the real object. What if the instructions could be better integrated with the actual task?

With one Smart-Its application, shown in Figure 1, we propose a framework for proactive guidance that aims to overcome limitations of today's printed instructions.[3] We chose as our example the task of assembling a piece of flat-pack furniture—a seemingly simple process but one that often goes wrong. By attaching computing devices and multiple sensors onto the parts of the unassembled furniture, the system can recognize the actions of the user and determine the current state of the assembly process. It can then suggest the next most appropriate action at any point in time. We used the sensors in Smart-Its devices attached to the furniture to gather information about the user's actions. A separate laptop collected the information through wireless communication with the Smart-Its.

To assist the user, we need to create a plan that represents the different ways to assemble the piece. The plan consists of states and interconnecting actions, similar to those in a finite state machine. We can divide the actions required to assemble the furniture hierarchically into partial actions. We can detect these partial actions with the accelerometers and force sensors of the Smart-Its attached to the boards. Furthermore, a gyroscope in the screwdriver supplies additional information. Using a large enough number of sensors, we achieve high-quality perception and provide precise instructions to the user.

One important problem is how to present the information. Originally, we used a laptop screen to present the user's progress. However, this still presents a division between the instructions and the actual object. Instead, to integrate the instructions into the objects perfectly, we mounted LEDs directly on the boards, shown in Figure 2. By using different colors and blinking patterns, we can provide a limited but precise set of instructions: The underlying principle is to enhance the physical object's static affordances by additional dynamics hints.[4] User studies revealed that the integration of the instructions into the objects provides a significantly better understanding than presenting descriptions on a separate screen.

المنارة للاستشارات
www.manaraa.com

Other Physical Prototyping Kits

A main feature of intelligent environments is the ability to sense different contexts, and there are several frameworks for creating context-aware applications. For instance, the context toolkit[1] presents a software framework to support the implementation of context-aware applications. This framework enables exploration of context-aware design space, particularly in dealing with ambiguous or inaccurate sensor data. However, the level of abstraction is high, and this toolkit doesn't present the integrated hardware-software solution that Smart-Its represents.

As small computing devices that integrate physical interaction and wireless networking, Smart-Its shares several features with other wireless sensor platforms. The Berkeley Motes platform was originally developed for collection of sensor data in large-scale ad hoc networks but recently researchers have considered it for ubiquitous computing applications.[2] Motes places more emphasis on networking, communication, and data propagation. To support this, Motes provides the Tiny OS for very small networked devices and a protocol stack that supports ad hoc routing in large networks—features not available in Smart-Its. Smart-Its, on the other hand, provides distinct support for rapid assembly and design iterations over device concepts.

Other small-computing platforms to which Smart-Its relates include the i-Bean from Millennial Net and MIT's Sensor Stack. The i-Bean is a self-contained, miniaturized computer with a digital I/O interface, analog-to-digital and digital-to-analog converters, and a radio frequency transceiver for bidirectional communication.[3] The i-Bean functions as a data-acquisition and processing device and is primarily used for healthcare applications. The Sensor Stack platform for high-density wireless sensing stresses device modularity and fast prototyping.[4] The modules are fabricated in a 3D stackable manner, allowing for the development and easy addition of extra panels.

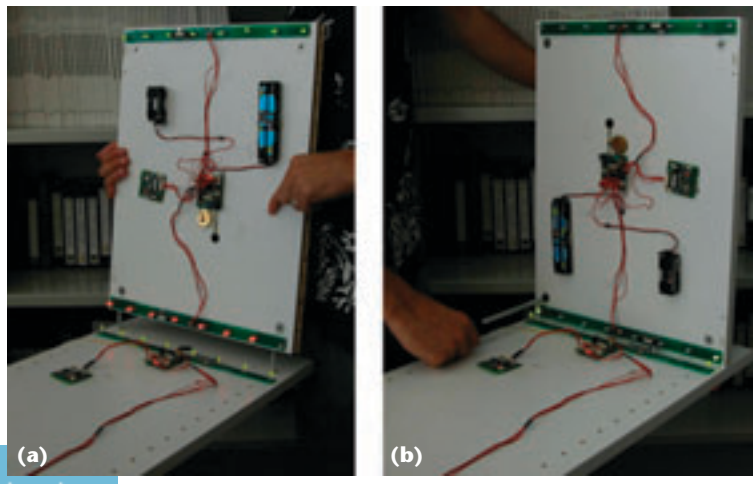
Two European platforms to which Smart-Its relate are the Eyes sensor node,[5] which is under development by a European project consortium for sensor network research, and the BTNode developed at ETH Zurich.[6] The BTNode Bluetooth-based sensor node stresses interoperability as the main design goal. To support Bluetooth networking, BTNode uses a more powerful processor and more memory than most other sensor-node platforms. Among these platforms, the Smart-Its technology is unique in its focus on rapid prototyping and development of hardware and software.

Several toolkits exist for prototyping tangible input and output systems not based on wireless communication. Recent examples include Phidgets[7] and iStuff.[8] Such toolkits help developers create customized input and output devices for a variety of applications running on a dedicated computer. Examples of input devices include physical buttons, sliders, and tag readers. Output devices include motors, buzzers, and lights as well as a computer screen. Such toolkits could potentially expand the vocabulary of interactive applications beyond the mouse and keyboard.

References

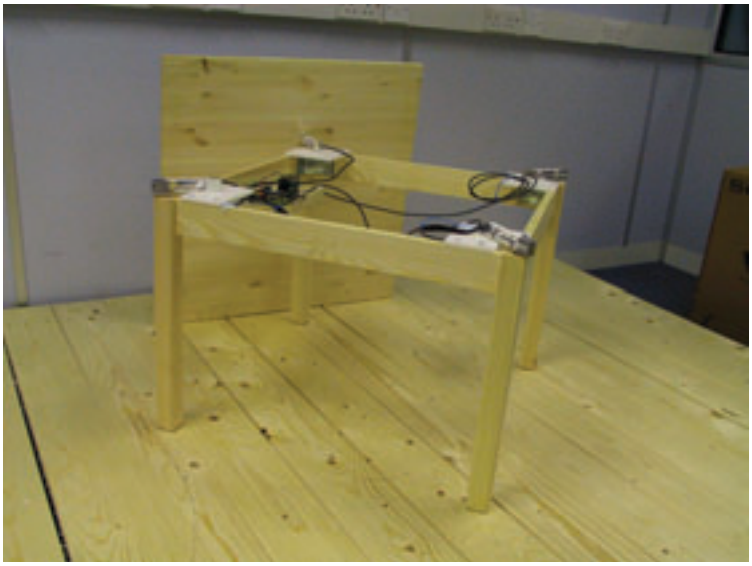
1. A.K. Dey et al., "Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Human-Computer Interaction (HCI) Journal*, vol.16, no. 2-4, 2001, pp. 97-166.
2. J. Hill et al., "System architecture directions for networked sensors," *Architectural Support for Programming Languages and Operating Systems*, 2000, pp. 93-104.
3. S. Rhee and S. Liu, "An Ultra-low Power, Self-Organizing Wireless Network and Its Applications to Noninvasive Biomedical Instrumentation," *Proc. IEEE/Sarnoff Symp. Advances in Wired and Wireless Communications*, IEEE Press, 2002.
4. J. Barton et al., "Miniature Modular Wireless Sensor Networks," *Adjunct Proceedings 4th Int'l Conf. Ubiquitous Computing* (Ubicomp), Viktoria Institute, Gothenburg, Sweden, 2002, pp. 25-26.
5. S. Dulman and P. Havinga, *Operating System Fundamentals for the EYES Distributed Sensor Network*, Utrecht, 2002.
6. J. Beutel et al., *Thiele: Bluetooth Smart Nodes for Ad-Hoc Networks*, TIK Report No. 167, ETH Zurich, April 2003.
7. S. Greenberg and C. Fitchett, "Phidgets: Easy Development of Physical Interfaces through Physical Widgets," *Proc. 14th Ann. ACM Symp. User Interface Software and Technology* (UIST 01), ACM Press, 2001, pp. 209-218.
8. R. Ballagas et al., "iStuff: A Physical User Interface Toolkit for Ubiquitous Computing Environments," *Proc. ACM Ann. Conf. Human Factors in Computing Systems* (SIGCHI), 2003, pp. 537-544.

Although this example concerns furniture, we can generalize it to many other assembly tasks, such as a car or airplane factory assembly line. For many safety-critical assembly and maintenance tasks, this type of recognition and verification of procedural actions might also prove useful. We predict the initial adoption of this technology happening in this area. Another possibility is quality control. Whereas it currently does not make economic sense to embed sensors in furniture that costs a few hundred dollars, it can pay off quickly when assembling an expensive



2 LED lights on the components guide the user in the assembly process: (a) rotation and (b) screwing.

www.manaraa.com



3 Load-sensing furniture has load cells installed in each corner, monitored by a central Smart-Its.

item (such as a car) or when maintaining a safety-critical part of a power plant. If even one mistake per item is avoided, this could easily justify the whole investment.

Load-sensing furniture

So what happens when the pieces have been assembled? If we leave the Smart-Its and sensors in the furniture, ordinary furniture can be made smart. One area in which we have found surprising potential is load sensing, which effectively means turning ordinary pieces of furniture into weight-measuring devices.[5]

Using Smart-Its, we embedded load-sensing technology in several pieces of furniture: a coffee table, a dining table, shelves, and drawers. The tables in particular are augmented using industrial load cells, as shown in Figure 3. We can unobtrusively install the cells between the tabletop and frame so that the tabletop rests on a load cell at each corner. In our latest version, we placed the boxes containing the load cells under the table legs, making augmentation of the table much more flexible. For each surface, a Smart-Its with a load add-on module interfaces with the load cells. The system samples and reads the values from all four load-sensing cells at about 200 Hz. Because we build the technology on top of the Smart-Its platform, each augmented object becomes a wireless sensor in the Smart-Its network.

By measuring the load on each corner, we can easily calculate the center of gravity's location on the surface. By observing how the center of gravity moves, we can detect interaction on the surface and recognize specific patterns. We can process this information into interaction primitives, such as tracking an object's position and weight or a finger's track as someone traces it over the surface. In essence, the entire table surface becomes a sensor system because we can precisely pinpoint the location and weight of all objects placed on it. Together, several load-sensing chairs and tables could collect detailed information about the activity in an environment.

On the basis of these interaction primitives, we developed several applications. For instance, one application lets the user use a surface, such as a tabletop, to manipulate a mouse pointer on a computer screen. The movement of the finger or of an object on the surface—captured in the trace interaction primitive—is then converted in increments for the mouse movement in a conventional system. In other words, instead of manipulating a mouse on the table, the user can regard the entire table as a pointing device. This could be quite convenient when users must control an electronic device, such as a television set, but don't wish to use yet another remote control or pointing device.

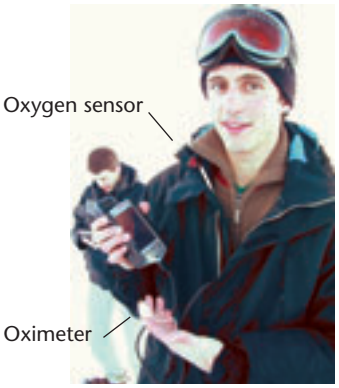
Supporting avalanche rescue

Smart-Its is a general platform suitable for many types of applications, including mobile and wearable computing systems. For example, the prototype of the avalanche lifeguard system, A-Life, incorporates Smart-Its technology.[6] The goal of the A-Life project was to explore the applicability of wearable sensors for mountaineers, skiers, and snowboarders. We involved domain experts such as emergency physicians, professional rescuers, and avalanche researchers early in the development process.

In alpine areas, many landslides each year cause accidents and even deaths. The time it takes to find and extricate victims is extremely crucial: Once a person is buried, the survival chances drop dramatically after the first 15 minutes. Current technology, such as radio-based tracking systems, can only offer the rescue team information on the location of a single victim at a time. However, statistics show that in many cases there are multiple victims, and the order in which rescuers save them can be crucial. Some victims might survive for hours, whereas others only have minutes. If rescuers could find a means to address the most urgent cases first, they might save many additional lives.

The A-Life system, shown in Figure 4, addresses this issue using wearable Smart-Its-based sensors. These sensors measure vital signs and environmental conditions of avalanche victims; the system then broadcasts the information to rescuers. Visualizing this information will help rescuers in the field separate victims by urgency and coordinate the rescue process more efficiently. The Smart-Its technology helped in constructing a functional rapid prototype for a first evaluation with experts: Off-the-shelf oximeters, oxygen sensor, and accelerometer connect to a Smart-Its communication board that establishes a connection to a handheld computer (shown in Figure 5) and wireless connectivity among different prototype units. Different focus groups have used this prototype for participatory evaluations with practitioners in the field. In particular, we demonstrated it to avalanche researchers of the Swiss Federal Institute for Snow and Avalanche Research, to a practicing emergency physician, and to skiers at an avalanche rescue-training course.

المنارة للاستشارات
www.manaraa.com



4 The A-Life system uses an Oximeter and other sensors to provide rescue teams with updated information about avalanche victims.

5 The rescue team's interface is PDA-based and gives instant access to vital information to aid in their work.

6 To create a smart restaurant environment, we augmented several everyday artifacts with Smart-Its.

7 Together, the items in the smart restaurant installation created an interactive environment where visitors could interact with smart objects.

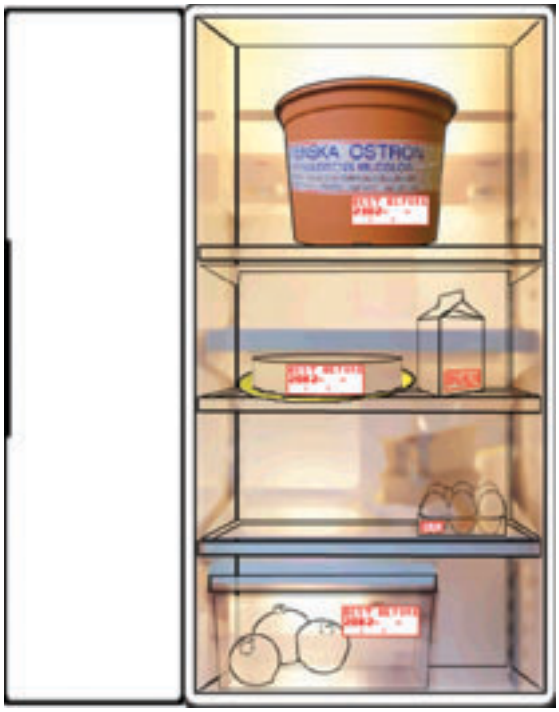
Generally, we received positive feedback for the idea of applying wearable sensors to avalanche rescue, as this was a completely new idea for all participants. We had an interesting discussion with emergency physicians about the appropriateness of the different sensing technologies. The avalanche researchers suggested recording sensor values over time for analyzing the process of an avalanche accident, much like black-box devices do in airplanes. Ethical issues initiated a discussion on whether a device should suggest to rescuers which victim to rescue first. The Smart-Its technology prototype enabled us to show the opportunities of sensing technology for avalanche rescue before we actually develop the final product. We believe that Smart-Its also could help in prototyping many other wearable applications.

Envisioning the smart restaurant

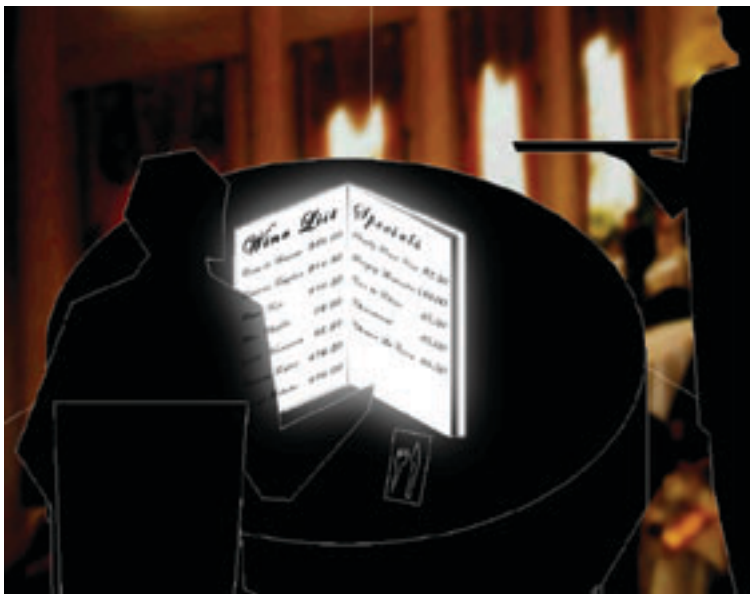
What would it be like to have a complete environment augmented with smart technology? In a collaboration between an interaction designer and a technical team, we created a set of demonstrators that used Smart-Its in conjunction with various presentation techniques to envision a complete future workplace.[7] We presented the demonstrations as an interactive exhibit, where visitors could manipulate different augmented objects to trigger various responses. The demonstration made use of actual sensor values derived from Smart-Its attached to the objects. To flesh out the environment, we provided more expansive parts of each scenario, presented as animations that visitors' actions would trigger.

We chose restaurants as the domain. A restaurant is a dynamic environment where people act in many different roles—as guests, waiters, chefs, and so on—performing different tasks, such as ordering from a menu, serving a dish, or preparing the food. These activities rely on a diverse set of artifacts, like kitchen utensils, food items, and furniture. All this seemed to make restaurants a perfect environment for Smart-Its augmentation (see Figures 6 and 7).

www.manaraa.com



8 Animations show aspects of the demonstrations that we did not directly implement in physical artifacts.



9 We used animation to illustrate how the status of food items could be directly reflected in the cost for customers.

In the first scenario, as illustrated in Figure 8, we showed how sensitive food items could keep track of their lifecycle status, and how they could communicate in internal and external networks. In the demonstration, we augmented a box of oysters in a refrigerator with a Smart-Its. When visitors opened the fridge door, an image of the oyster box would appear on a separate screen. A dynamic best-before label on this animated box would change according to how long the fridge door was kept open, the shorter the time indicated on the label before the oysters expired. This best-before value changed in real-time but was simulated because we did not have access to a sufficiently complex set of sensors to correctly determine the oysters' freshness.

Instead, we simply determined if the fridge door was open by using the light sensor in the Smart-Its. If the visitor kept the door open too long, an animated sequence would run, showing how new oysters were arriving in a delivery truck and a dock at the harbor. The idea was that if all oyster boxes were augmented, they could start an internal negotiation of their value. For example, a box that would soon expire might, if there were fresh boxes arriving at the local distributor, be prepared to lower its asking price quite significantly to restaurateurs shopping for oysters.

The second scenario, shown in Figure 9, illustrated how the status of food items could be directly reflected in the cost for customers. Visitors were invited to manipulate a bottle of wine, augmented with a Smart-Its. The sensors in the bottle would monitor temperature, light exposure, angle and movement used to calculate a decay curve reflecting if the wine had been correctly stored. When the bottle was shaken—which would release sediment and thus degrade the quality—an animated menu would appear. The prices of the different wines on the menu would change dynamically so that the price of a bottle that was mistreated might go down, whereas the price of well-treated bottles would go up.

If the user put the bottle back in the correct position for storage, its price would eventually stabilize and go up again, reflecting that the sediment had settled. We did not measure the actual sediment, but used the Smart-Its' accelerometer to detect movements and position. An accompanying video showed how the bottles in the wine cellar would negotiate a sort of stock market value and how the current menu would be advertised on an external billboard.

The final scenario showed how several objects could collaborate and share data to support a task. It took the form of an interactive order preparation where Smart-Its-augmented items included a piece of cheese and a wine bottle. Visitors would place the items on a tray, and by simply moving the tray to the serving counter, the items would become grouped together and open a communication channel. The grouping was determined by comparing the movement of the objects as read by the Smart-Its' accelerometers. Only the objects on the tray would share the same values. This is similar to the Smart-Its friends technique, where users would hold objects together and shake them to explicitly create a grouping.[8]

When the objects had been grouped, the objects themselves could start to determine the optimal serving conditions. The cheese requires a correct temperature, the wine must settle, and so forth. An animation showed these values as they changed, but because temperature changes much too slowly for use in an exhibition setting, we simulated some numbers. When all

www.manaraa.com

items were ready, the Smart-Its devices would signal the waiter and an animation would show how the waiter could now serve the completed order.

Taken together, this suite of presentations showed how sensors, communication, and computation could augment and support several objects and tasks. By mixing real sensor data and visual scenarios, we created a complete vision of an intelligent environment of the future, with which visitors could interact.

10 In the future, a multitude of everyday objects could be equipped with embedded sensing, computation, and communication capabilities.

Future Smart-Its technology

With Smart-Its, we have taken the first steps toward a platform for creating intelligent environments like the kind shown in Figure 10. As shown in the application examples, such environments could potentially support users in many different tasks, from the mundane assembly of a piece of furniture to life-and-death situations in the field. With the increased proliferation and availability of Smart-Its and other platforms for creating computationally augmented physical environments, we will see a lot of experimentation that could lead to products that enter the mainstream. In fact, this is already happening. A multitude of smart consumer products that use this type of sensor and communication technology are in development at research labs all over the world. It's only a question of time before we can buy them in the shops.

But it is obvious that there still must be several key advances before this type of technology can easily be embedded in everyday objects. In particular, the size must shrink, as must the cost, to make possible the realistic inclusion of this kind of technology in consumer products. Power consumption must become significantly lower; if only because users will not want to change batteries in their sensing tables or recharge their smart coffee cups very often. For prototyping and product development, standardized tools and APIs must make application development easier. Finally, if we envision a world with a multitude of smart objects that interoperate and communicate, we must create common communication protocols and agree upon semantics for data sharing.

Enabling platforms such as Smart-Its allows us to investigate these important issues. By letting researchers quickly build and evaluate smart and context-aware applications, they can give us a glimpse of the intelligent environments of the future. ■

Acknowledgments

We thank all members of the Smart-Its consortium for their role in the collaborative development of the Smart-Its applications. In particular, contributors to the implementation of the platform and demonstrators include Timo Ahonen, Christian Decker, Lalya Gaye, Peter Ljungstrand, Magnus Nilsson, Tobias Rydenhag, Daniel Spanagel and Nicolas Villar. Hanna Landin created the video from which Figure 10 is taken. The Smart-Its project is funded in part by the Commission of the European Union under key action "Future and Emerging Technologies" (contract IST-2000-25428), and by the Swiss Federal Office for Education and Science (contract BBW 00.0281). Lancaster's contribution was made possible with additional support from the UK Engineering and Physical Science Research Council as part of the Equator project (grant GR/N15986/01).

References

1. M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 265, no. 9, 1991, pp. 66-75.
2. H. Gellersen, et al., "Physical Prototyping with Smart-Its," to appear in *IEEE Pervasive Computing, October-December 2003.*
3. S. Antifakos, F. Michahelles, and B. Schiele, "Proactive Instructions for Furniture Assembly," *Proc. 4th Int'l Conf. Ubiquitous Computing* (Ubicomp), Springer Verlag, vol. 2498, 2002, pp. 351-359.
4. F. Michahelles et al., "Instructions Immersed into the Real World How Your Furniture Can Teach You," *Adjunct Proc. 5th Int'l Conf. Ubiquitous Computing* (Ubicomp), Seattle, Washington, 2003, pp. 155-156.
5. A. Schmidt et al., "Context Acquisition based on Load Sensing," *Proc. 4th Int'l Conf. Ubiquitous Computing* (Ubicomp), Springer Verlag, vol. 2498, 2002, pp. 333-351.
6. F. Michahelles et al., "Applying Wearable Sensors to Avalanche Rescue: First Experiences with a Novel Avalanche Beacon," *Computers & Graphics*, vol. 27, no. 6, 2003, pp. 839-847.
7. L.E. Holmquist, R. Mazé, and S. Ljungblad, "Designing Tomorrow's Smart Products: Experience with the Smart-Its Platform," *Proc. Designing for User Experience* (DUX 03), ACM Press, 2003, electronic publication.
8. L.E. Holmquist et al., "Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts," *Proc. 3rd Int'l Conf. Ubiquitous Computing* (Ubicomp), Springer Verlag, vol. 2201, 2001, pp. 116-122.

www.manaraa.com

Lars Erik Holmquist *is leader of the Future Applications Lab, a research group at the Viktoria Institute in Gothenburg, Sweden. His research interests include ubiquitous computing, mobile services, and information visualization. Holmquist received a PhD in informatics from Gothenburg University.*

Hans Gellersen *is a professor of interactive systems in the department of computing at Lancaster University. His research interests include ubiquitous computing and human-computer systems that take the real world into account. Gellersen received a PhD computer science from the University of Karlsruhe.*

Gerd Kortuem *is a lecturer in computer science at Lancaster University. His research interests include user interface technologies, ubiquitous computing, wearable computing, and software engineering. Kortuem received a PhD in computer science from the University of Oregon. He is a member of the IEEE Computer Society and the ACM.*

Albrecht Schmidt *is a researcher with the media informatics group at the University of Munich. His research interests include novel user interfaces and new forms of interaction enabled by ubiquitous computing. Schmidt received a PhD in computer science from Lancaster University.*

Martin Strohbach *is a PhD student at Lancaster University. His research interests include development support for smart objects. Strohbach received an MSc in context aware systems from the University of Karlsruhe.*

Stavros Antifakos *is a PhD student in the perceptual computing and computer vision group at ETH Zurich. His current research interests include perceptual computing with a diverse set of sensors, ambient displays, and applications for ubiquitous computing environments. Antifakos received the equivalent of a Master's degree in computer science from ETH Zurich.*

Florian Michahelles *is a PhD student in the perceptual computing and computer vision group at ETH Zurich. His research interests include participative design of wearable computing applications with end users and self-organizing sensors for perceptual computing. Michahelles received an MSc in computer science from Ludwig-Maximilians Universität München, Germany.*

Bernt Schiele *is an assistant professor at the computer science department at ETH Zurich. His research interests include computer vision, perceptual computing, statistical learning methods, wearable computers, and integration of multimodal sensor data. Schiele received a PhD in computer science from INP Grenoble, France. He is a member of the IEEE Computer Society and the ACM.*

Michael Beigl *is a senior research assistant at the University of Karlsruhe. His research interests include people at the center of communication and information technology, with specific interest in novel information appliances, electronic artifacts, mobile and ubiquitous networks, human-computer interaction, and context awareness. Beigl received a PhD in computing from the University of Karlsruhe.*

Ramia Mazé *is the director of the PLAY studio of the Interactive Institute in Sweden, which focuses on new materials and methods in the design of ubiquitous computing applications. Her research interests include user-centered methods and strategies for prototyping new systems, products, and concepts. Mazé received an MA in computer design from the Royal College of Art, London.*

Readers may contact Lars Holmquist at Viktoria Inst., Box 620, SE 405 30; Goteborg, Sweden; leh@viktoria.se.

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

www.manaraa.com