**Editors: M. Brian Blake** • *mb7@georgetown.edu*
**Michael N. Huhns** • *huhns@sc.edu*

# Embedded Interaction

## *Interacting with the Internet of Things*

The Internet of Things assumes that objects have digital functionality and can be identified and tracked automatically. The main goal of embedded interaction *is to look* at new opportunities that arise for interactive systems and the immediate value users gain. The authors developed various prototypes to explore novel ways for human-computer interaction (HCI), enabled by the Internet of Things and related technologies. Based on these experiences we derive a set of guidelines for embedding interfaces into our daily lives.

**Matthias Kranz**
*Technische Universität München*

**Paul Holleis**
*DOCOMO Euro Labs*

**Albrecht Schmidt**
*University of Duisburg-Essen*

Technological advances and new usage models can cause computing to undergo a stark transformation. Automatic object identification (such as RFID or Near Field Communication, and visual markers), ubiquitous connectivity, improved processing and storage capabilities, various new display technologies, sensor device availability, and decreasing hardware costs all lay the foundation for a new computing era. We can now build vehicles, devices, goods, and everyday objects to become a part of the Internet of Things. The combination of high-bandwidth connectivity, the availability of Internet-based services, and ubiquitous computing allows for communication, interaction, and information access everywhere and anytime to be embedded into anything. We call the resulting artifacts *netgets*, specialized networked gadgets with sensors and actuators that let users seamlessly manipulate digital information and data in the context of real-world usage.

Here, we present the underlying concepts of *embedded interaction*, the technological and conceptual phenomena of seamlessly integrating the means for interaction into everyday artifacts. Technically, this requires embedding sensing, actuation, processing, and networking into common objects. Conceptually, it requires embedding interaction into users' everyday tasks. The technical and conceptual perspectives are interlinked and aim to provide

1089-7801/10/$26.00 © 2010 IEEE

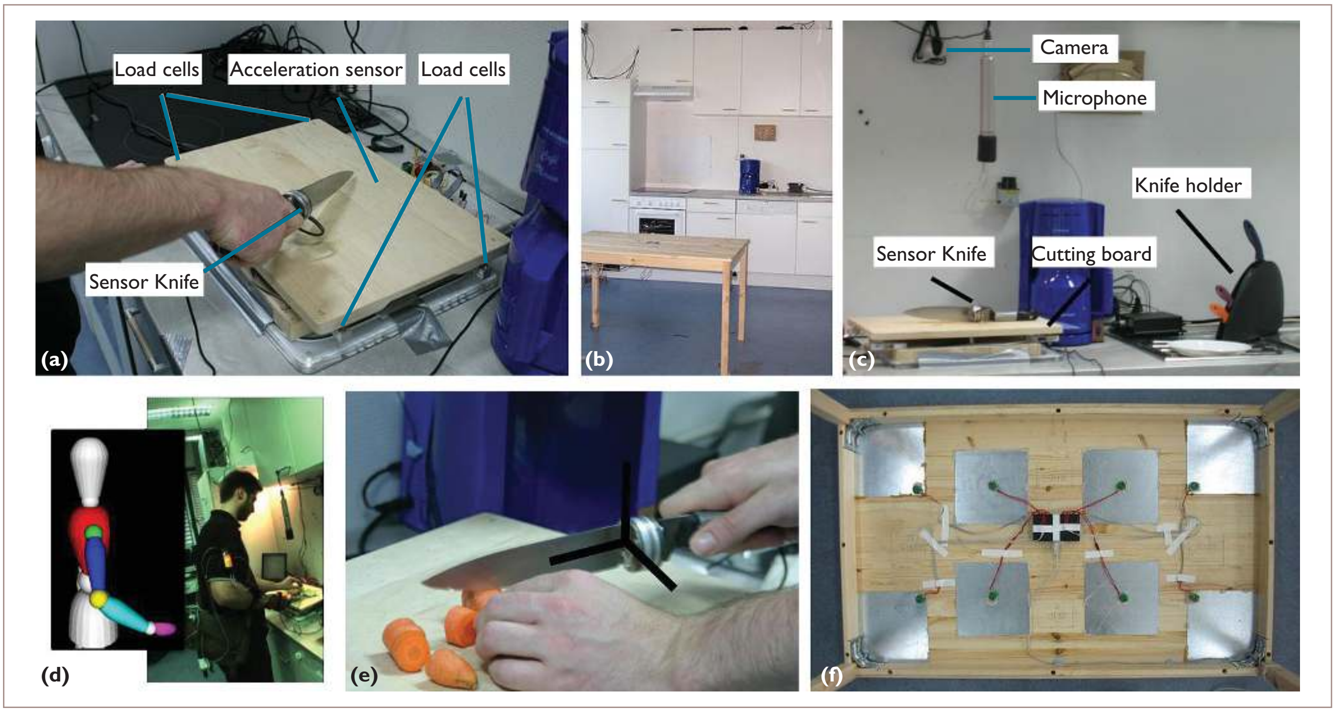

*Figure 1. Netget ecology in a kitchen environment. Device prototypes include (a) an instrumented cutting board acting as a mouse pad and scale; (b) the kitchen; (c) vision and sound-based activity detection; (d) activity recognition using body-worn sensors; (e) a sensor-augmented knife netget; and (f) a table with a capacitive sensing system for recognizing table-top interactions during meals.*

optimal support for users as they interact with information that isn't disrupting or distracting to their primary tasks and goals.

We also look at key enabling factors and accelerators for embedded interaction with the Internet of Things, such as technologies and toolkits. Several case studies outline embedded interaction's potential in an Internet of Things context. We've also developed several guidelines for developing embedded interactive systems and applications that are grounded in a variety of prototypes we have constructed and deployed over the past five years.

## Foundations and Case Studies

To realize the vision of embedded interaction with the Internet of Things, we must first identify and augment everyday objects with input or output facilities. This ecology of networked, self-configuring, and discoverable objects then constitutes a virtual overlay on the physical world. Additional knowledge emerges from both single objects and their individual states as well as from their relation to each other. This assumes several technical developments and creates new challenges. Let's look at four particular areas and their characteristic interaction methods that are closely related to the Internet of Things and embedded interaction concepts.

### Case 1: Context-Aware Kitchen Utilities

Kitchens are social places where families meet for cooking, discussions, and other interactions, so technology should stay unobtrusively in the background. Various netgets are suitable for activity recognition and interaction in everyday kitchen environments.

All netgets allow for seamless and unobtrusive interaction with everyday objects. The netgets in the kitchen scenario comprise a cutting board, acting both as scale and mouse interface, a video camera and microphone mounted above the working area, a sensor-augmented knife, and a sensor-equipped table (see Figure 1). Details are available elsewhere.[1–3]

The cutting board senses the weight of various foods processed on it as well as cutting actions during meal preparation. Additionally, it lets users control a computer system by employing the cutting board as a mouse pad, without the need for clean hands. The load cells below the board measure the weight and weight change of a finger moving over the surface in

*Figure 2. Touch sensors in clothing and accessories. We put (a) flexibly soldered sets of touch sensors (QProx QT110) into several device prototypes, including a (b) phone bag, (c) bicycle helmet, (d) and piece of clothing with different designs for touch areas.*

mouse mode, and also when food is cut with the sensor-augmented knife.

The knife features a three-axes-of-force and three-axes-of-torque sensor, commonly found in robotic arms. By analyzing the force and torque changes occurring when users cut food, this netget can determine the type of food being cut — for example, the blade's torsion against the handle when it cuts through food is characteristic for different food types, such as an apple versus a bell pepper. This lets the digital system infer the meal cooked and suggest additional ingredients or possible variations. Although the knife itself enables correct food detection, so does the sound of the cut, which is captured by a microphone placed above the cutting board. Because certain commercial scales can already visually identify food using image-processing algorithms, we use the camera only to log and annotate our experiments (see Figure 1c). The table detects how many people are having a meal and their interaction on the table top (see Figure 1f). This is done via capacitive sensors placed underneath that generate an electric field and measure its strength. When an object, such as a hand or finger, interferes with the electric field, the measured capacitance changes and can be used to detect proximity or touch.

This netget ecology lets digital systems infer the context and activities occurring in the kitchen. In addition, the intelligent environment can learn, recognize, and use behavior to provide a variety of novel services to household members, such as reminders and suggestions for improving nutrition. In the context of ambient assisted living, the ecology of netgets can detect deviations from normal behavior and offer assistance. Extending interaction beyond a single device enables networked systems to gather and share complex information to provide more natural interaction and services to human users and context awareness to digital systems.

## Case 2: Capacitive Touch Input on Clothes

Another project we worked on focused on connecting various pieces of clothing directly or via the Internet. We built a set of prototypical devices and garments to enable testing, demonstrations, and study applications using touch input on clothing. The prototyping platform we built uses capacitive sensing as described previously and is connected to our Embedded Interaction Toolkit (EIToolkit), which we describe in more detail later.[4]

The platform lets designers quickly add touch controls to nearly arbitrary clothing or objects, such as accessories and other devices. It also allows for a simple, on-the-fly remapping between controls and applications (such as a game, an N800 Internet tablet, or a home cinema application running on a local or remote PC). This functionality proved to be not only important for initial application development but also vital during user studies.

Figure 2 shows a selection of the prototypes we developed and studied, which included off-the-shelf phone bags and bicycle helmets with touch areas integrated into the design, gloves with different layouts of touch controls on the back, and an apron that prototyped different styles for touch areas and buttons.

To evaluate the wearable input, we conducted two user studies during which participants could try out the described prototypes. Whereas the first concentrated on gathering opinions and feedback about wearable computing in general and the prototypes in particular, the second focused on using the apron and its different controls.[4] We've integrated several results from these studies into the guidelines we discuss later.

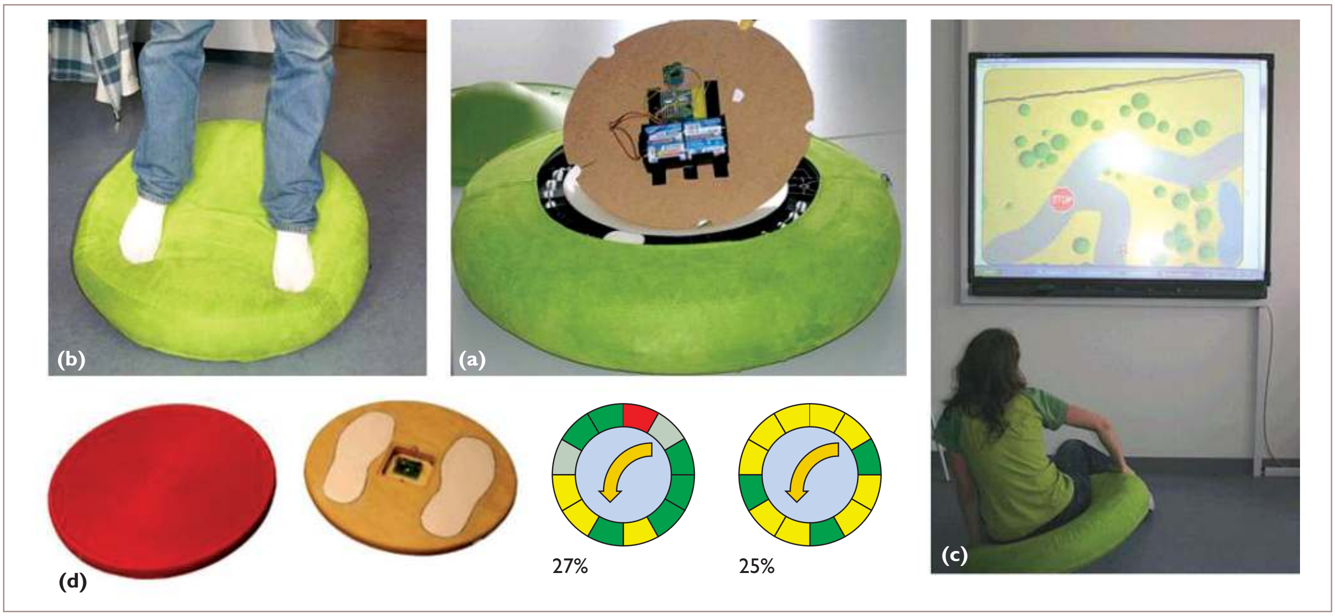

Figure 3. Embedded computing for sports and entertainment. We prototyped several projects, including (a) an Ikea cushion with built-in orientation sensors. A person can (b) stand or sit on it to control applications such as (c) an educational racing game. (d) The same technical setup enables an application used for physiotherapy and sports.

## Case 3: Embedded Computing for Entertainment and Sports

We employed an initial prototype of our EIToolkit[4] to create a platform for applications with a special focus on gaming and sports for educational purposes that used and combined non-standard input and output devices.[5] For one project, we used an off-the-shelf, flat Ikea balance cushion, which features a robust hemispherical base (see Figure 3).

The electronic hardware inside the cushion comprises a compass (measuring horizontal rotations), a 2D acceleration sensor (to measure tilting), a pressure sensor (for detecting people and hopping), a microcontroller with a communication module, and a large set of batteries for long-term use.

One example application we implemented was the Virrig Race Game, an "edutainment" application in which players control an onscreen car by tilting the cushion (see Figure 3c). At crossings, the game displays a multiple choice question players answer by tilting the cushion and clicking — that is, briefly hopping on the cushion as a physical translation of a mouse click. In an informal study using 20 primary-school children, we confirmed our assumptions that the game is fun and physically challenging, fosters collaboration, and supports at least short-term memorization, which is necessary to finish the game.

A second project using a very similar setup focused on sports and rehabilitation.[5–7] We augmented various fitness devices with the same or slightly adjusted sensor hardware as the cushion. As Figure 3d shows, the therapy top is a plastic or wooden disk with a round bottom. Users can choose from more than 30 different exercises targeted at strengthening and convalescing leg muscles and joints. Through the employed framework, we were able to use the existing setup and replace only the part providing audio-visual feedback to the user. The two right-most images in Figure 3d show the visual feedback the user receives according to how accurate his or her movements on the device are.

In conjunction with an RFID-based user authentication and recognition system, a graphical and simple-to-use editor lets the physiotherapist or coach specify new exercises along with restrictions (for example, using two therapy tops at the same time). The trainer can at any point review the recorded data offline and adapt the following session accordingly.

## Case 4: Small Embedded Objects

To further study small, Internet-connected devices, we designed a simple application that visualizes room occupation in a building (see Figure 4a). We built a digital version of the small posters you often see at office doors, where people specify their state, such as as "busy" or "out

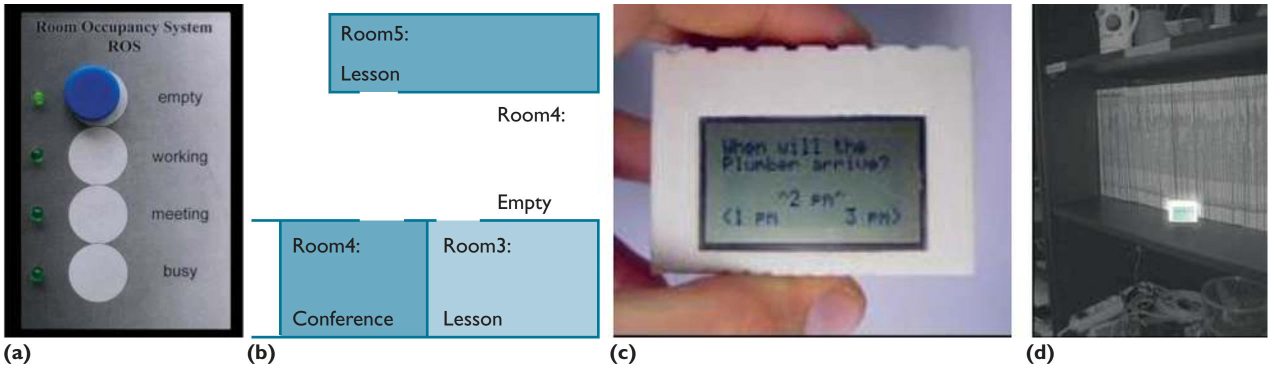

*Figure 4. Small embedded object prototypes. We created (a and b) an interactive room information system as well as (c and d) a wireless display that allows gesture input (only) and somewhat fades into the background if not in use.*

at lunch" with a magnet on the poster. Using hall sensors, the device can detect the magnet's position and then wirelessly transmit the state to an EIToolkit component connected to a central database. Thus, others can query this state via a simple Web application. Similarly, office owners could set their state remotely — for example, to indicate they've gone home straight after a meeting. However, this illustrates a design dilemma: because the magnets can't be moved remotely, the physical device can potentially show wrong information. Replacing the magnet with push buttons and feedback lights is an option, but departs from the original, known interaction using a magnet. A hybrid solution like the one Figure 4a shows can partially help, but will also potentially confuse the user.

Another embedded netget extends an idea by Tara Matthews and her colleagues[8] by adding a display to a wardrobe. In addition, each hook can sense whether a particular piece of clothing is present. Using weather information, the system can indicate and justify a suggestion. The example application could determine exactly what type of garment is hanging where using RFID-tagged clothes. If users already know what they want to wear, they can easily ignore the application's suggestions.

A toolkit, both in hardware and software, offers many advantages, including the variety of interactions we prototyped with the same technology. As an example, we developed small wireless displays (as Figures 4c and 4d show) and created various applications with them. The connected displays[9] let users with such devices easily and effortlessly vote, for example, on when or where to meet by simply turning their display to a specific side. The SkypeTUI prototype,[10] on the other hand, uses the same device to control the communication software Skype. We further generalized this concept to be able to use any device with an orientation sensor to control the user's possible state in any IM application.[11]

## Current Projects, Challenges, and Guidelines

Advancing research in various fields, especially ubiquitous computing, robotics, artificial intelligence, and human-computer interaction (HCI), leads to these disciplines converging. Middleware such as the Player/Stage[12] or Robot Operating System (ROS) from robotics, MundoCore[13] from pervasive computing, and Papier-Mâché from the HCI[14] disciplines can be used with embedded interaction to help integrate physical interaction, communication, and data exchange, enabling a holistic approach toward interaction with the Internet of Things.

We extracted a set of challenges from different projects, such as those we described previously, that people developing Internet of Things applications focused on embedded interaction are facing today. Furthermore, we derived various guidelines that will help ensure that future projects emphasize the most important aspects for embedding interaction with the Internet of Things already in the planning phase.

### Emerging Challenges

The embedded Internet of Things poses many challenges to researchers, developers and users. Let's look at the selected challenges we consider to be most important.

**Embedded devices vs. interaction devices.** Embedding interaction in an Internet of Things context means integrating interaction opportunities into existing artifacts, devices, and environments. Unlike interaction devices, embedded interaction mostly utilizes objects people already use or are familiar with and broadens their impact and functions. However, it can be difficult to add functionality without radically changing the way an object originally behaves or looks. In addition, the new features are subject to the invisibility dilemma described next.

**Invisibility dilemma.** When embedding information and interfaces into objects, a vital design element is to hide this augmentation and leave the original function, look, and feel the same. However, this physical disappearance and embedded sensing, actuation, and interaction can affect the user's perception and lead to the invisibility dilemma. Users must still be able to identify digitally enhanced artifacts known and used in everyday life as more potent than meets the eye. In addition, the users must recognize this added value to accept and use such artifacts.

**Implicit vs. explicit interaction.** Explicit use means that a user operates a system knowingly to achieve a certain goal — that is, the user is fully aware of the tool he or she is using (as in the Case 2 example). Implicit use, in contrast, means the user concentrates on his or her prime goal or targeted activity; tool use is intended, but the user isn't actually aware of the interaction with the computer system. This interaction occurs implicitly, but on purpose (in contrast to the idea of incidental interaction[15]). Depending on the application and intended usage of the netget at hand, the decision for either implicit or explicit interaction must be made carefully.

**Context dependence.** The value of having access to information depends on context. Many different contexts (an overview is available elsewhere[16]) make a whole range of sensors and input processing systems necessary. For most context-aware applications, focusing on just a person, an object, or a specific environment is meaningful, but in the Internet of Things, these borders merge and vanish.

**Interaction and multimodality.** Embedded interaction with netgets is characterized by multimodal interaction. Specialized devices, as interaction gateways to the Internet of Things, gain particular importance based on the user's context. They can, if carefully designed, let users interact via modalities more suitable and adapted to the task, environment, and context. This ideally reduces cognitive load, supports interaction execution and goals, and uses interaction channels that leverage users' overall performance.

**Development support.** Toolkits, frameworks, and APIs let designers or developers more effectively prototype, test, evaluate, and develop embedded interaction applications. In our research, we identified four main prerequisites for such toolkits:

- support various hardware, software, and development paradigms;
- support the creation of simple and complex applications;
- support debugging and changing applications; and
- integrate (into) the whole development process.

**We developed our EIToolkit to support these requirements.** The toolkit is a component-based architecture in which each component is represented by a proxy-like object called a *stub*. These stubs translate messages between a general communication area to devices' specific protocols and back. Any component can then register to listen to messages directly addressed to it or broadcast to all. This enables component exchange at runtime. The system also allows the developer to change the message protocol on a per-component basis. EIToolkit supports various transmission protocols and formats and several microcontroller platforms. Although a more thorough description is out of scope and available elsewhere,[17] we want to stress that a toolkit like this can tremendously ease development, especially for people without proliferate programming expertise. Thus, sample stubs are available — for example, to control the media player Winamp, for MIDI output, or for keyboard emulation (see www.eitoolkit.de for more information). This lets developers quickly use and connect various smart artifacts within the Internet of Things.

## Lessons Learned and Implications

We drew several conclusions from our experiences within the field of embedded interactions and came up with a list of design guidelines:

- *Information when and where it's useful (case 4).* Provide information to increase users' ability to make informed choices. Usually, information is embedded at points where decisions are made or where the user has choices.
- *Information provision without explicit interaction (cases 1, 4).* Account for providing relevant information without requiring the user to explicitly trigger it. You can, for example, exploit a phone's screensaver for this purpose.[18]
- *Overprovisioning (cases 2, 4).* Enable many methods, input and output devices, and locations to achieve a task. The Internet of Things provides many opportunities and removes user limits and constraints.
- *Specialized components (cases 1, 3, 4).* For specific tasks or target users, consider using a specific device or interaction technique. This can be much more efficient or easier to use than a generic, all-in-one device such as a powerful smart phone.
- *Visibility (cases 1, 2, 3).* Carefully consider the trade-off between clearly visible controls and those seamlessly integrated into (a possibly existing) product design.
- *Accidental use (cases 2, 4).* Prevent accidental use, otherwise people might be afraid of initiating actions involuntarily and hence might refrain from using a device at all.
- *The invisibility dilemma (cases 1, 2, 3).* Seamlessly transfer objects into the Internet of Things — an object's original behavior shouldn't change, but the user must still perceive added value.
- *Short- and long-term life cycle (all cases).* Take care that devices run for a satisfactory amount of time before they need recharging or replacement. One way to tackle this is to provide easy ways to access technological components, even if it is only to replace the battery.
- *Rapid prototyping (cases 2, 3, 4).* To (cost-) effectively explore the design space, employ prototyping tools to create early prototypes. Hardware and software components such as the EIToolkit are available that accelerate presentation and evaluation for developers' application ideas.
- *Modeling support (all cases).* Provide support for formal models, which can save tremendous amounts of time and money, for example, by providing some usage estimates. We can, for example, extend methods such as the Keystroke Level Model even for novel types of interaction[19] and predict metrics such as task-completion times.

This list of guidelines can of course be continued with more general ones that apply to a wider set of projects. However, for the subject of embedded interaction with the Internet of Things at hand, we think it already covers a major part of potential issues extracted from our experiences. We see a considerable potential for others to avoid potential problems right at the initial phases of a project in this area.

Embedding interaction opportunities into everyday objects such as cups or mechanical tools lets us seamlessly communicate and interact with the Internet of Things, creating a link between physical and digital worlds to an extent previously unknown. We unobtrusively and implicitly achieve our goals, complete our tasks, and thereby enable and use services our environment provides.

The challenges this article introduces can help us ask the right questions at a project's start, and the list of guidelines can point us toward solutions within the desired design space. Following those, applications focused on human-machine interaction are carried beyond ubiquitous computing to everyday computing with and within the Internet of Things.

## Acknowledgments

This work has been funded in parts by the German DFG funded Cluster of Excellence "CoTeSys — Cognition for Technical Systems."

## References

1. M. Kranz et al., "Context-Aware Kitchen Utilities," *Proc. 1st Int'l Conf. Tangible and Embedded Interaction* (TEI 07), ACM Press, 2007, pp. 213–214.
2. M. Kranz et al., "Sensing Technologies and the Player Middleware for Context-Awareness in Kitchen Environments," *Proc. 4th Int'l Conf. Networked Sensing Systems* (INSS 07), IEEE Press, 2007, pp. 179–186.
3. R. Wimmer et al., "CapTable and CapShelf — Unobtrusive Activity Recognition Using Networked Capacitive Sensors, *Proc. 4th Int'l Conf. Networked Sensing Systems* (INSS 07), IEEE Press, 2007, pp. 85–88.
4. S. Paasovaara et al., "Evaluating Capacitive Touch Input on Clothes," *Proc. 10th Int'l Conf. Human-Com-*

*puter Interaction with Mobile Devices and Services* (MobileHCI 08), ACM Press, 2008, pp. 81–90.
5. P. Holleis et al., "Playing with the Real World," *J. Virtual Reality and Broadcasting*, vol. 3, no. 1, 2006, pp. 1–12.
6. M. Kranz et al., "The Therapy Top Measurement and Visualization System – An Example for the Advancements in Existing Sports Equipments," *Int'l J. Computer Science in Sport*, vol. 5, no. 2, 2006, pp. 201–210.
7. A. Schmidt, P. Holleis, and M. Kranz, "Sensor Virrig: A Balance Cushion as Controller," *Adjunct Proc. 6th Int'l Conf. Ubiquitous Computing* (UbiComp 04), 2004; www.hcilab.org/documents/sensorvirrigabalancecushionascontroller.pdf.
8. T. Matthews et al., "Augmenting Collections of Everyday Objects: A Case Study of Clothes Hangers as an Information Display," *Proc. 2nd Int'l Conf. Pervasive Computing* (Pervasive 04), Springer, 2004, pp. 340–344.
9. P. Holleis et al., "Displayed Connectivity," *Adjunct Proc. 7th Int'l Conf. Ubiquitous Computing* (UbiComp 05), 2005; www.hcilab.org/documents/DisplayedConnectivity-HolleisKranzSchmidt_UbiComp2005Demo.pdf.
10. P. Holleis, "Programming Interactive Physical Prototypes," *Proc. 1st Int'l Workshop on Design and Integration Principles for Smart Objects* (DIPSO 07), 2007; http://eis.comp.lancs.ac.uk/workshops/dipso/dipso2007/programme.php.
11. 11. M. Kranz, P. Holleis and A. Schmidt, "Ubiquitous Presence Systems", *Proc 2006 ACM symposium on Applied Computing*, ACM Press, 2006, pp. 1902–1909.
12. T. Collet, B. MacDonald, and B. Gerkey, "Player 2.0: Toward a Practical Robot Programming Framework," *Proc. Australasian Conf. Robotics and Automation* (ACRA 05), IEEE Press, 2005, pp. 1–9.
13. E. Aitenbichler, J. Kangasharju, and M. Mühlhäuser, "MundoCore: A Light-Weight Infrastructure for Pervasive Computing," *Pervasive and Mobile Computing*, Elsevier Science Publishers, vol. 3, no. 4, 2007, pp. 332–361.
14. S. Klemmer et al., "Toolkit Support for Vision-Based Tangible Interfaces," *Proc. 22th Conf. Human Factors in Computing Systems* (CHI 04), ACM Press, 2004, pp. 399–406.
15. A. Dix, "Beyond Intention – Pushing Boundaries with Incidental Interaction," *Proc. Building Bridges: Interdisciplinary Context-Sensitive Computing*, Glasgow Univ., 2002, pp. 1–6.
16. A. Schmidt, M. Beigl, and H.-W. Gellersen, "There Is More to Context than Location," *Computers and Graphics*, vol. 23, no. 6, 1999, pp. 893–901.
17. P. Holleis and A. Schmidt, "MakeIt: Integrate User Interaction Times in the Design Process of Mobile Applications," *Proc. 6th Int'l Conf. Pervasive Computing* (Pervasive 08), Springer, 2008, pp. 56–74.
18. A. Schmidt et al., "Utilizing Mobile Phones as Ambient Information Displays," *Extended Abstracts of the 24th Conf. Human Factors in Computing Systems* (CHI 06), ACM Press, 2006, pp. 1295–1300.
19. P. Holleis et al., "Keystroke-Level Model for Advanced Mobile Phone Interaction," *Proc. 25th Conf. Human Factors in Computing Systems* (CHI 07), ACM Press, 2007, pp. 1505–1514.

---

**Matthias Kranz** is a professor of distributed multimodal information processing at Technische Universität München, Germany. His research interests are in ubiquitous computing, human-computer interaction, cognitive systems, and especially multimodal user interfaces. Kranz has a PhD in computer science from Ludwig-Maximilians-Universität München. Contact him at matthias.kranz@tum.de.

**Paul Holleis** is a researcher in the Smart and Secure Services research group at DOCOMO Euro Labs in Munich. His main research interests focus on gathering and using context and other information to provide matching mobile services to users, modeling mobile interactions, and developing and evaluating novel types of mobile and embedded interfaces. Holleis has a PhD in computer science from the Ludwig-Maximilians University in Munich. Contact him at paul@hcilab.org.

**Albrecht Schmidt** is a professor of pervasive computing and user interface engineering at the University of Duisburg-Essen, Germany. His teaching and research interests are in media informatics and ubiquitous computing (in particular, the area of user interface engineering). Schmidt has a PhD in computer science from Lancaster University. Contact him at albrecht.schmidt@acm.org.

---