International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 3 (2011) pp. 905-911

Towards A Semantic World: Smart Objects In A Virtual World

Akihiro Eguchi and Craig Thompson

Department of Computer Science and Computer Engineering
University of Arkansas, Fayetteville, AR, USA 72701
{aeguchi, cwt}@uark.edu

***Abstract*: The coming Internet of Things will usher in a semantic world (analogous to the semantic web) where physical objects will be networked so that they can communicate with each other and with humans. This paper identifies protocols that smart objects will need to follow and how to use today's 3D virtual worlds to better simulate and better understand protocols for tomorrow's smart world.**

Keywords: pervasive computing, semantic Web, Internet of Things, virtual world, smart objects, semantic world

I. Introduction

Pervasive computing is a megatrend – computing has migrated from mainframes to desktops, laptops to cell phones, and embedded computing is increasingly integrated into objects like cars and washing machines. Similar to the way we develop a semantic web [1] by adding metadata to web objects, we can see a coming Internet of Things where every individual physical object has a unique identity provided by technologies like RFID. We talk about smart worlds full of smart objects [2]. But what makes a smart object smart?

The objective of our project (http://vw.ddns.uark.edu) is to gain an understanding of and learn how to design "smart objects". Our long term aim is to help to create a collection of interoperability standards that provide a migration path to convert a world of ordinary objects into a semantic world containing smart objects, incrementally, one smart object and one protocol at a time.

Many technologies are contributing toward a smart, semantic world. The term "Internet of Things" seems to have had its origin with the original Auto-ID Center founded at MIT in 1999 which later became EPCglobal, focuses on a suite of radio frequency identification (RFID) technologies. Sensor networks generalize this concept. The pervasive computing, embedded computing, and smart home communities have contributed. Recent papers have begun to generalize this work to explore frameworks for smart objects [3,4] that identify some of the attributes that make an object smart. Meanwhile, rather separately, the 3D virtual world community has developed Second Life, OpenSimulator, Open Wonderland, Open Cobalt, and many other virtual worlds. Our *Everything is Alive* project at University of Arkansas [5] initially focused on RFID middleware [6] but realized we could use 3D virtual worlds to model the future Internet of Things [7]. Our work differs in that we believe a smart object is more or less smart depending on the standard protocols it supports (which can change over time). It also differs in that we use 3D virtual world technology to construct and demonstrate the protocols in an understandable manner. Two potential advantages of using 3D virtual worlds to understand a future smart semantic world are: modeling is low cost when compared to developing a deploying real world technologies, and it may be that modular services that we develop for interoperating with virtual worlds will transfer more or less directly to the real world.

II. Protocols for Smart Objects

In the 1990s, a puzzle for the AI agent community was to distinguish what made an agent into an intelligent or mobile agent. Since 3D virtual worlds can be viewed as composed of objects (agents), the same puzzle occurs. Our resolution is to identify a family of protocols that, if followed, enable us to classify objects as more or less smart objects. Therefore, we believe that what makes a smart object smart is the protocols it obeys.

Today's ordinary objects (a chair, a lamp, a can of corn, a pet, …) have interfaces. For example, a lamp has a physical interface consisting of size, shape, flexibility, weight, and composition; a visual appearance interface with aesthetic properties including color, brightness, and texture; a functional interface with an application program interface (API) that humans use for turning the lamp on or off; a power interface for connection to the electric grid; an implicit identity so people can tell two lamps apart even if they look the same; an implicit ownership (the new car I saw at the dealer; I just bought it so now its mine and used); and a compositional interface typically used for repairs. Objects may also have a corresponding repair manual (typically kept at home in a drawer), associated images that appear in retail catalogs or in photographs, and a history and/or schedule of use. This list is not complete.

Conventional objects are typically not be very smart – they do not have explicit identity, are not self aware, and cannot interoperably communicate with other smart objects or with humans. We need to begin to develop an initial framework for making objects smarter.

What additional interfaces would make an ordinary object into a smart object?

- Explicit identity – Explicit identity can be implemented using RFID tags or by other means [8]. Identity provides a way (for machines) to address each object uniquely. A legal ownership and an object's type are additional interfaces related to aspects of identity. Nearly everyone could afford hundreds of RFID tags (at $.07 each) to explicitly identify all the objects they own (though, of course, it is not yet cost effective to do this with inexpensive items at grocery stores). Local identities could be used within an enclave so only members know the mapping to global identities – so the RFID tags in your home cannot be meaningful if read from outside.

- APIs supported – A smart object can support one or multiple application program interfaces (APIs), and these different APIs might be available for different purposes and to different personnel. The owner may be able to use the object (turn the thermostat on or off, up or down), but a qualified repair person might be required to repair it.
- Security – Not just anyone should be able to command, control and communicate with my possessions. Access control could be used to specify any user's digital rights along with encryption to communicate securely over less secure channels. Many objects will only communicate with their owners or qualified repair person.
- Object-to-object communication – A networked object is an object that humans or other objects can communicate with. The network can be wired or wireless, LAN or WAN, use 802.11* or RFID, and some messaging language. There may be several messaging languages such as SNMP or WSDL.
- Human-to-object communication – A person needs a way to command, control, and communicate with smart objects. Assuming a person has a way to designate a device and can upload information about that device (its ownership, API, …), then, a GUI or menu-based interface could be used to control or query the device (possibly from a remote location).
- Micropayments – There may be a cost to accessing, communicating with, or using an object which one does not own.
- Plugins – A basic device might be extendible with plugin behaviors. A simple thermostat might only be able to turn on or off and turn temperature up or down, but it could be extended with a scheduler plugin for scheduling time of day and days of the week and/or with a history logging plugin for remembering all past settings (useful for calculating energy usage, another plugin).

Is this protocol list complete? No (other useful protocols are mentioned below). Each bullet needs refinement and one could argue about any or many of the characterizations. For instance, implicit identity is sufficient for many purposes – "buy me one of those lamps--I don't care which one."

Does an object have to support all the interfaces to be smart? Is there a core set? One possible answer is "no" – a degenerate smart object might contain none of the additional interfaces as long as it is possible to add additional interfaces from the list. The binding time for adding smart object protocols could be during design, assembly, or dynamically, during use, on an as-needed basis. As interfaces are added (or removed), the object becomes increasingly smart.

A significant challenge for widespread adoption of smart objects involves reducing complexity while increasing functionality. Today, managing 5-10 network objects is challenging and requires humans to run virus scans, set up firewalls, change permissions, run defragmenters, and download security updates. Many users (e.g., the elderly) are challenged by this complexity and just want unintelligent, simple, and reliable objects that lower maintenance requirements. This means smart complex objects will compete with unintelligent conventional objects on criteria like cost, reliability, functionality, and ease of use. In a world where every user controls hundreds to millions of smart objects, having hundreds or millions of separate remote controls (one per object) does not scale, so truly universal remotes (e.g. smarter smart phones) will be needed – we call these soft controllers (think Star Trek communicators – see [9]) because they import different object interfaces from the objects and network. Also, different users may see the object differently – so one user has a simple controller and another has a more sophisticated controller (solving the problem of hitting the *input* button on a TV remote and not understanding how to reset it, a typical problem with edging today's remote controls that plagues naïve users).

III. Using Virtual Worlds to develop Smart Object Protocols

In the future, when people go to the store, buy a smart object and bring it home, a 3D model of the object will be installed into the virtual model of their smart home (another protocol). Changes people make in the real world will or may affect the model and vice versa – a bidirectional mirror world [10].

In the meantime, before the real world converts to smart object protocols, we need to understand how such a world will function. What will it be like to manage and maintain thousands of smart objects, especially when today many of us have trouble maintaining under ten complex fairly dumb objects (laptops, stereo and TV, and a drawer of user manuals). Certainly, we do not want to have to remember to set manual permissions on the TV channel by channel when a house guest visits (but every family member might still want their own list of favorite channels via a personalization service). The world needs to become simpler, not more complex. Therefore, we need uniform and simple ways to manage a smart world.

IV. "Intelligence" does not always reside in objects

People tend to think that to make an object smart, the object itself has to be smart. But, do all the smarts really have to be located inside the object? No, though some might be. Here is a simple algorithm for making a smart world. Add item level RFID to many or all objects. Add an RFID reader to a smart phone (the way GPS was recently added to cell phones and RF plugins are now being added to control one's TV and stereo). Since the smart RFID enabled phone can now read the tags of any object and since the phone is already connected to the Internet, all information about the object can be downloaded from the web-cloud. Chairs with RFID tags will immediately become smart.

We have built a bookshelf to demonstrate this idea first (see Figure 1). People usually do not consider a book itself to be a smart object. However, in the real world, any published book is assigned a unique ISBN. We can use this identifier to retrieve more information about each book. This setting assumes that each book is labeled with some identifier like RFID which stores ISBN number. Therefore, if you scan any book with a mobile RFID scanner, which is wirelessly connected to the Internet, you can get additional relevant information from the Internet.

Figure 1. Smart Shelf

Following are capabilities that illustrate a smart bookshelf containing smart books.

- A user's avatar wanting to search for a book (using its title or ISBN number) can ask the book shelf for the book's location. The bookshelf consults a remote database, finds the book and displays the location.
- A user asks a book for more information, its ISBN is used to remotely query Amazon for basic information on the book or reader recommendations. Received information is displayed on the screen of the remote control smart phone device.
- The smart shelf can periodically inventory itself for smart books and then sends updates to the remote inventory database.

Of course, to get full value, future devices will need to be manufactured with network controls so that people can remotely control their behaviors. This is not to say that smart objects will themselves contain no processing; rather, knowledge and processing that makes a smart object smart might be contained in the object, the universal controller (see below), the user, and/or various information sources on the Internet and different smart objects might distribute this information differently (for instance the Internet might be only intermittently available and smart objects might need to cache some of their log history to upload it later.

V. Prototype of Smart Objects and Soft Controller

3D virtual worlds give us a way to learn to manage and manipulate smart objects, and simulating those objects in a virtual world helps people imagine how new devices would change the world. Also, since developing and testing in a virtual world may eventually require less cost than in the real world, this approach to prototyping and testing could provide advantages over real world prototyping and testing. In all likelihood, the smart object interface protocols we develop can be platform-agnostic, so they can operate either in the real or virtual world.

To experiment with some of the smart object protocols, we developed a collection of smart healthcare objects in the virtual world Second Life (http://secondlife.com). Second Life is a reasonable platform for this experiment - but it has limitations as a long-term virtual world platform, in part because its server farm is owned by Linden Labs. But we could have performed the same demonstrations in OpenSimulator (http://opensimulator.org).

We visited the University of Arkansas School of Nursing's training facilities. One of our interests was to determine how to overlay training scenarios on virtual world architectures (considered briefly later in this paper). This paper mainly focuses on the subproblem of how we built smart objects needed for the training. Screenshots appear below to give the idea of what we developed (see Figures 2-8), and videos are available on the web to see the function of these objects (videos at http://vw.ddns.uark.edu/index.php?page=media: infant warmer [14MB], device controller [6MB], IV Drip [3MB]). We developed a collection of smart objects as follows:

- A hospital bed has several functions to make the patient feel comfortable. The angle of the bed is adjustable. The bed has a table which users can pull out.
- An air conditioner/heater is attached to a wall of the room – it can be turned on or off or from cool to warm visually displayed with blue and red particle effects.
- A nursing dummy the same scale as a human is used for training a nurse in real life. We developed an infant dummy. The dummy can be opened to show its internal organs. Each organ displays its name when it is clicked so that the nurse-in-training can learn which organs are which.
- An infant warmer is a machine which keeps a baby warm. It has mechanical arms to give the infant oxygen and measure his suction. It can display an X-ray from a nearby portable X-ray machine.
- A portable X-ray machine has a screen where the digital picture of the X-ray taken by this machine is displayed. When the machine is clicked, it moves its arm upward and approaches the object. Then, it accesses the X-ray sheet in the infant warmer and displays it on the digital screen. When clicked again, it goes back to its original state and turns the digital screen off.
- A search robot roams the virtual healthcare clinic to search for and catalog other smart objects - which are objects that obey our reflection protocol that will download their API to the search robot from a database we keep in a web cloud. The robot has a remote control and is able to leave the users sight to discover new smart objects as it traverses the clinic independently. A GPS control is a handheld device which has scheduling capabilities. It can store the current location of the user as he enters the checkpoints which the robot will follow. This way the user can create previously fixed paths which the robot will now be able to traverse by itself. An RFID tag is an identification tag which responds to an RFID source. It responds by providing identification information and its location.

Figure 2. HILL-ROM Stabilet Infant Warmer

Figure 3. Infant dummy in infant warmer

Figure 4. Modeled infant warmer machine with multiple functions

Figure 5. Nursing dummy which has several training function for nurses

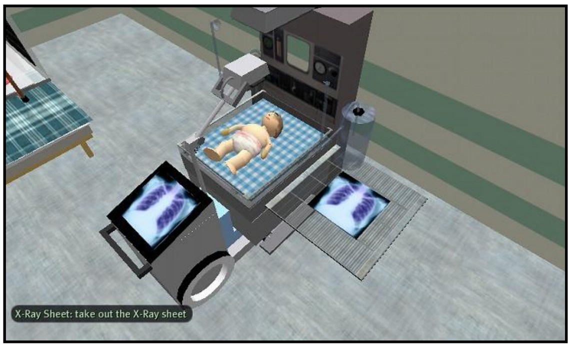

Figure 6. The portable X-ray machine can X-ray the baby

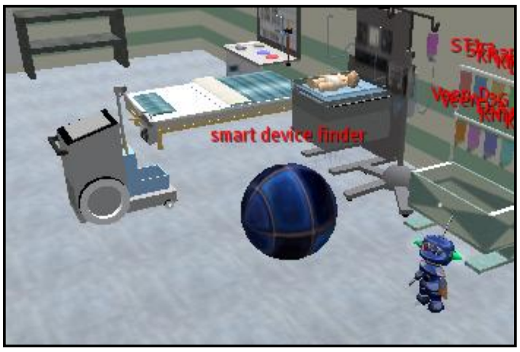

Figure 7. Smart object locator

Figure 8. In-world soft controller can control any smart device

In addition to the scripted smart objects described above (all somewhat smart based on their functioning scripts), we developed a protocol that all of our smart objects follow so that they can be controlled in a uniform manner. All our smart objects in Second Life use a `listen()` event handler (using Linden Scripting Language). This is executed when the `llListen` function receives a chat message that satisfies a condition in an assigned channel. As a result, all devices accept commands from external sources, either from an avatar or from other smart objects. [Off course, we are not saying `llListen` is the way all Internet of Things objects should reflectively provide their APIs - just that such a general discover API protocol would be generally useful.]

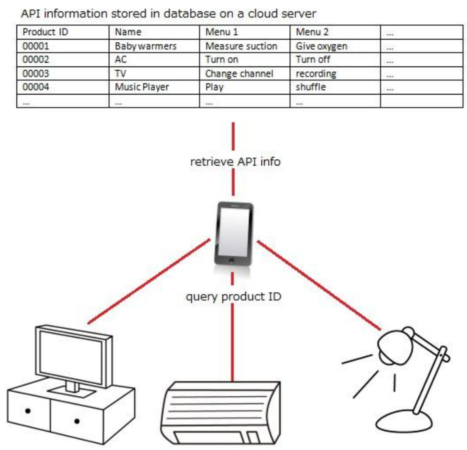

Figure 9. Architecture for API discovery

Humans need a way to control smart objects near them. A controlling device (Figure 8) shows the menu of functions for each smart object, which is retrieved from an online database by sending a query with a set of unique ids associated with each smart object (Figure 9), on the soft controller screen. Avatar users can choose from the menu and control the devices. The controller has text based instructions on its screen. The names of smart objects in range are displayed on the screen, and users are asked to choose one. Then, the functions of the chosen device are displayed on the screen, and the avatar can choose from the list and send the command to the device. Also, it is important to point out that since we use the same format of input for each smart object, there can be more than one controller. An example is a web-browser based controller that also uses the soft controller architecture (Figure 10) [12].

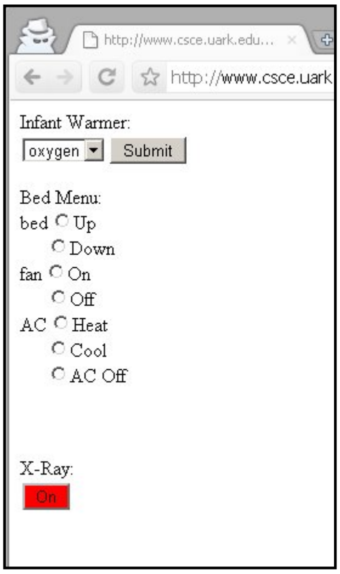

Figure 10. Web browser based controller

Instead of a frequency band like infrared in the real world, in Second Life, various channels are used to communicate between devices and avatars or devices and other devices. All smart objects are in one sense connected to each other because all the devices can be controlled by a soft controller. Although each device has a different use and different commands and works independently, by implementing a common format of the input, they have a universally formatted API.

While the implementation described above is Second Life specific, a similarly functioning implementation could be implemented for other virtual worlds or for smart phones in the real world. For instance, we could have used a web service discover the API of smart objects.

- Extending smart phones with RFID readers and smart objects with network actuators. Building phone apps for smart objects. Interfacing phone apps not just to the real world but to virtual worlds.
- Arranging the hundreds of thousands of real world smart objects into "lower" or product ontologies to make it easier to develop protocols using categories and inheritance [13].
- Identifying additional smart object protocols (e.g., touch, taste, smell).
- Making objects even smarter [14].

Standards will be needed. By our description of smart objects, some objects are already somewhat smart, and more are becoming smarter every day, so a migration path is already in place. There is a manifest destiny that more and more object types will become smarter. So far, this is happening application by application (for example, smart home entertainment, security systems, and washing machines - but these are not typically not interoperable). To get the most value, interoperability standards will be needed to enable plug-and-play so that all objects obey a suite of smart object protocols, possibly with many implementations. Understanding more about such a suite and testing it early can accelerate progress toward a universally smart world. As we said above, virtual worlds give us a way to design and test these protocols in the near term.

VI. Training

One additional result from our work regards training. We developed two training scenarios:

- The first involved training to monitor the patient's blood oxygen supply. The infant mannequin has a function of saturation of peripheral oxygen (SpO2).. When a trainer avatar types "/5 start SpO2", the infant starts with 100% SpO2 but the level decreases. The infant's face looks paler as it loses SpO2, and if it goes under 75%, the infant dies. As SpO2 goes under 95%, a monitor shows the message "problem zone", and as it goes below 85%, an alarm message indicates "danger zone." When the nurse avatar in training administers oxygen, the infant gradually stabilizes until it returns to 100%.
- In the second training exercise is shown in Figure 11. The training serves the purpose of virtually giving nurses practice in setting the proper infusion bottles. Training is begun by touching the console on the IV drip stand. A prompt instructs the nurse to set a certain bottle. If selected successfully, a new bottle is prompted, and this continues until training is complete. Once training is complete, an overall score is produced.

Actual nursing mannequins are expensive and a nurse-in-training must go to a nursing school or similar facility to train with them. We originally conjectured that using virtual worlds to simulate nursing dummies and associated procedures could accelerate nurse training for nurses anywhere in the world at any time at no cost. We still believe our conjecture is valid but to a more limited extent – the virtual world can familiarize nurses-in-training with devices, their operation, and with procedures and thus can be used for training. However, certain actions such as learning the physical action of administering a shot or the fine motor skills needed to open a latch still require some hands on experience. We should note that we did not actually train nurses using our two training scenarios - our interest was whether we could involve smart objects in training protocols and how difficult it would be to set up such protocols since experience in setting up serious training simulations is still very limited in virtual worlds (though some work has been done at Imperial College London and others).

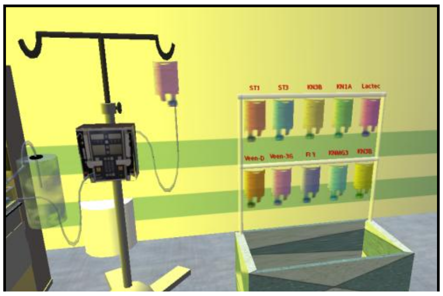

Figure 11: IV drip machine, which has a function to change the bottle.

VII. Potential Impact

We humans are pretty self-centered. We think being smart distinguishes us from reactive objects like thermostats and pets and passive objects like chairs. But this is about to change as we associate knowledge, action, and rules with objects in the world around us. We can learn how to do this using 3D virtual worlds but this also helps us understand how to translate these capabilities to the real world - so putting RFID tags (or barcodes) on objects, we can use smart phones to identify and associate information with objects and that information can include means to control the objects as well as security APIs to insure only authorized personnel can fire a gun or turn up the thermostat.

An interesting exercise is to consider an object and ask, if this object could talk, what would I want to ask or tell it. It might know about its manufacture history, its similarities and differences to other types of devices, its maintenance requirements and history, its location and environment, etc. Simulating devices in a virtual world potentially provides people with a new means of understanding how devices operate, and how to repair devices – potentially a new, more interactive approach to a traditional training manual or training video. Usually, to create a real-world test model requires significant funds; however, a virtual world simulation often is much less expensive and can be available anywhere in the world for low or no cost. For instance, instead of spending tens of thousands of dollars on training manikins, a health care training service could provide a virtual mannequins for no cost available world-wide. Although there will be some differences between the object in the virtual world and the real world, we can view simulations as having a useful place in our view of how we can control the real world.

A problem we had with many scripted projects in Second Life is that avatars other than the developer do not know that the object is scripted or how to operate it. Even if a device has many functions, it will be useless if a user cannot learn how to control the device. Therefore, it is important to focus on not only the communication between devices but also on the communication between the devices and people.

Establishing a standard interoperability infrastructure for smart objects makes it possible to mass produce interoperable smart objects (both in the real and virtual world) that are available to users anywhere in the world, accelerating our move toward a smart world. Creating a unified, extensible standard protocol for controlling smart objects solves this problem and makes it possible to control all such devices from a standard controller device (e.g., a smart phone). The controller can upload the controls from any smart network device, including devices it has never encountered before. Separating the interface of a device from the implementation benefits end users and developers for the same reason that pull down menus benefitted them in the 1980s giving a common look-and-feel to a wide variety of applications. Developers because separating the interface from the device can reduce costs of designing physical interfaces (where there are no standards), and the end user benefit because it becomes easier to control devices a person has never seen before because the interface style can be familiar.

With a uniform interface for smart objects, it becomes easier to build higher level interaction protocols for controlling assemblies of objects. Many of the "business rules" (another protocol) for such assemblies are application-specific, but the ability to see physical objects as exporting their interfaces in an object-oriented programming style bodes well for providing higher level mechanisms for composing them together.

Just as the World Wide Web uses URLs to link information, we can use virtual world URLs (which include a region and x/y/z location) to teleport to a location in a virtual world. We can similarly use RFID tags and smart phones to locate objects in the real world. The real or virtual objects have unique identities. We can associate additional ontology information with these identities (in web or cloud based data sources) and associate information and rules with these objects. In this way, we can view our work as extending the "semantic web" directly toward a "semantic world" where more information about any physical thing and the ability to control those things (subject to access control permissions) may be available to humans via their soft controller smart phones.

VIII. Future Work

Areas for future work include:

- Determining and removing limitations of Second Life and other 3D virtual worlds as a simulation platform [11]. Integrating virtual worlds as web browser add-ons so that following a link can lead to a web page or a virtual world in uniform manner.
- Determining standards for representing 3D objects – SL *prims* (primitive graphics objects in Second Life) do not mesh well with gold standard representations like CityGML/Collada.
- Determining how to represent interfaces in a general manner (e.g., can we use WSDL or other already standard approaches)
- Gaining experience in combining the smart object protocols and implementing them in a variety of ways including using smart phones as platforms.

References

[1] T. Berners-Lee, J. Hendler, O. Lassila. "The Semantic Web", *Scientific American*, , May 2001, pp. 29-37.

[2] G. Kortuem, F. Kawsar, D. Fitton, V. Sundramoorthy. "Smart Objects as Building Blocks for the Internet of Things", *IEEE Internet Computing*, vol. 14, no. 1, Jan./Feb. 2010, pp. 44-51.

[3] S. Charlton. *Ubiquitous Computing: Smart Devices, Environments and Interactions*, John Wiley, New Jersey, 2009.

[4] P. Jorissen, M. Wijnants, W. Lamotte. "Dynamic Interactions in Physically Realistic Collaborative Virtual

Environments”, *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 6, , 2005, pp. 649-660.

[5] C. Thompson. “Everything is Alive”, *IEEE Internet Computing*, Jan./Feb., vol. 8, no. 1, 2004, pp. 83-86.

[6] J. Hoag, C. Thompson. “Architecting RFID Middleware”, *IEEE Internet Computing*, vol. 10, no. 5, 2006, pp. 88-92.

[7] C. Thompson, F. Hagstrom. "Modeling Healthcare in a Virtual World," *IEEE Internet Computing*, vol. 12, no. 5, 2008, pp. 80-84.

[8] C. Thompson, D. Thompson. “Identity Management”, *IEEE Internet Computing*, Vol. 11, no. 3, 2007, pp. 82-85.

[9] C. Thompson. “Smart Devices and Soft Controllers”, *IEEE Internet Computing*, vol. 9, no. 1, 2005, pp. 82-85.

[10] P. Martin et al. “Mirror Worlds”, *X10 Workshop on Extensible Virtual Worlds*, venue: Second Life, 2010, http://vw.ddns.uark.edu/X10.

[11] K. Perkins. “Virtual Worlds as Simulation Platform”, *X10 Workshop on Extensible Virtual Worlds, venue: Second Life*, 2010, http://vw.ddns.uark.edu/X10.

[12] J. McCrary, T. Kumar, C. Thompson, "Communicating with Smart Objects via a Soft Controller and PIE 2.0”, *X10 Workshop on Extensible Virtual Worlds, venue: Second Life*, 2010, http://vw.ddns.uark.edu/X10.

[13] T. Censullo, T. “Ontologies for the Virtual World”, *X10 Workshop on Extensible Virtual Worlds*, venue: Second Life, 2010, http://vw.ddns.uark.edu/X10.

[14] S. Soares, T. Takáo, A. da Rocha, R. de M. Araújo, T. de A. Barbosa. "Building Distributed Soft Sensors," *International Journal of Computer Information Systems and Industrial Management Applications*, Volume 3, 2011, pp. 202-209.

Author Biographies

Akihiro Eguchi is a senior at the University of Arkansas double majoring in computer science and psychology with a minor in mathematics. His interests include smart world technologies, artificial intelligence, computational linguistics, and image processing.

Craig Thompson received his BS degree in mathematics from Stanford in 1971 and his PhD degree in computer science from the University of Texas in Austin in 1984. He is currently a professor and the Charles Morgan Chair in the Computer Science and Computer Engineering Department at the University of Arkansas, where he leads a research project called *Everything Is Alive* related to pervasive computing.