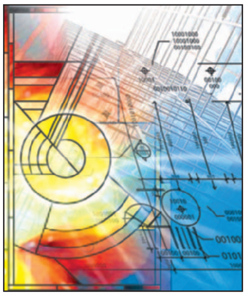

Smart Devices and Soft Controllers

Craig W. Thompson • *University of Arkansas*

We are in the early stages of a transition to a world in which everything is alive[1] — that is, a world where common objects (including those that are inanimate and abstract) can have individual identities, memory, processing capabilities, and the ability to communicate and sense, monitor, and control their own behaviors. As the Semantic Web gets smarter and reaches out into the world, the "Internet of Things" will encompass everyday things, connecting virtually all devices in our houses, cars, offices, and so on.

People are already overwhelmed with the complexity of devices that they own — there are so many complicated interfaces to negotiate and so far, there is not much relief in sight. If we have trouble managing and maintaining the small number of computers, laptops, and PDAs in our lives today, how can we expect to manage the coming explosion of complexity when faced with hundreds of networked devices?

What if things (initially devices, but perhaps expanding to include much more pets, places, and even abstractions) were smarter, easier to use, and could diagnose their own problems, present different interfaces to different people, and communicate with each other — all seamlessly, without our having to learn much to install, use, or repair them? Could it be that we are wasting a lot of time accommodating our devices while they should be accommodating us?

Why Do Complex Devices Make Us Feel Dumb?

Like many people, I have a drawer at home filled with 30 to 50 instruction manuals for appliances — thermostat, microwave, TVs, DVD-VCRs, stereo, washing machine, lawn mower, and so on. I also have manuals for software, hardware, wireless devices, scanners, and digital cameras. Although I've learned how to install and operate many of these devices, I often run into problems — even after hours of puttering, for example, I sometimes find it's not possible to train my universal remote to operate my new DVD-VCR. That means I have no choice but to keep another remote lying around, and if I break or lose one, I'm stuck until I get a replacement and relearn how to configure it.

With the change of seasons, again comes the half hour spent reprogramming my so-called "smart" thermostat, which can control temperature ranges in several areas of the house for several times of day for any day of the week. Yet, how many times have I attempted to use the Hold feature to temporarily override the standard temperature only to find a week later that I have been inadvertently heating the seldom-used attic office? My similarly complex watering controller comes on, even on rainy days, and needs to be reprogrammed before winter. Hmmm, where are those instructions, anyway? Then there are mystery areas — controls such as the oven's preset timer, which I know exists but have never successfully used. With effort, I can handle maintenance on my PCs (several email accounts, spam and ad blockers, virus updates, firewalls, backups, and so on). I spend at least several hours a week shopping for new devices, installing, reprogramming, maintaining, and debugging them — or getting experts to repair them.

Smart Devices

Sophisticated programmers find it useful to assume that objects can be implemented to present multiple interfaces that represent different views. Consider the common light switch: from its beginning, it's had multiple interfaces — a functional control interface for end users to turn the light on or off and a physical switch for controlling the connection to the electrical grid. Starting in the 1980s with the smart-home movement and X10-based home automation (www.x10.com), electrical

 Published by the IEEE Computer Society 1089-7801/05/$20.00

devices began to support a third interface — a smarter remote control interface, piggybacked on existing home wiring — but the movement toward smarter devices (or, more generally, smart objects) didn't quite take off.

About the same time, computers moved into our homes and became part of our lives, soon connecting us via the Internet and Web to the online world of shopping, email, and information sharing. In the same time frame, pervasive cell phone proliferation began increasing human connectivity. With many home Internet surfers now using broadband, and with increasingly pervasive wireless connectivity (including WiMax right around the corner; www.wimaxforum.org), it might be time to consider the next generation of Internet and Web-enabled smart devices. The Internet of Things will add new interfaces for smart devices so that they can have memory, sensors, actuators, computation, and communication.

What functionalities can we expect a smart light to support? It should be able to:

- Respond to commands, such as "turn on" or "turn off," or queries such as "Are you on or off?"
- Keep a log of when it was turned on and off, for what reason, and by whom. Queries or summaries could provide aggregate views of this information, or we could use this log to monitor usage.
- Keep a history of its manufacture and ownership.
- Provide self-diagnostics and instructions for use.
- Take orders from other objects; for example, a motion sensor could tell a light when to turn on. This means objects should be able to communicate with each other. Some commands could affect a single device; others could affect collections of devices — for instance, all the lights in the house.
- Remember information about other objects in a room or relay information to objects in other rooms. If you were to tell a light that you planned to take a bath in 15 minutes, for example, it could tell the tub to start filling 10 minutes later.
- Remember an activation sequence and repeat it. Alternatively, a schedule might control when a light turned on or off – different for vacations than day-to-day.

The light's brain and memory could be local to the light or remote. The information might be decentralized in storage so that the usage data is stored by the home lighting controller, in your laptop, or at the electric company — or all three.

Could it be we are wasting a lot of time accommodating our devices while they should be accommodating us?

Benefits

It might be interesting for devices to log their every state change, but how would humans benefit? Smart devices promise advantages, including:

- *Convenience and capability amplification.* Humans can control more of their world if smart objects help them. If we design them right, we will be able to control more smart devices in less time, which will lead to a sea change in productivity.
- *Remote control.* Devices can be controlled at a distance. Sometimes this provides time savings and convenience (turning on your house's sprinklers from the office, for instance); sometimes safety or security (using a robot to defuse a bomb).

Like ordinary devices, smart objects will have settings, but their controls can be remote. *Soft controllers* are remote interfaces that control one or more smart objects. Using a model-view-controller design pattern, they can present different look-and-feel skins and expose different capabilities to different users. Soft controllers for smart objects could make it

- easier to add a new object to a system, so that its soft controller extends the user interface dashboard in a modular, compositional way;
- possible to provide different users with different levels of control (letting one family member see a simple thermostat, for example, while another gets an interface that can program the heater and air conditioner schedule along with rules for interacting with other smart objects);
- easier to provide multimodal controls, including GUIs, speech, or controls for users with disabilities, for all sorts of smart objects; and
- possible for devices to communicate with each other.

If we're not careful, the first generation of smart objects will have significant defects. Perhaps if we isolate the requirements for smart objects, we can predict potential problems and find ways to reduce or eliminate them.

Smarter Device Requirements

Individual smart objects need some or all of the following capabilities:

- *Communications.* We will want to send and receive messages from smart objects, and they must be

able to accept queries and commands from each other, using wired or wireless connectivity.
- *Identity and kind.* Each smart object will need its own identity. Because different kinds of objects perform different functions, they will need different custom interfaces. Interfaces should be reflective, that is, self-describing — other objects should be able to ask them what they can do.
- *Memory and status tracking.* Smart objects must have persistent memory to ensure that they can maintain their settings and histories, but we should also be able to erase these memories when we discard the objects.
- *Sensing and actuating.* Smart objects will be able to monitor their environments and, if needed, change them.
- *Reasoning and learning.* Smart objects might or might not include various learning plug-ins. A simple form of learning might be to save settings; a little smarter than that might be to play back a recent log — for example, turn on lights today as they were activated yesterday. The first generation might not be very smart, but reasoning-capability plug-ins will improve.

Not all smart objects will have all capabilities, but it would be useful to be able to upgrade them over time to include additional capabilities. Today, we accomplish this by throwing out old objects and replacing them with new ones. In contrast, smart objects should be able to evolve, at least in capabilities and intelligence. If we develop smarter controls during a device's several-year life span, we should be able to download and install them at our convenience. Moreover, smarter objects should actively participate in their own just-in-time maintenance and track their own repair histories — or some other smart object might do it for them.

Smart objects are smart, in part, because they can interoperate with each other. The following is only a partial list of end-to-end capabilities that smart-object collections might have.

Although we have much of the technology we will need for smart objects to storm the world, we don't yet have a smart-object architectural roadmap.

- *Controllability.* If we just count larger physical objects we own and control, they might number in the hundreds, but if we count sensors, motes, and objects we pass near, we could interact with thousands or millions.
- *Maintainability.* We won't want to continually boot or upgrade individual smart objects, but we might need to turn them on or off or reset their controls at certain points.
- *Scalability.* Numerous smart objects must be able to dynamically join and leave various enclaves, leaving the store behind to join your household, recording a change of ownership, and getting to know a new collection of other smart objects.
- *Interoperability.* If several manufacturers develop competing variations on how to configure and connect smart objects, we could well experience smart object wars regarding interoperability standards.
- *Security.* We don't want just anyone to be able to turn our ovens on, but we might want our employees, teammates, or guests to be able to control the temperature in rooms they occupy. We can also expect analogs to viruses and spyware, which means we'll need to be able to distinguish between friendly, neutral, and enemy objects or capability plug-ins.
- *Privacy.* People will need ways to control how much information smart objects collect. Can we turn off or on various forms of collection? Can information, once recorded, be deleted or modified? Who has the digital rights to access this data? Does it become private when a person purchases a smart device? If the object monitors someone other than the purchaser — such as a family member or employee — do they have rights to control it? Who manages this data and ensures that it's backed up? How can we specify rules for which objects control others?
- *Reliability.* Will smart objects work and fail in ways we understand, helping us to pinpoint these failures?
- *Survivability.* If we add a new smart object, or if one stops working correctly, will that cascade and harm others? If our lives depend on smart objects (which exist to enhance our quality of life), will we be vulnerable to new forms of attack — such as a virus that causes intermittent braking problems on cars throughout the country?

The world won't likely be one vast sea of smart objects. Instead, enclaves might partition the smart-object world according to physical location, ownership, or trust. We might use policy management to state rules that govern smart objects' behavior within an enclave, although these rules might change at enclave boundaries.

Designing Smart Devices and Soft Controllers

Are we prepared for smart objects' inevitable entry into our lives? Happily, we already have much of the technology we will need to transition to a world in which smart objects are everywhere. In fact, some kinds of smart objects already exist, including PCs, laptops, PDAs, memory sticks, external disk drives, cameras, TVs, and cars. X10 and IR controllers are a half step further to increasingly automated environments. Some of these products and technologies expose APIs and can be controlled remotely. RFID tags are increasingly widely used to identify pallets, cartons, library books, and passports, among other things, and the ability to tag real-world objects is just starting to be pervasive. The Internet of Things appears to be an extension of the current Internet, which connects not only tethered machines but wireless entities such as laptops and PDAs as well. If it is, we can expect to use existing or evolving technologies to help control complexity. For instance, we've learned to avoid or recover from computer viruses — surely we'll handle smart object viruses in a similar manner.

At the computing infrastructure level, we have an embarrassment of riches. Java's "write once, run anywhere" technology, device plug-in platforms like Sun's Jini (http://wwws.sun.com/software/jini/), and increasingly general plug-in platforms (such as www.eclipse.org) provide us loosely coupled architectures for dynamically building systems from components. Web services provide XML languages such as the Web Service Description Language (WSDL) and SOAP to make distributed remote procedure calls easy while UDDI provides a registry for storing service advertisements. The grid community builds on these services to define computational and data grids for sharing computation and data across federated machine collections. Agent technology, which seems to be the natural support structure for smart objects, provides agent communication languages and ontology standards but only recently has explored agent capability composition. As described elsewhere,[2] the middleware community underwrites many of these technologies, providing the design patterns and distributed glue technologies that connect the parts. The agent community is now joining forces with the grid community to explore smart grids. The Semantic Web, too, adds metadata and ontology information to Web pages to make using the Web easier for both humans and programs. We must believe that this will be equally important for smart objects.

Despite all these technologies, we are missing some key ingredients. We still really don't understand how to prove a system's end-to-end properties. Aspect-oriented computing is in its infancy, compositional software is still elusive, and there's not yet a universal language for policy management. Digital licensing remains largely unexplored. We are far from being able to use natural language commands to control collections of devices. Hopefully, we will be able to separate most of these problems from each other and add them modularly into a smart object architecture.

Although we have much of the technology we will need for smart objects to storm the world, we don't yet have a smart object architectural roadmap. Many of the enabling technologies are being developed by research communities that haven't always communicated with each other historically. But convergence is in the air — and smart objects and controllers will require it. As the Web reaches out into the Internet of Things, we will increasingly add memory, processing, communications, and intelligence capabilities to the things around us. Targeting a world full of smart objects might provide us a clear grand challenge problem that can lead us to the world where everything truly is alive.

References

1. C. Thompson, "Everything is Alive," *IEEE Internet Computing*, Jan./Feb. 2004, pp. 83–86.
2. I. Foster, N.R. Jennings, and C. Kesselman, "Brain Meets Brawn: Why Grid and Agents Need Each Other," *Proc. 3rd Int'l Conf. Autonomous Agents and Multi-Agent Systems*, 2004, pp. 8–15.

Craig W. Thompson is professor and Acxiom Database Chair in Engineering at the University of Arkansas and president of Object Services and Consulting. His research interests include data engineering, software architectures, middleware, and agent technology. He received his PhD in computer science from the University of Texas at Austin. He is a senior member of the IEEE. Contact him at cwt@uark.edu.