Smart Objects as Building Blocks for the Internet of Things

Gerd Kortuem
and Fahim Kawsar
Lancaster University

Daniel Fitton
University of Central Lancashire

Vasughi Sundramoorthy
University of Salford

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.

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 alterna tive 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.

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 rep-resent 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."

Smart Objects for Industrial Workplaces

Our exploration of smart objects and the Inter-net 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; the second case study looked at "road patching," a typical maintenance task aimed at repairing defects in a road's surface (see Figure 2a).

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

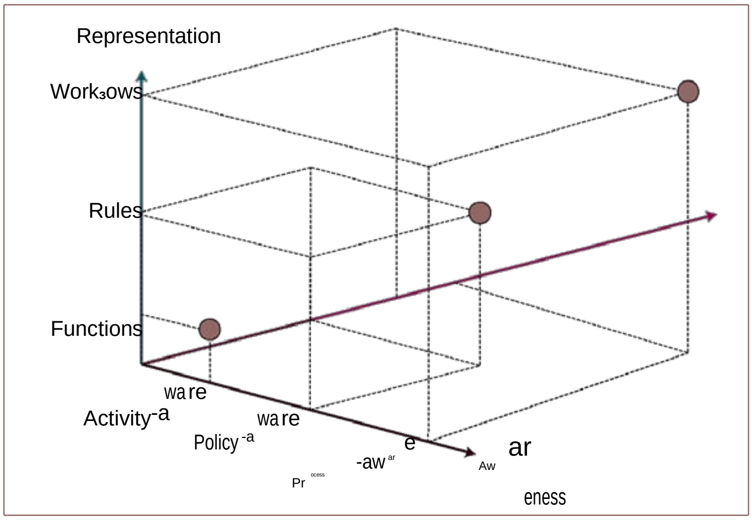

Figure 1. Smart-object dimensions. We can see the three canonical object types, activity-aware, policy-aware, and process-aware.

decisions, but also communicate and cooper-ate 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 plat-forms 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 occur-ring 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-

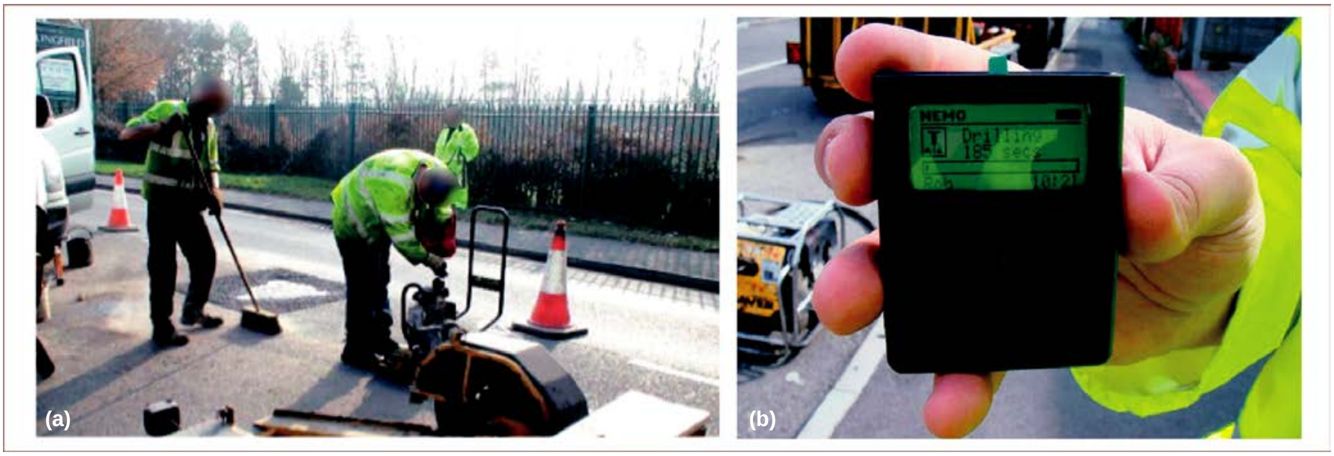

Figure 2. Road-patching case stud:. (a) a smart object deployed at a road construction site. Workers used (b) wearable user interface devices that showed personal health records containing information about a worker's exposure to hazardous equipment vibration.

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 inter-esting smart-object applications. For the con-struction 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 activities 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 activi-ties with respect to predefined organizational policies. We can describe it within our design parameters as follows:

- *Awareness*. A policy-aware object under-stands to what extent real-world activi-ties and events comply with organizational policies.
- *Representation*. Its application model con-sists 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 sce-narios. In the first case, we developed a smart barrel with embedded storage rules for various chemicals.[2] Depending on temperature, vibra-tions, and barrels' relative proximity, it informs workers about safety violations and prompts them to take appropriate action. In our con-struction 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 *vibra-tion white finger* (VWF), a painful and poten-tially 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 work-er'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

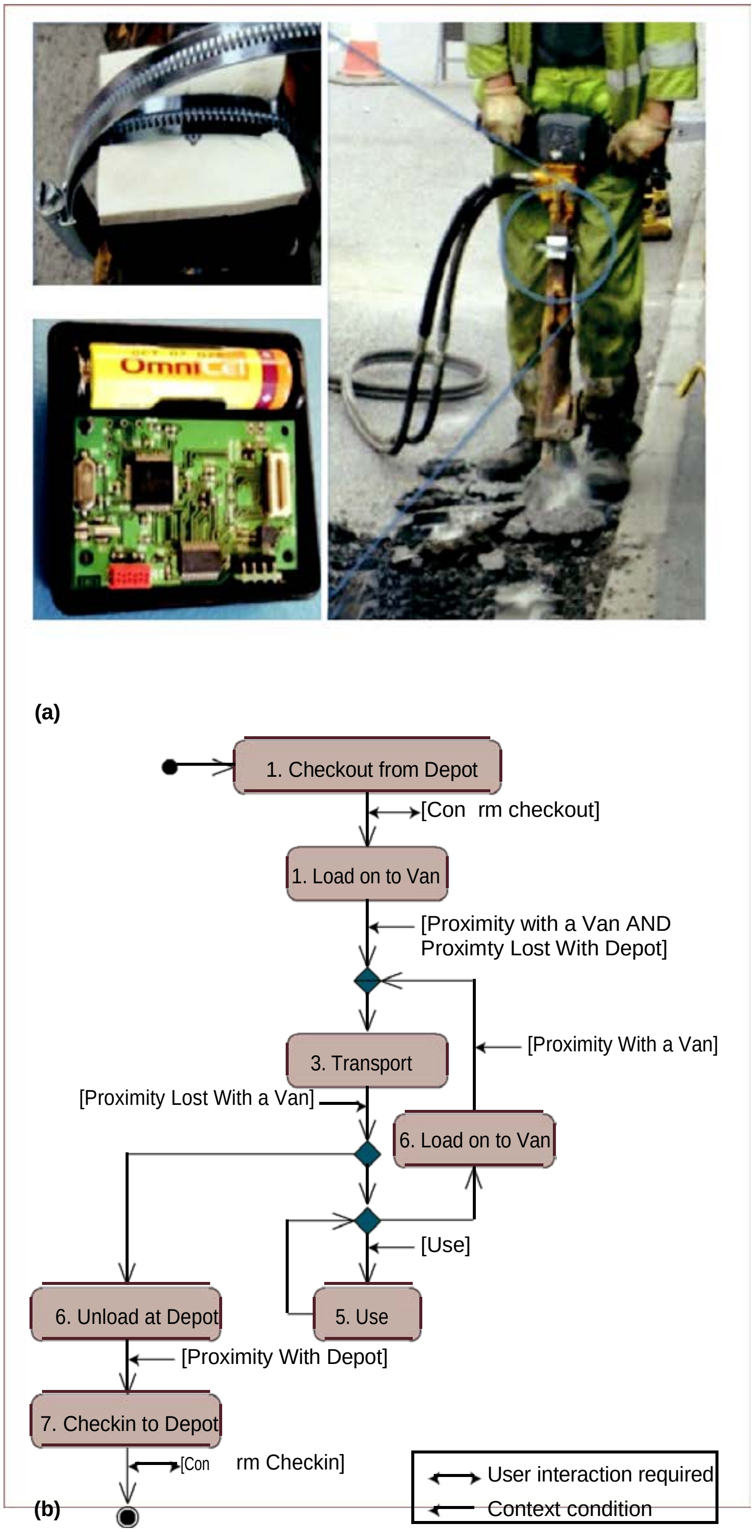

Figure 3. Smart objects in the field.