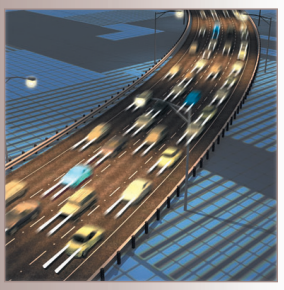

Editors: Frédéric Thiesse • *frederic.thiesse@unisg.ch*
Florian Michahelles • *fmichahelles@ethz.ch*

Smart Objects as Building Blocks for the Internet of Things

The combination of the Internet and emerging technologies such as near-field communications, real-time localization, and embedded sensors lets us transform everyday objects into smart objects that can understand and react to their environment. Such objects are building blocks for the Internet of Things and enable novel computing applications. As a step toward design and architectural principles for smart objects, the authors introduce a hierarchy of architectures with increasing levels of real-world awareness and interactivity. In particular, they describe activity-, policy-, and process-aware smart objects and demonstrate how the respective architectural abstractions support increasingly complex application.

Gerd Kortuem and Fahim Kawsar
Lancaster University

Daniel Fitton
University of Central Lancashire

Vasughi Sundramoorthy
University of Salford

The term *Internet of Things*[1] has recently become popular to emphasize the vision of a global infrastructure of networked physical objects. Although this vision is compelling, no consensus exists about how to realize it. The Internet of Things is partly inspired by the success of RFID technology, which is now widely used for tracking objects, people, and animals. RFID system architecture is marked by a sharp dichotomy of simple RFID tags and an extensive infrastructure of networked RFID readers. This approach optimally supports tracking physical objects within well-defined confines (such as warehouses) but limits the sensing capabilities and deployment flexibility that more challenging application scenarios require.

We're working toward an alternative architectural model for the Internet of Things[1] as a loosely coupled, decentralized system of *smart objects* – that is, autonomous physical/digital objects augmented with sensing, processing, and network capabilities. In contrast to RFID tags, smart objects carry chunks of application logic that let them make sense of their local situation and interact with human users. They sense, log, and interpret what's occurring within themselves and the world, act on their own, intercommunicate with each other, and exchange information with people.

 1089-7801/10/$26.00 © 2010 IEEE

The vision of an Internet of Things built from smart objects raises several important research questions in terms of system architecture, design and development, and human involvement. For example, what is the right balance for the distribution of functionality between smart objects and the supporting infrastructure? How do we model and represent smart objects' intelligence? What are appropriate programming models? And how can people make sense of and interact with smart physical objects?

A key insight of our work is that the answers to these questions are interrelated, so it doesn't make sense to attempt to answer each question in isolation. Through practical experimentation and by prototyping many generations of smart objects, we identified three canonical smart-object types (see Figure 1) that we believe represent fundamental design and architectural principles: *activity-aware objects*, *policy-aware objects*, and *process-aware objects*. These types represent specific combinations of three design dimensions that we'll discuss later. Here, we aim to highlight the interdependence between design decisions and explore how smart objects can cooperate to form an "Internet of smart objects."

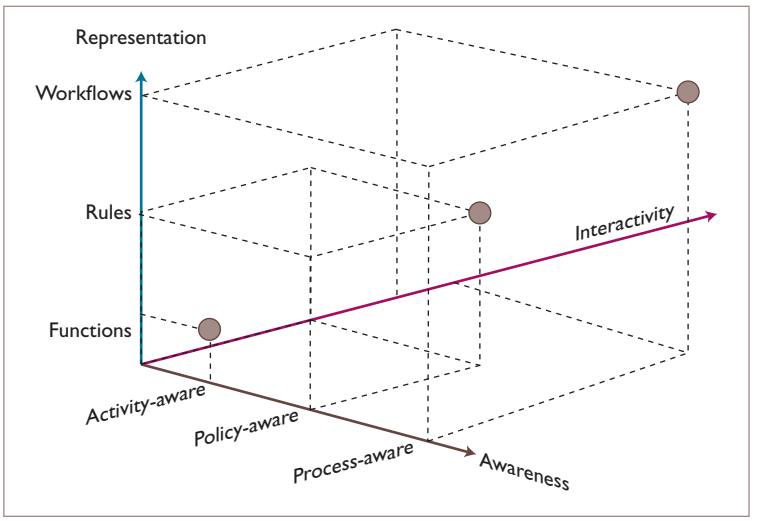

Figure 1. Smart-object dimensions. We can see the three canonical object types, activity-aware, policy-aware, and process-aware.

Smart Objects for Industrial Workplaces

Our exploration of smart objects and the Internet of Things is informed by the requirements of industrial application scenarios — in particular, in the petrochemical and road construction industries. Our first case study investigated chemical storage at a processing plant, in particular, the use and handling of chemical drums;[2] the second case study looked at "road patching," a typical maintenance task aimed at repairing defects in a road's surface (see Figure 2a).[3]

Although RFID technology is widely deployed in many industries, its use in temporary and highly dynamic work environments such as construction sites is severely restricted. To overcome the handicap of an extensive external infrastructure, we chose to convert existing work objects such as containers and tools (pavement breaker, drum roller, and wacker plate compactor) into smart objects by augmenting them with embedded sensor devices (based on an ARM7 processor) and wireless capabilities (following the 802.15.4 near-field radio standard). The resulting smart work objects can autonomously interpret sensor data and make decisions but can also communicate and cooperate with each other. To enable user input and output, we equipped smart objects with a small, embedded display and a set of buttons. In addition, we developed a wireless wearable device that functions as a remote interface device for smart objects (Figure 2b).

Smart-Object Typology

Through a multiyear collaboration with industrial partners, we were able to build various design alternatives for smart objects and explore the smart-object design space in depth. Although we deployed several hardware platforms to accommodate increasing computational requirements and emerging standards, we essentially kept the same hardware design throughout. The key differences in our designs can be found along the following three design dimensions:

- *Awareness* is a smart object's ability to understand (that is, sense, interpret, and react to) events and human activities occurring in the physical world.
- *Representation* refers to a smart object's application and programming model — in particular, programming abstractions.
- *Interaction* denotes the object's ability to converse with the user in terms of input, output, control, and feedback.

Through iterative exploration and testing of various designs, we discovered that the most useful designs weren't evenly spread through-

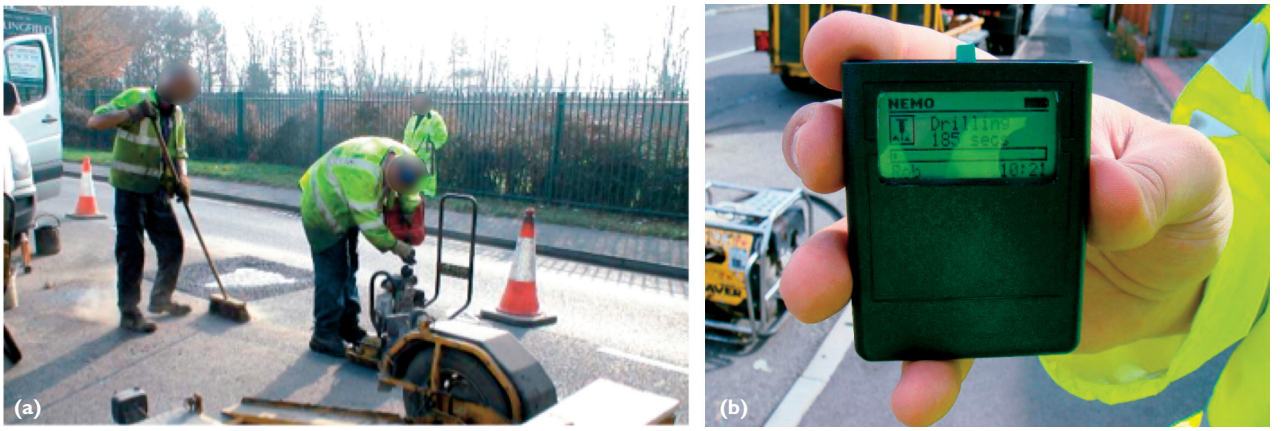

Figure 2. Road-patching case study. This study used (a) a smart object deployed at a road construction site. Workers used (b) wearable user interface devices that showed personal health records containing information about a worker's exposure to hazardous equipment vibration.

Table 1. Summary of smart-object types.

	Awareness	Representation	Interaction	Augmentation	Example application
Activity-aware object	Activities and usage	Aggregation function	None	Time, state (on/off), vibration	Pay-per-use
Policy-aware object	Domain-specific policies	Rules	Accumulated historical data, threshold warnings	Time, vibration, state, proximity	Health and safety
Process-aware object	Work processes (that is, sequence and timing of activities and events)	Context-driven workflow model	Context-aware task guidance and alerts	Time, location, proximity, vibration, state	Active work guidance

out the design space but clustered around the three main object types we introduced previously (see Figure 1). Table 1 summarizes these object types and how they relate to the three design dimensions just introduced.

Activity-Aware Smart Objects

An activity-aware object can record information about work activities and its own use. In particular, we can characterize it as follows:

- *Awareness*. An activity-aware object understands the world in terms of event and activity streams, where each event or activity is directly related to the use and handling of the object (pick up, turn on, operate, and so on).
- *Representation*. Its application model consists of aggregation functions for accumulating activities over time.
- *Interaction*. Activity-aware objects primarily log data and don't provide interactive capabilities.

Activity-aware objects are the simplest of the three types, and they already support interesting smart-object applications. For the construction case study, for example, we developed a pay-per-use tool that uses sensors to record data about the timing and duration of its use and how workers handle it.[4] The tool converts this usage data into a financial cost figure, which equipment rental companies can use to realize a pay-per-use business model. The tool also detects worker misuse (for example, dropping the tool to the ground or overheating it) and automatically takes into account necessary maintenance and repair costs. (Most equipment in the construction industry is rented on a contractual basis, but rent prices depend only on contract length.) Pay-per-use tools benefit construction companies as well because they support real-time cost capturing in the field.

Technically, an activity-aware smart object analyzes the data stream from its sensors, uses recognition algorithms to detect activi-

ties and events, and applies application-specific aggregation functions. Further discussion of usage-based pricing policies for smart products appears elsewhere.[5]

Policy-Aware Smart Objects

A policy-aware object is an activity-aware object that can interpret events and activities with respect to predefined organizational policies. We can describe it within our design parameters as follows:

- *Awareness*. A policy-aware object understands to what extent real-world activities and events comply with organizational policies.
- *Representation*. Its application model consists of a set of rules that operate on event and activity streams to create actions.
- *Interaction*. A policy-aware object provides context-sensitive information about object handling and work-activity performance. In particular, it can issue warnings and alerts if workers violate policies.

We've used policy-aware object design to develop health- and safety-aware smart objects for chemical storage and road construction scenarios. In the first case, we developed a smart barrel with embedded storage rules for various chemicals.[2] Depending on temperature, vibrations, and barrels' relative proximity, it informs workers about safety violations and prompts them to take appropriate action. In our construction case study, we developed a family of vibration-aware tools that can monitor workers' exposure to dangerous vibrations.[3] These smart tools aim to minimize the occurrence of *vibration white finger* (VWF), a painful and potentially debilitating disease caused by long-term accumulative exposure to vibrations. The smart tools carry an explicit model of legal health and safety regulations, which state maximum daily and average exposure levels.[6] The tools record equipment use and send information to a worker's wearable tag, where it's stored as a personal health log. The tag visually indicates current exposure levels (Figure 3b) and, if vibrations exceed legal limits, alerts workers.

Technically, a policy-aware object is an activity-aware object with an added embedded policy model. The user interface is an important aspect of policy-aware objects; they not only

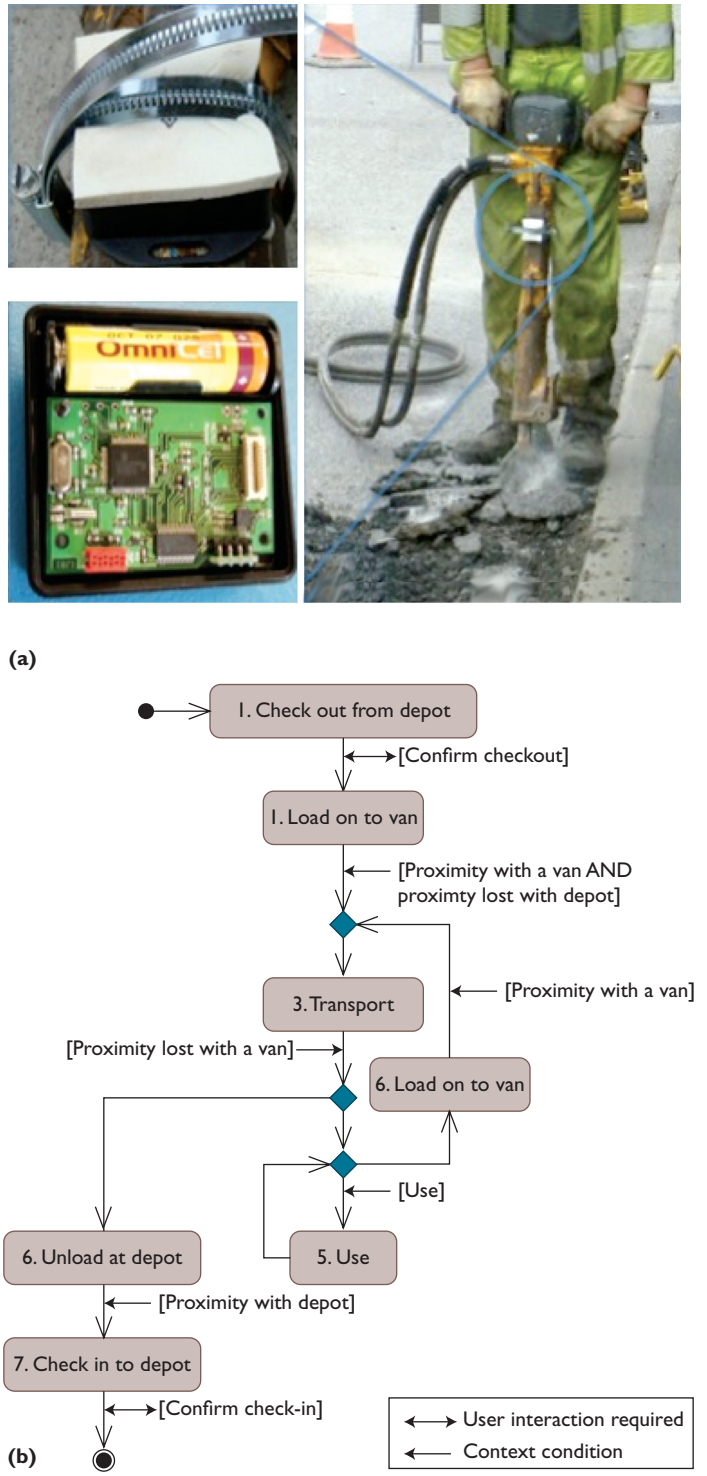

Figure 3. Smart objects in the field. We designed and field tested (a) a pneumatic pavement breaker prototype that gathers data about usage patterns and provides context-aware guidance during construction work. The top left image shows the provisional attachment of the sensor board to the pavement breaker; the lower left image shows the sensor board. To model the tool's organizational process, we use (b) a workflow that defines the work activities in which the smart object is involved.

Related Work in Smart Objects

Research on smart objects and the Internet of Things has been going on for more than a decade and reaches back to Mark Weiser's original vision of ubiquitous computing. Bruce Sterling recently popularized the idea of smart objects and the Internet of Things; Sterling coined the term *spime*[1] to describe a new category of *s*pace-*time* objects that are aware of their surroundings and can memorize real-world events. Julian Bleeker advocated a similar notion of *blogjects* (objects that blog) in his "Manifesto for Networked Objects."[2] This more visionary work has been met by a growing body of technology- and business-focused research on RFID, smart objects, and smart products.[3]

Roy Want and his colleagues augmented physical objects with passive RFID tags so that they were uniquely identifiable and information related to them could be presented to their users.[4] Michael Beigl and his colleagues defined a smart object as "an everyday artifact augmented with computing and communication, enabling it to establish and exchange information about itself with other artifacts and/or computer applications."[5] Friedemann Mattern formulated in a similar way: "Smart objects might be able not only to communicate with people and other smart objects, but also to discover where they are, which other objects are in the vicinity, and what has happened to them in the past."[6] Norbert Streitz and his colleagues looked at smart objects from two perspectives: one model has system-oriented, importunate smartness in which smart objects can take certain self-directed actions based on previously collected information; the other is people-oriented, empowering smartness where smart objects empower users to make decisions and take mature and responsible actions.[7]

Most recent work on smart objects has focused on technical aspects (hardware platforms, software infrastructure, and so on[8]) and application scenarios. Application areas range from supply-chain management and enterprise applications to (home and hospital) healthcare[9] and industrial workplace support.[10–13] Human-interface aspects of smart-object technology are just beginning to receive attention,[14] yet design principles and methods for smart objects that go beyond mere hardware have yet to be explored. Our work on exploring the smart-object design space and identifying canonical smart-object types is a first step in this direction (see also Fahim Kawsar's dissertation.[15]). In particular, we view as paramount to holistically investigate sensing, modeling, and user interface issues.

References

1. B. Sterling, *Shaping Things*, MIT Press, 2005.
2. J. Bleecker, "A Manifesto for Networked Objects — Cohabiting with Pigeons, Arphids, and Aibos in the Internet of Things," blog, 2006; http://research.techkwondo.com/blog/julian/185.
3. F. Thiesse and M. Kohler, "An Analysis of Usage-Based Pricing Policies for Smart Products," *Electric Markets*, vol. 18, no. 3, 2008, pp. 232–241.
4. R. Want et al., "Bridging Physical and Virtual Worlds with Electronic Tags," *Proc. ACM SIGCHI 99*, ACM Press, 1999, pp. 370–377.
5. M. Beigl, H.-W. Gellersen, and A. Schmidt, "MediaCups: Experience with Design and Use of Computer-Augmented Everyday Objects," *Computer Networks*, vol. 35, no. 4, 2001, pp. 401–409.
6. F. Mattern, "From Smart Devices to Smart Everyday Objects," *Proc. Smart Objects Conf.* (SOC 03), Springer, 2003, pp. 15–16.
7. N. Streitz et al., "Designing Smart Artifacts for Smart Environments," *Computer*, vol. 38, no. 3, 2005, pp. 41–49.
8. F. Siegemund, "A Context-Aware Communication Platform for Smart Objects," *Proc. 2nd Int'l Conf. Pervasive Computing* (Pervasive 04), Springer, 2004, pp. 69–86.
9. S. Konomi and G. Roussos, "Ubiquitous Computing in the Real World: Lessons Learnt from Large-Scale RFID Deployments," *Personal and Ubiquitous Computing*, vol. 11, no. 7, 2007, pp. 507–521.
10. M. Strohbach et al., "Cooperative Artefacts: Assessing Real World Situations with Embedded Technology," *Proc. Int'l Conf. Ubiquitous Computing* (Ubicomp 04), Springer, 2004, pp. 250–267.
11. G. Kortuem et al., "Sensor Networks or Smart Artifacts? An Exploration of Organizational Issues of an Industrial Health and Safety Monitoring System," *Proc. Int'l Conf. Ubiquitous Computing* (Ubicomp 07), Springer, 2007, pp. 465–482.
12. D. Fitton et al., "Exploring the Design of Pay-Per-Use Objects in the Construction Domain," *Proc. 3rd European Conf. Smart Sensing and Context*, Springer, 2008, pp. 192–205.
13. C. Efstratiou et al., "Experiences of Designing and Deploying Intelligent Sensor Nodes to Monitor Hand-Arm Vibrations in the Field," *Proc. 5th Int'l Conf. Mobile Systems, Applications, and Services* (MobiSys 07), ACM Press, 2007, pp. 127–138.
14. L. Nelson and E.F. Churchill, "User Experience of Physical-Digital Object Systems: Implications for Representation and Infrastructure," *Smart Object Systems Workshop*, in conjunction with *Int'l Conf. Ubiquitous Computing* (Ubicomp 05), 2005; http://elizabethchurchill.com/professional/pubs/Papers/2005UbicompNelsonChurchill.pdf.
15. F. Kawsar, *A Document-Based Framework for User Centric Smart Object Systems*, PhD dissertation, Dept. Computer Science, Waseda Univ., Feb. 2009.

record and interpret sensed data but also give users timely information. In this sense, policy-aware objects are interactive systems.

Process-Aware Smart Objects

Processes play a fundamental role in industrial work management and operation. A process is a collection of related activities or tasks that are ordered according to their position in time and space. A process-aware object represents the most accomplished of our three object types; we characterize it as follows:

- *Awareness*. A process-aware object under-

stands the organizational processes that it's part of and can relate the occurrence of real-world activities and events to these processes.

- *Representation*. Its application model consists of a context-aware workflow model[7] that defines timing and ordering of work activities.
- *Interaction*. A process-aware object provides workers with context-aware guidance about tasks, deadlines, and decisions.

We designed a process-aware tool for the construction industry that helps workers by providing just-in-time information about required work activities. To model the organizational process, we use a workflow-like notion that defines work activities in which the smart object is involved. Figure 3b provides an example workflow for a pneumatic pavement breaker (shown in Figure 3a). The workflow contains activities and transitions between activities. Transitions are annotated with context conditions that refer to sensor or human input. A workflow continues along a transition if input satisfies a condition.

The motivation for this smart object stems from construction work sites' complexity and the large number of available tools used for specific purposes and from different sources (the construction company rents most equipment from plant and machinery rental companies). Consequently, tools are part of a range of different processes at business, organizational, and physical work-activity levels. These processes cross boundaries between different organizations — for example, between the rental company and the construction company. This complexity makes it challenging for workers to ensure that they are following the correct process for each work object at each level.

The process-aware tool "understands" how workers are supposed to use it in each context and which work activities ought to be done next. It uses this understanding to provide context-sensitive guidance about tasks and processes. To give workers active guidance, we slightly enhanced the display device from the second-generation prototype to incorporate four buttons. These let workers view the current activity in which the object is involved (along with the time started) and navigate forward and backward in the flow to see the workflow's past execution and the activities they must carry out in the future.

The three design dimensions we developed for designing smart objects provide a structured approach. The right balance of representation, awareness, and interactivity depends on the application scenario's requirements; more complex and abstract designs aren't always better. In this sense, our smart-object types represent true design alternatives and not a necessary progression toward a final design.

Toward an Internet of Smart Objects

As our preceding examples demonstrate, individual smart objects working in isolation create interesting opportunities for novel information services. Yet, smart objects' true power arises when multiple objects cooperate to link their respective capabilities. Our early example of cooperating smart objects, the safety-aware chemical drum,[2] is a policy-aware smart object whose application model consists of a set of rules for determining to what extent workers handle it in accordance with safety rules. When we bring multiple smart drums together in close physical proximity, they act as a collective system: drums let each other access their respective rule sets and can thus make collective assessments about their safety status as a group (for example, whether the overall volume of all drums exceeds a dangerous limit). In this example, the drums achieve cooperation via a peer-to-peer (P2P) reasoning algorithm for collocated smart objects, in which the reasoning process physically "jumps" from one smart object to the next. All drums that have been part of the collective assessment display notices for users.

This example highlights two key research areas for smart objects and the Internet of Things.

Dynamic Ad Hoc Composition of Models

As we described, smart objects are autonomous objects that carry chunks of application logic. How can we combine these chunks into a coherent collective application model? How can we do this in an ad hoc manner whenever objects come together within the same physical environment? The P2P reasoning algorithms we mentioned (see Figure 4)[2] provide one example for policy-aware objects, but this approach doesn't address performance and security concerns. Further-

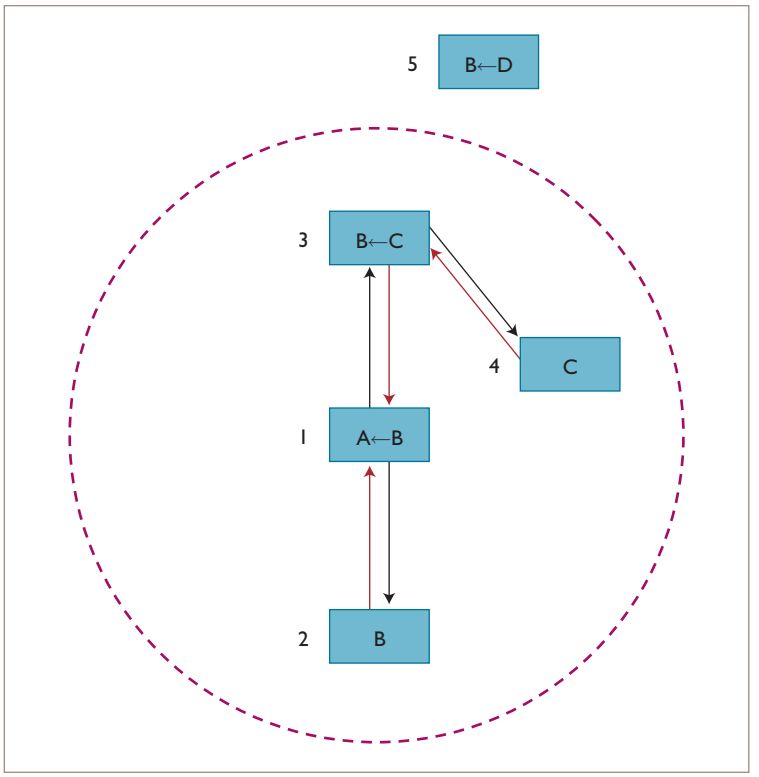

FIgure 4. Peer-to-peer reasoning with smart objects. Colocated smart objects with local rule databases (1–5) form an ad hoc reasoning system in which logical queries are sent from object to object and results are passed back along the inference chain. In this example, smart object 1 contains one rule, A if B; to evaluate B, it sends a query to objects 2 and 3, which in turn asks object 4. Reasoning chains can be limited to physical areas around the originating object. In this example, smart object 5 is beyond the distance limit set by smart object 1 (the circle indicates the maximum distance from object 1).

more, it's unclear how we might combine high-level application models — such as workflows for process-aware smart objects — ad hoc.

Dynamic Ad Hoc Composition of Interactive Capabilities

Smart objects are more than just sensor nodes; they're interactive tools designed to help people accomplish tasks in the real world. As such, smart objects' interactive input and output capabilities are key to their success. This was highlighted through an ethnographic workplace study that uncovered the impact of smart objects' interactive capabilities on people's understanding of and attitude toward smart-object technology.[2] The research community has yet to address the question of designing distributed user interfaces for smart-object collections.

Smart objects provide a distributed architectural model for the Internet of Things. Due to their dual nature as physical and digital entities, such objects highlight the fact that the Internet of Things can't be viewed only as a technical system but must also be considered as a human-centered interactive one. This implies that we must expand smart-object design beyond hardware and software to include interaction design as well as social aspects. Our work on smart-object types has discovered important design patterns for smart objects that will not only inform their future design but also identify important technical requirements for the emerging Internet of Things.

Our current work is focused on two areas. First, we're developing a new flow-based programming paradigm for smart objects and the Internet of Things. This work involves developing new workflow models suitable for embedded devices and orchestration techniques for the ad hoc combination of smart-object flows. Second, we're developing a means to increase smart objects' interactive capabilities, with the goal of supporting people in performing complex physical tasks involving smart objects.

Acknowledgments

This research has been funded by the UK Engineering and Physical Sciences Research Council (EPSRC) through grant EP/C014677/1 — NEMO, and by the European Commission through grant EC 213339 — ALLOW: Adaptable Pervasive Flow, whose support we are pleased to acknowledge.

References

1. C. Floerkemeier et al., *Proc. 1st Int'l Conf. Internet of Things* (IOT 08), Springer, 2008; www.springer.com/computer/hardware/book/978-3-540-78730-3.
2. M. Strohbach et al., "Cooperative Artefacts: Assessing Real World Situations with Embedded Technology," *Proc. Int'l Conf. Ubiquitous Computing* (Ubicomp 04), Springer, 2004, pp. 250–267.
3. C. Efstratiou et al., "Experiences of Designing and Deploying Intelligent Sensor Nodes to Monitor Hand-Arm Vibrations in the Field," *Proc. 5th Int'l Conf. Mobile Systems, Applications, and Services* (MobiSys 07), ACM Press, 2007, pp. 127–138.
4. D. Fitton et al., "Exploring the Design of Pay-Per-Use Objects in the Construction Domain," *Proc. 3rd European Conf. Smart Sensing and Context*, Springer, 2008, pp. 192–205.

5. F. Thiesse and M. Kohler, "An Analysis of Usage-Based Pricing Policies for Smart Products," *Electronic Markets*, vol. 18, no. 3, 2008, pp. 232–241.
6. "The Control of Vibration at Work: Regulations 2005," *The Stationery Office Limited*, statutory instrument, no. 1093, 2005.
7. K. Herrmann et al., "An Emerging Technology for Pervasive Adaptation," *Proc. 2nd IEEE Int'l Conf. Self-Adaptive and Self-Organizing Systems Workshops* (SASOW 08), IEEE CS Press, 2008, pp. 108–113.

Gerd Kortuem is a senior lecturer in mobile and ubiquitous computing at Lancaster University. His research interests include ubiquitous information systems, augmented environments, smart artifacts, and mobile user interfaces. Kortuem has a PhD in computer science from the University of Oregon. He's a member of the IEEE, the ACM, and the British Computer Society, and is a former Fulbright fellow. Contact him at kortuem@comp.lancs.ac.uk.

Daniel Fitton is a lecturer in the School of Computing, Engineering, and Physical Sciences at the University of Central Lancashire. His research interests are the exploration of new applications and associated human-computer interaction issues in ubiquitous computing and emerging technologies, particularly mobile and embedded systems. Fitton has a PhD in computing from Lancaster University. Contact him at dbfitton@uclan.ac.uk.

Fahim Kawsar is a researcher in Lancaster University's computing department. His research revolves around ubiquitous computing with a specific focus on smart-object systems, human-centric system infrastructures, and tangible interfaces. Kawsar has a PhD in computer science from the Distributed Computing Lab at Waseda University. He's a recipient of the 2006-08 Microsoft Research (Asia) fellowship and a member of the IEEE and the ACM. Contact him at fahim.kawsar@gmail.com; www.fahim-kawsar.net.

Vasughi Sundramoorthy is a research scientist at the University of Salford in the Digital Environment Home Energy Management System (EU) project. Her research interests include mobile and network computing and human-computer interaction. Sundramoorthy has a PhD in computer science from the University of Twente, the Netherlands. Contact her at v.sundramoorthy@salford.ac.uk.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.

2 Free Sample Issues!

A $26 value

The magazine of computational tools and methods for 21st century science.

http://cise.aip.org
www.computer.org/cise

Send an e-mail to **jbebee@aip.org** to receive the two most recent issues of CiSE.
(Please include your mailing address.)

Recent Peer-Reviewed Topics:

Cloud Computing
Computational Astrophysics
Computational Nanoscience
Computational Engineering
Geographical Information Systems
New Directions
Petascale Computing
Reproducible Research
Software Engineering

MEMBERS $47/year
for print & online

AIP
IEEE
IEEE computer society