SDBI: An Ontology Based Smart Home Lab Environment

Antonio Sanchez[1], Ricardo Tercero[2], Diana Saldaña[3], Lisa Burnell Ball[1]

[1]Texas Christian University, [2]LANIA, [3] UABC
[1]Fort Worth, Texas, 76129, USA
a.sanchez-aguilar@tcu.edu

Abstract. This paper discusses development of an ontology-based smart home control system and the active learning pedagogy employed within our apartment-style lab. The system employs Sun Spots and standard cameras for sensing and FireCrackers and X10 devices for actuation. The software is written using Java, the JESS rule-based system shell, and Protégé, an OWL compliant ontology development tool. Furthermore, it is designed to integrate with our existing smart home infrastructure and systems. A key objective of the lab is to mentor student researchers (both graduate and undergraduate) in a multidisciplinary, collaborative environment. The chosen domain of intelligent environments, e.g. smart homes, affords us many opportunities to achieve this objective. For the past three summers, we invite international graduate students to work in the lab with our own undergraduates, all with close faculty mentorship. We have learned a number of lessons from these experiences, some of which we report here.

"Give the students something to do, not something to learn; and if the doing is of such a nature as to demand thinking ... learning naturally results"

John Dewey (1859 – 1952)

Keywords: *Intelligent Environments, Smart Homes, Active Learning, Ontological engineering*

1 Introduction

A smart home can be considered the integration of various automated services intended to enhance the lives of its inhabitants. This integration is done mainly by software that controls the many physical devices commonly found in homes, like kitchen appliances, lighting, and fans. Communicating with these devices involves trade-offs, including resource availability and desirability. For the sake of argument, let us consider a grandmother living in a 50 year old house, she may not have the resources or desire to completely change the way she performs daily activities like cooking, bathing, or watching her favorite television programs.

The goals of TCU Crescent Lab for Intelligent Systems are twofold: (1) to investigate smart environments to improve the lives of others, and (2) to mentor students by immersing them into applied research. The Crescent Lab essentially is an apartment, complete with a working kitchen, furnished living and dining areas, and spaces for one bedroom and bathroom.

Smart homes, and other intelligent spaces, continue to attract us with their promise of anticipating and meeting our needs as they unobtrusively adapt to our changing preferences and goals. The delivery of this promise, however, has met with limited success in terms of functionality and consumer acceptance. Our interest lies in studying the high-level reasoning necessary to exploit the next generation of smart home devices; we seek the *sweet spot* of automation between available comprehensible systems of limited flexibility and potentially powerful yet unproven autonomous robotic systems that are error-prone and complicated.

Finding this *sweet spot* involves investigating approaches for decision-making, adaptation, representing domain-specific knowledge, and new user interfaces. Among other aspects we can cite the construction of intelligent multi-agent systems that integrate function specific agents (e.g. lighting, temperature control, image, audio and image recognition) with the appropriate interface through mediator agents such that new standards, protocols, and components can be added over time.

As stated before, we are committed to the development of a learning space for students; a space that allows them to be creative problem solvers who communicate and work with others. In this context students are given the opportunity to work individually and in teams on complex, open-ended problems, under the mentorship of the faculty. We invite international graduate students to concentrate on their research but focusing in a current topic within the smart home lab to develop a demo prototype to be used later by undergraduate students as a starting point. It is clear that in any collaborative endeavor, especially one that includes expectations and pace of work of young minds can lead to miscommunication, failed deadlines, and rework. It is important to keep students engaged and this requires that they feel appropriately challenged by interesting activities. To fulfill this double mission we invite graduate students to be more purposive (i.e. goal definers) while the undergraduates are guided to be more purposeful (i.e. goal achievers). This is perhaps the greatest challenge, and greatest reward, of our lab environment. A discussion of this effort and their past results has been presented elsewhere **[1].**

Smart home technology serves another important role when considering care for the elder. Aging is often accompanied with health problems that often lead to a decrease in quality of life. To live with autonomy in their homes, they need assistance with daily activities in different ways: they may need help to complete an activity **[2].** Elders may need to be warned when facing risks associated with performing an activity; as they suffer from natural cognitive impairment due to aging, they may forget some events, situations or tasks than impede completing an activity appropriately or suffer some risks during the activity. This is another important reason to focus our research on smart home technology.

We use an ontology driven system which has the necessary rules for the desired performance of the appliances. Our smart home research is eclectic in nature due to the fact that it is mainly designed as a student learning aid; moreover we feel that it is this precisely this type of environment that entice students to get involved in AI research.

2 SDBI Architecture

The general framework of the SDBI architecture is designed to operate within a home environment, sensing the environment and inhabitants, reasoning about the appropriate actions, and responding accordingly (**Figure 1**). For example, in an elder care application scenario, a smart home should respond to the needs and disabilities of elderly inhabitants to prevent injury or death. Additionally, the home should provide commonly available capabilities to aid the inhabitants and their caregivers, e.g. health care appointment reminders, medication monitoring, inventory management of basic necessities like food, and remote monitoring. We will focus on the safety monitoring components in this article, for which the system needs to sense the environment, determine risk, take some immediate actions, and notify caregivers.

The SDBI software architecture is based on the following five components:

- Sensor Information Acquisition
- Sensor Communication
- Sensor Information Fusion
- Decision Model and Ontology
- Actuator Controllers

We employ the Model-View-Controller (MVC) architectural design pattern to integrate diverse hardware and software components and to allow for iterative, incremental development. In previous work **[3,4]** we have explored a number of architectural and development models, during which we have identified a number of issues. Some are related to our mission to provide an active learning environment for undergraduate students. Complexity, sensor/actuator capabilities, component communication standards, and systems integration have all contributed to challenges in creating a unified smart home environment.

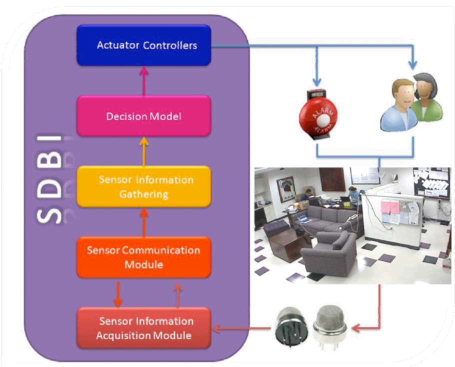

Figure 1. The SDBI System Architecture.

In the following sections, we provide an overview of each component in our most recent, simplified architecture. Later work will attempt to integrate previous capabilities, such as our smart kitchen, into this new architecture.

2.1 Sensing people and the environment: Sensors

Elder care requires a sophisticated sensor processing system and while others have made significant contributions in this area **[2],** our interests center on knowledge representation and reasoning, therefore we restricted the lab environment to sensors to detect the following:

(1) relative humidity
(2) ambient temperature
(3) movement,
(4) natural gas LPG
(5) Air Quality Control such as smoke, CO_2, CO and other pernicious gases

We used sensors from Hope Microelectronics**[5]** and SPOTs (Small Programmable Object Technology)from SUN Microsystems**[6]** to establish bidirectional communication with the server and also unidirectional with the required services (**Figure 2**). If the Base-Station finds at least one of the required services, it sends a Services Request to the Free-Range SPOT.

We were looking for a student-friendly hardware with software tools that were open-source and familiar to the student researchers. SPOTs allowed the students, already skilled in JAVA programming, to experiment with communication protocols for the SPOTs. The detailed implementation is provided in **[7].** We have found that our undergraduate students can become overwhelmed by the plethora of available hardware and software from which to develop prototypes, and it then becomes more difficult to keep them focused. This is perhaps one our biggest lessons learned, and as most are, a lesson that seems obvious to us now. It is easy to get lured by the promise of the "next big thing".

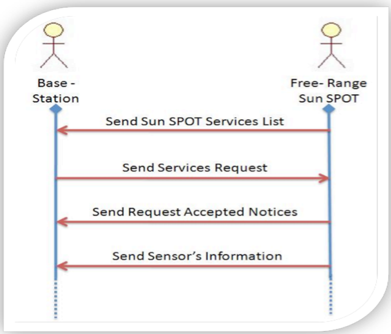

Figure 2. SPOT Communication Scheme.

All sensor data is transmitted to the decision model before any actuation occurs. Figure 3 depicts the sensors and spots used.

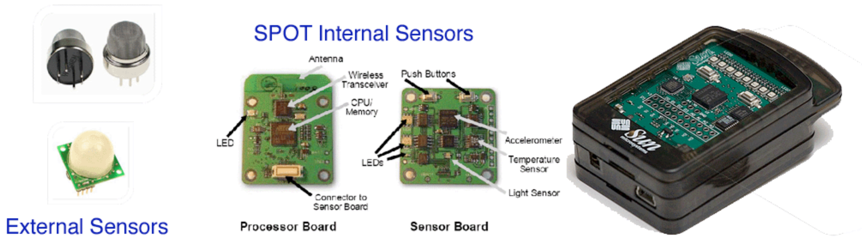

Figure 3. Sensors and Spots used (*source* **[5][6]).**

2.2 Deciding what to do: The Decision Model

We used the OWL specification language to define an ontology that relates sensor features and actions. Some of the objects and their relationships are shown in **Table 1** and **Figure 4**. The ontology defines the possible actions, environmental parameters (sensor inputs), and relationships between the inputs and the rules. A more detailed discussion of these type of ontologies is given is presented by the ELDer ontology proposed by Saldaña, Rodriguez and Gacia-Vazquez **[8],** the reader is invited to review it in the literature.A simple scenario is one in which the inhabitant gets distracted while they are cooking something on the stove. Getting absorbed in something on television, they forget about the can of soup they were heating on the stove, and it's a gas stove. A fire starts. Detecting a rapid change in temperature and smoke in the kitchen, the system decides, there is a high risk of a fire, so it sends a notification to the fire department and the caregivers, and activates exhaust fans and a fire-suppression system.

Table 1. Some common household risks. Note that the rules and actions are shown in a simplified form. For example, actions often will include notifying emergency responders through 911.

RISKS DETERMINATION AND ACTIONS		
Rules	**Risks**	**Notification**
If detect natural gas and inhabitant motion	CO poisoning, fire, explosion	Turn on exhaust fans, extinguish open flames, notify inhabitants and caregivers.
If detect smoke and a rapid change in ambient temperature that exceeds 100° F	Fire	Turn on lights and exhaust fans, notify inhabitants and caregivers.
???? If detect smoke and motion	Intoxication	Turn on lights and exhaust fans, notify inhabitants and caregivers.
If detect cigar smoke	Unsafe inhabitant behavior	Tell the inhabitant: "DO NOT SMOKE INSIDE THE HOUSE"; possibly notify caregiver.

When the rules are parsed, the system evaluates the parameters. For example we have defined four basic Jess files (see **[9,10]):** “burning.clp”, “ignition.clp”, “intoxication.clp” and “cigar.clp”; each one of them with different parameters. Based on the actual values the decision module sends to the Actuator Controller a set of commands as described in table of risks. The general idea is that new rules be constantly incorporated into the system following some learning pattern.

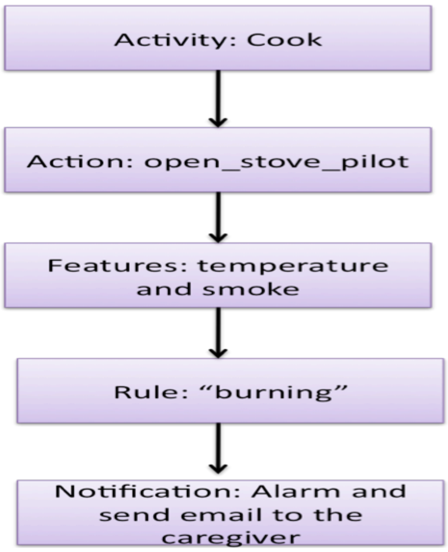

Figure 4. A scenario in which the inhabitant has forgotten that he had started cooking a meal.

2.3 Taking Action

The actuation module provides an easy way to use the different actuators (e.g. alarm, water sprinkler, fan, lamp). In order to use commonly available infrastructure, we use a “FireCracker” to generate signals received by X10-controlled electrical appliances, thus allowing existing electrical wiring. By programming the various ports and devices a simple actuator interface can be readily implemented. Added to this we have three cameras on he lab that take shots of the situation and using the Java Mail services a notification is sent to the caregiver and other entities.

3 Implementation

We present here the initial development that implements the five tier architecture of the SDBI. **Figures 5 & 6** present screen shots of the GUI. The system is fully functional and various scenarios have been tested. An examples of the notification sent is given in **Figure 7**. We have use the MVC (Model View Controller) programming architecture using Java extends and implements clauses extensively, in this way SDBI is a good for teaching object oriented programming. For the case of the Ontology this was develop in Protégé and the rules in Jess and they are given to the system as an external Jar file, and the contents of them can be read as text files.

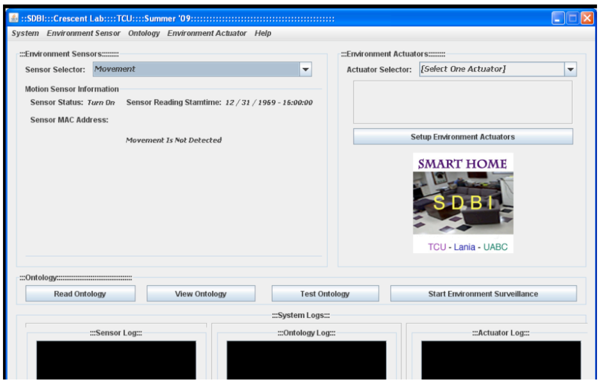

Figure 5. SDBI General GUI showing Sensors

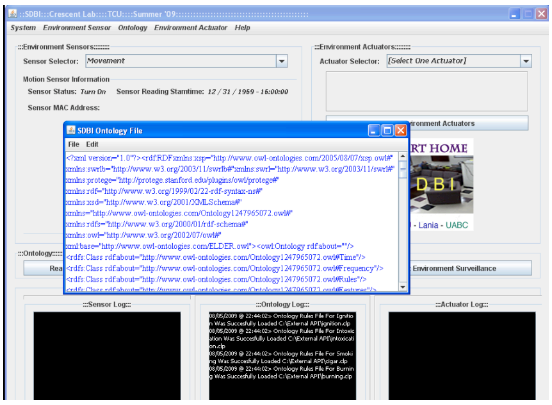

Figure 6. SDBI: Ontology Display

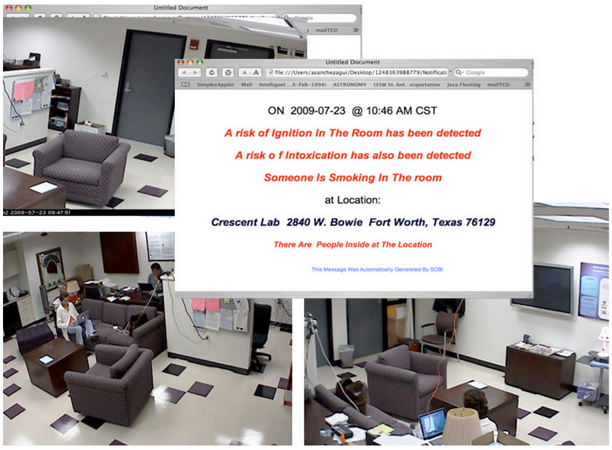

Figure 7. Email message sent with pictures

4 Pedagogical Results

We are highly satisfied with bringing visiting graduate students into our undergraduate research environment. A benefit of this increased international research collaboration among faculty and students opens new ways to achieve our academic goals, specifically in Computer Science. We see the field as increasingly about how we use massive amounts of data, information, and knowledge than about any particular technology. We strive to help our students learn how to solve real information problems. The development of ontology-based information services requires the collaboration of many points of view and this international component has proven beneficial. Furthermore if we want to increase our student population, our view of computer science must have new views, as suggested by Denning **[11].**

Believers in our University's strong commitment to undergraduate research, active learning, and individualized mentoring, we have found that smart home research has been a rewarding experience for faculty and students (and the University seems to like the marketing aspects as well). Many students that have worked in our lab go directly into respected graduate schools (e.g., CMU, University of Colorado), while others find that their experiences are directly transferrable to the work environment. Our industry advisory board commonly bemoans the lack of "soft skills (communication, managing stress under deadlines, working in a team with diverse backgrounds, personalities, and experiences) that they see in many new graduates. By giving our undergraduate students an immersive experience, performing "real" research tasks alongside faculty and visiting graduate students, they graduate with many of the skills they need to take their place as professionals.

5 Conclusions and Future Work

"Arriving at one goal is the starting point to another"
John Dewey (1859 – 1952)

As teaching facility SDBI complies with the desired goals, yet this is an initial prototype. The system has some limitations that must be addressed in the near future. We mention here some of the enhancement we will add to it. At the sensor level more sensor must be tested and connected to the SPOTS. Concerning the SPOTS significant testing on the communication protocol must be tested to optimize their performance. As a product, SPOTS are a new technology and we expect new developments soon. Concerning the Ontology, a more detailed set of rules must be appended to the system to considered more detailed cases. For the case of the X 10 protocol, it is well known technology readily available and if proper appliances are used the system can do the required jobs. Although the FireCrackers over the serial port are now rare, USB interfaces are in the market. Finally we barely touched the issues of send mail services and we can think about using SMS for mobile phones both for voice and text messaging.

Acknowledgements

This work was sponsored in part by a VIA grant from TCU and funds from the graduate students' home institutions. We wish to acknowledge the contributions of the students and the following Mexican universities: Universidad de Guadalajara, Lania, Universidad Autonoma de Baja California, Itesm-Monterrey.

References

1. Burnell Ball, L. Sanchez, A. Priest, J. and Hannon, C."The Crescent Lab: A smart home lab for students" in *ENC'06 Seventh Mexican International Conference on Computer Science*, SLP, Mexico, September 2006.

2. Consolvo, S. Roessler, P. Brett E. Shelton, B. LaMarca A.and Schilit,B. "Technology for Care Networks of Elders" in *PERVASIVE Computing*, *IEEE CS* and *IEEE ComSoc*, 1536-1268/04.

3. Sanchez, A. Hannon, C. Juan P. García-Vázquez, JP. Garcia, C. Ceballos, H. and Cetina, O. "Service Robots on a Smart Home Lab" in *Workshop on ServiceRobots of the 7th MICAI,* Mexico, November 2008.

4. Hannon, C. Burnell Ball, Rinewalt, R. and Sanchez, A. "The Crescent Lab experience for international graduate students" in *PIC ENC 2007,* Morelia, Mexico, September 2007.

5. Hope Microelectronics Co., *Futurlec* HUMTEMPSENS - *LPGSENSOR* - *LPG Sensor MQ135* - *Air Quality Control Sensor*.. Retrieved June 2009, from http://www.futurlec.com

6. SUN Labs. *SunSPOTWorld*. Retrieved June 2009, from https://www.sunspotworld.com/

7. Tercero R. and Saldaña-Jimenez, D. S. *Selecting Digitally Basic Intentions,* Crescent Lab Technical Report, TCU, Summer 2009.

8. Saldaña-Jimenez D. Rodríguez, M. D. Espinoza, A.N, García–Vázquez, J.P, "A Context-Aware Component for Identifying Risks Associated to Elder's Activities of Daily Living", In *IEEE Proceedings PERVASENSE '09*, London, UK

9. Jess: a rule engine rule for the Java platform (n.d.). Retrieved June 2009 from http://www.jessrules.com/.

10. Web Ontology Language Overview. Retrieved June 2009 from http://www.w3.org/TR/owl-features/.

11. Denning, P.J.*"*Visions of Computing*", CACM*, August 2008, 51(8), pp. 19-21.