
Our article focuses on the role component-based software may play in realizing smart environments. A scenario describes a future smart environment in a university campus setting. The scenario illustrates how software components might contribute to the various phases of development, deployment and use of smart environments. Key research areas identified include: development of component frameworks, component interface standards, addressing schemes, and protocols.

Component-Based Software Systems for Smart Environments

Charles Herring and Simon Kaplan, University of Queensland

This article was presented at the DARPA/NSF/NIST Inter-agency Workshop on Smart Environments held at the Georgia Institute of Technology on 25-26 July, 1999. The call for participation asked for white papers to address research directions in smart environments. Also, the call for articles asked that a scenario be developed to illustrate how the proposed research would help in the realization of smart environments. Our scenario shows how smart environments could be designed, implemented, and deployed based on software components. We incorporate into the scenario some enabling technologies and approaches we have been investigating. Each phase of the scenario is discussed in terms of needed research. A summary of research directions for component-based smart environments is given.

Scenario

On a Campus far, far away ... The new Information Environments Center building is being completed, just in time for the start of the semester. An engineer at SmartEnvironments Pty Ltd is using the SmartComponents™ WorkBench to assemble the component software system that will be installed in the new building. The WorkBench design package provides prototype software components for buildings, levels, and rooms. These components are based on the *de facto* industry standard CyberNetic™ Control Model for advanced feedback control systems. Additionally, these component containers implement the SmartAddressSpace® protocols that permit them to seamlessly interoperate with other smart spaces in the system. Working from the building plans and the list of sensors, controls, furnishings, and appliances, the engineer creates a model of the building. This model is easily constructed by drag-n-drop of components from pallets in the WorkBench. Based on projected usage profiles developed during the design of the building, the engineer runs a simulation of the building for the next year. Satisfied with the component system design, the engineer downloads the software into the building, level, and room servers.

Later that day... Workmen begin delivering and installing the furnishings for the new building. In a commons room a workman hangs a picture on the wall. He hangs a clock on the opposite wall. A lounge, several chairs, and a table are also placed in the room. In an adjacent room a workman hangs a picture and clock on the wall. A desk and chair are brought in. Also, a table is wheeled in and a printer is placed on it and plugged in. (The picture is a SmartRoom server and contains

various sensors that monitor and control the room. The SmartRoom server establishes connection with its level server and downloads specific policy, planning, and execution instructions. The clock contains a camera, motion detector, and other sensors that are used by the room server. The other furnishings (lounge, chair, table, etc.) are also "smart" in that they contain embedded systems that communicate with the room server and the other furnishings. In this manner the room discovers what furnishings and appliances it contains and associates specific components with them in its control model. The room server uses the SmartAddressSpace protocols to establish contact with adjacent rooms and learns what services they provide (e.g., the printer in the next room).)

The next day... As students arrive the campus packet radio network downloads class schedules, campus maps, security access codes, etc., into their SDA (Student Digital Assistant). Students in the Information Environments program check their schedules and are guided to the new building. As they approach the building, the campus packet network hands their SDAs over to the building's radio frequency network. The doors open (based on their security codes) and they explore the new building. In the commons room two groups of students spontaneously form a wireless (infrared) network to start work on their first assignment. One group finishes and sends the assignment to a printer. The smart room server locates the closest printer (the one in the next room) and sends the print job to it. The students are informed where they can pick up the printout. (The smart building, level, and room controllers are recording and monitoring the comings and goings of the building's users, their activities, as well as the other environmental variables. From this information the room servers learn how the building is being used and to anticipate future requirements.)

Scenario Analysis and Implications for Smart Environments

The scenario above illustrates three distinct phases of a smart environment: design and implementation, deployment, and use. We now analyze each of these phases and discuss their implications in relation to component software systems.

Design and Implementation – We think the component-oriented approach to software development, coupled with commercial component marketplaces, represents the next

stage in the evolution of software practice. Accordingly we postulate that smart environments will be based on component software. Further, in our scenario we propose smart environments will be designed and engineered alongside and in the same manner as other aspects of the built environment. We used several “props” to illustrate this. First, the engineer used a smart environments CAD package to design the software for the new building. This package allowed him to choose from a range of third-party components to model and simulate the levels and rooms in the building. This presupposes that manufacturers of the various hardware systems that go into the building will make available compatible component software for inclusion in overall smart environment systems.

We mentioned two specific facilities: the Cybernetic Control Model and the Smart Address Space. Smart environments are complex systems requiring principled design. Thus far researchers have concentrated mainly on the development of independent technologies and techniques. However, now researchers are beginning to look at overall system architectures [1]. We are exploring the use of cybernetic control theory as a basis for modeling organizations and smart environments. Smart environments are not passive; they are active. We see elements of control theory leading to software component architectures that go beyond robust stability to permit planning and anticipation to better support users. Our first steps in this direction are described [2, 3]. Also, we think new and “richer” address space models are required. We have begun looking at this issue [4].

Deployment – A major feature of smart environments brought out in this phase of the scenario is device and service discovery, self-configuration, and self-organization. As the various servers (the picture) and sensors (the clock) are plugged into the building network, there is a discovery/lookup/join protocol that permits them to determine their roles as participants in the overall building infrastructure. Similarly, the furnishings (chairs, desks, etc.) and appliances (printer) have embedded systems that permit the rooms to register them. There must be protocols for the room server to download the component software that represents these elements. Most likely these will take the form of proxies within the room services component container. These proxies provide the interface needed to communicate and control the device. We believe most furnishings and appliances will be equipped with such embedded systems and communications will be primarily wireless. Once again we see a central role for a Smart Address Space in the task of coordination and control of the myriad devices and sensors.

Use – Another important aspect of smart environments is that they are composed of many hierarchical, nested, and overlapping address spaces. Smart environments are both systems-of-systems and systems-within-systems. In this scenario the students are in constant communication with the campus-building-level-room-personal system. At any one time they may be using one or more of these wireless and wireline systems. The wireless overlay communication system represents yet another facility that must be incorporated into the smart environment. Although we have used physical spaces to motivate this scenario, smart environments are not constrained by physical boundaries. The groups of students establishing spontaneous networks for a collaborative task illustrate this. This mode of operation will be the norm in civic places such as airports and shopping towns.

Research Directions for Component-based Smart Environments

From the above scenario and analysis we see the following as major research directions for component-based smart environments:

- Development of component frameworks for smart environments based on control theory models to integrate existing and emerging embedded devices into self-organizing, self-regulating, and adaptive systems. In the near term, these frameworks will most likely take the form of advanced component containers (based on EJB or COM+) to make up the middle-tier of an overall three-tier approach.
- Development of software component interface standards for use in describing, finding and controlling hardware components. As the number of different types of embedded devices grows and their complexity increases, it will be increasingly important for industry standards to emerge. Software component standards should eventually permit for modeling and simulation of entire systems built from the hardware components.
- Investigation of new addressing mechanisms and schemes to overcome limitations of current systems. Such addressing systems must take into account the extremely large number of potential devices, mobility, and high volatility. They must also facilitate multi-modal addressing based on name, identifier, location, time, and content.
- Development and integration of protocols for the many layers and complex interactions needed within smart environments [5, 6]. There are currently many ongoing efforts (e.g., Jini, Bluetooth, 1394, Universal Plug-N-Play, CEBus, etc.) to develop new protocols. Which ones will survive, how they will interoperate, and what new ones are required remains to be seen. For example, in our scenario we assumed integrated packet area radio, radio frequency, and infrared networking: this is not commercially available [7].

References

- [1] S. Shafer *et al.*, “The New EasyLiving Project at Microsoft Research,” *Joint DARPA/NIST Smart Spaces Wksp.*, July 30–31, 1998, Gaithersburg, MD, USA.
- [2] C. Herring, “Viable System Model and Component Software,” <http://www.dstc.edu.au/TU/staff/herring/Vsmc.pdf>, 1998.
- [3] C. Herring and S. Kaplan, “Cybernetic Components: A Theoretical Basis for Component Software Systems,” *Component Oriented Software Engineering Wksp. (COSE’98)*, 1998, Adelaide, Australia: (<http://www.dstc.edu.au/TU/staff/herring/cose98-ref.pdf>).
- [4] C. Herring and S. Kaplan, “Manifolds: Cellular Component Organizations,” *Proc. 28th Int’l. Conf. Technology of Object-Oriented Languages and Systems*, TOOL Pacific, 1998, Melbourne, Australia: (<http://www.dstc.edu.au/TU/staff/herring/awsa98-herring.pdf>).
- [5] K. Almeroth, K. Obraczka, and D. D. Lucia, “Pseudo-IP: Providing a Thin Network Layer Protocol for Semi-Intelligent Wireless Devices,” *Proc. DARPA/NIST Smart Spaces Wksp.*, July 30–31, 1998, Gaithersburg, MD, USA.
- [6] J. Heidemann, R. Govindan, and D. Estrin, “Configuration Challenges for Smart Spaces,” *Proc. DARPA/NIST Smart Spaces Wksp.*, July 30–31, 1998, Gaithersburg, MD, USA.
- [7] R. Katz and E. Brewer, “The Case for Wireless Overlay Networks,” *Proc. 1996 SPIE Conf. Multimedia and Networking*, MMCM’96 1996, San Jose, CA, USA.

Biographies

CHARLES HERRING (herring@dstc.edu.au) is a Ph.D. candidate in the Department of Computer Science and Electrical Engineering of the University of Queensland. He is investigating cybernetic, control theory and systems science models to aid in the development of software system architectures for complex applications including business-to-business e-commerce and smart environments.

SIMON KAPLAN (kaplan@dstc.edu.au) is professor and head of the School of Computer Science and Electrical Engineering at the University of Queens-

land, where he is also principal scientist of the CRC for Distributed Systems Technology. Simon received his BSc(Hons) and Ph.D. degrees from the University of Cape Town and was assistant and associate professor of computer science at the University of Illinois before relocating to Queensland in 1995. His research interests include software engineering, systems thinking, distributed systems, collaborative work, human-computer interaction, and applying design-based problem solving approaches to the teaching of computing.