

The Design and Implementation of an Energy-Smart Home in Korea

Jin Xiao*

Division of IT Convergence Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Korea
jinxiao@postech.ac.kr

Raouf Boutaba

Division of IT Convergence Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Korea
School of Computer Science, University of Waterloo, Waterloo, ON, Canada
rboutaba@uwaterloo.ca

Abstract

We present the motivation, design and implementation of a smart home system in Korea. Our system is open, extensible, integrated, intelligent, and usage-centric. We detail the challenges and key design requirements for the smart home system based on our past experiences, and show how convergence system design is a capable methodology for enabling an integrated and multi-faceted home management system that encompasses energy management, home appliance control, environment management, u-health, and living support functionalities under a single unified design. Using energy management as a specific case study, we demonstrate how convergence system design can encapsulate technology heterogeneity and hardware-software disparity without compromising simple yet powerful user interfaces.

Category: Convergence computing

Keywords: Smart home; Energy management; u-Health

I. INTRODUCTION

A. Overview of POSTECH Smart Home

With the accelerated introduction of powerful electronics, computing devices, and communication infrastructures to homes today, the concept of homes is involving. Home electronics are acquiring advanced communication and computing capabilities, and various new environmental, medical, and energy sensors are making their way into home deployment. Virtually all modern home devices can communicate and exchange information.

At the Division of IT Convergence Engineering of Pohang University of Science and Technology (POSTECH), we have designed and constructed a modern home equipped with diverse sensors, modern appliances, and home area networking infrastructures (power line communication, Ethernet, Wi-Fi, and ZigBee). Fig. 1 shows an actual view of the smart home. We have designed and are implementing a POSTECH Smart Home system that challenges this research frontier, and seek to obtain answers as to what functions are needed and how to realize them. The three areas we are examining in the POSTECH Smart Home are energy management, u-health, and envi-

Open Access <http://dx.doi.org/10.5626/JCSE.2013.7.3.204>

<http://jcse.kiise.org>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 12 June 2013, Accepted 4 July 2013

*Corresponding Author



Fig. 1. Pohang University of Science and Technology (POSTECH) Smart Home deployment.

ronment control and protection.

Previous work on smart home system design has generally focused on a specific problem area such as information correlation [1] or hardware (e.g., [2]), and therefore did not take on a holistic system engineering view. In this paper, we present the POSTECH Smart Home architecture, and discuss key design characteristics, user-friendly user-interfacing (UI) design, and non-intrusive network architecture. We also detail the design and implementation of an energy management solution currently being deployed to better illustrate the challenges that must be addressed in smart home research.

We found that there are four major key design requirements that every smart home of the future should meet:

- 1) **User-friendliness:** This is an often overlooked design criterion in many areas of system engineering. The user-friendliness we are interested in extends beyond UI issues, to how we can develop functionality that is most comfortable and helpful to typical, often non-technical, home occupants. In this regard, we find that timely and focused functionality combined with an easy-to-use interface achieves the best reception among households. However, it is not easy to obtain timely and focused system design, as it relies on a rather high degree of intelligent automation.
- 2) **Intelligence:** Intelligent automation is what general households expect from future smart homes. However, how to be intelligent in the correct way is a key issue for systems and application developers. In the context of the home, we find that intelligence gains the most appreciation when automation is provided for the most basic and sensible functions (such as turning on lights when coming home, and turning them off when going to sleep). The key enabler of such intelligent automation is not only algorithmic,

but also extremely knowledge-centric. The mere generation of user context or intent prediction (e.g., “I want to go to sleep”) requires complex information processing of diverse information sources.

- 3) **Non-intrusiveness:** Another aspect of user friendliness and intelligence comes from the ability of the technology to operate in the background. House occupants should not be constantly reminded of the need to command or interact with machines, and should not be bothered by streams of notifications and queries for the most basic and routine functions. Sometimes, it is better not to automate, and even to make imprecise decisions (for very simple, basic tasks), than to bother users for precise instructions.
- 4) **Security:** Security and its accompanying factor, privacy, are extremely important for the adoption of any smart home system.

II. SMART HOME SYSTEM ARCHITECTURE

In this section, we present the smart home system architecture (Fig. 2) that satisfies the key design factors outlined in Section I. The entire system is roughly composed of five substrates: device adaptation, knowledge base, decision and policy engine, service managers, and UI device integration. The entire system implementation is supported by Java Spring Framework [3], a lightweight open-source application framework for enterprise development, based on code published in ‘Expert One-on-One J2EE Design and Development’ by Rod Johnson. The Spring Framework is capable of integrating with various third-party frameworks and libraries. It comprises a data access framework, remote access framework, aspect-oriented programming framework, model-view-controller framework, and batch framework.

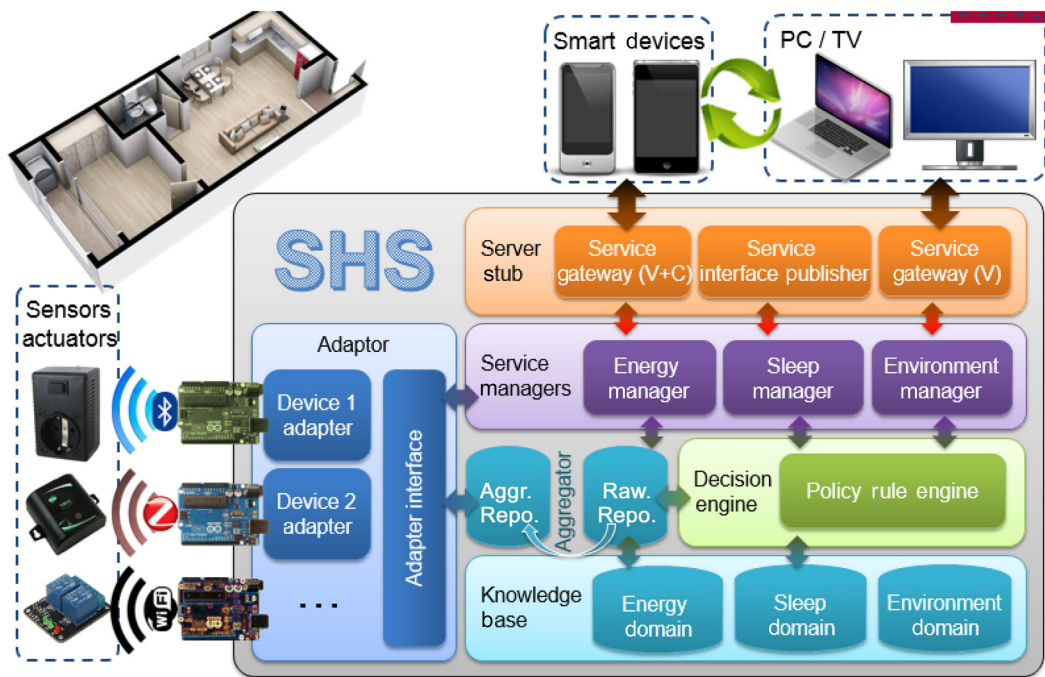


Fig. 2. Smart home system (SHS) architecture.

A. Device Adaptors

The device adaptors are the essential gateway between the smart home system and the hardware sensor devices. As such, they are responsible for providing technology-neutral interfacing between the devices (sensors and actuators) and the software system. This is an essential aspect of the adaptor design, because of the vast heterogeneity that exists among the different sensor devices, which include different communication protocols, message formats, data representations, control and query semantics, etc. The adaptors are device-class specific, meaning that for a specific product, a specific software adaptor needs to be developed. For our smart home, we have developed an adaptor for two different types of ZigBee protocols, for Samsung appliances, and for Bluetooth-enabled environmental sensing and control devices. Fig. 2 showcases some of the communication technologies we needed to adapt to, including Bluetooth, ZigBee, and Wi-Fi.

B. Knowledge Base

The successful adaptation and intelligent automation of a smart home relies on the ability of the smart home system to organize, process, and analyze different sources of information to drive the automation decision-making and user context determination. To this end, a strong and formal support for the knowledge base is central to the system design. The knowledge base is a vital design challenge for our smart home system. We have constructed formal information models that specify the relationship,

format, semantics, and context of each information source (e.g., environmental information, energy information, appliance usage data, appliance control information). We have also created a relational database structure that supports publish/subscribe semantics, meaning that software services or devices can subscribe to specific knowledge events in the database (e.g., a particular metric value has been changed/updated, a particular user context has been fulfilled). Three layers of information are organized in the database:

- 1) **Sensor information:** This is the basic sensor information gathered from specific sensor devices. This sensor information is not distilled and unrelated to each other. For processing efficiency, we also created aggregate classes of the base metrics such that aggregate values are generated (e.g., average, max., min., etc.) across a larger time interval than the sensor's sampling intervals.
- 2) **Context information:** The context information is generated by correlating multiple sensor information metrics under a specific criterion. Context information is model-driven by nature, in that to obtain specific context information (e.g., that the user is sleeping), we derive specific computational models that relate multiple pieces of sensor information together in a mathematically structured process.
- 3) **Rule-driven intelligence:** The intelligence of the system comes from the generation of rules or policies that specify the condition-action pairs among sensor information, context information, and control actions.

C. Decision and Policy Engine

The decision and policy engine in our smart home system is responsible for the intelligent automation and home-directed management. This functionality is supported by an event-condition-action (ECA) rule engine:

- **Event:** specifies the signal that triggers the invocation of the rule.
- **Condition:** a logical test that, if satisfied or evaluated to be true, causes the action to be carried out.
- **Action:** consists of database object manipulation, management actions, and device information and command exchanges.

Fig. 3 shows the rule engine’s process flow. Once an event occurs, a condition evaluator will perform a logical test on the event using defined policy rules. The policy rules are defined by homes or are built into the system using an extensible markup language (XML)-based scheme and interpreted by the ECA rule parser. The matching ECA rule is then invoked by the condition evaluator. If the event satisfies the condition set, the corresponding actions are scheduled by the action scheduler. As a part of the action, information from the knowledge base may be accessed and updated.

The ECA rule engine is implemented using a Java-based rule engine called Drools [4]. Drools uses a forward chaining inference engine that relies on the Rete algorithm for pattern matching. There are five sub-modules in Drools, and our system uses Drools Expert module. Fig. 3 shows an example syntax of Drool rules.

D. Service Managers

The service managers are where the domain-specific management intelligence, analytics, and control logics reside. In the next section, we will show some design specifics for home energy management and much of the pro-

cesses, data views, and intelligent scheduling functions implemented as an energy manager. The service managers interact with multiple component groups in the smart home system, including exchange with adaptors through adaptor interfaces to retrieve, configure, and execute device commands; interaction with server stubs to drive the UI and push selective information; querying the knowledge base for specific metrics and rule sets, and updating them based on intelligent processing and algorithms; modifying rules through the decision engine.

E. Multi-Modal UI

The abundance of multiple digital displays and interactive media in today’s homes calls for multi-modal user interface design, to facilitate the comfortable and easy use of a smart home system. Our smart home service utilizes smartphones for information presentation and home management, as well as televisions for augmented display and presentation. Fig. 4 shows the technology implementation that realizes this multi-modal UI design. We utilize an Adobe technology platform consisting of a Flash Player and Adobe AIR. A flex framework is built on top of the Flash Player application programming interfaces, and runs both Flash and Adobe AIR. It allows for a unified UI design that can be supported across multiple display platforms such as TVs, desktop computers, and smartphones. In the backend, a BlazeDS server [5], which is a server-based Java remoting and Web messaging technology, allows developers to connect to back-end service logic and databases.

Fig. 5 shows some of the UI screens of our system. The top two sub-figures show the login screen for the TV and the smartphone, while the bottom two show the home screens of the TV and the smartphone after successful login. The multi-modal UI design allows for information augmentation and synchronization between the TV display and the smartphone, while the actual control is com-

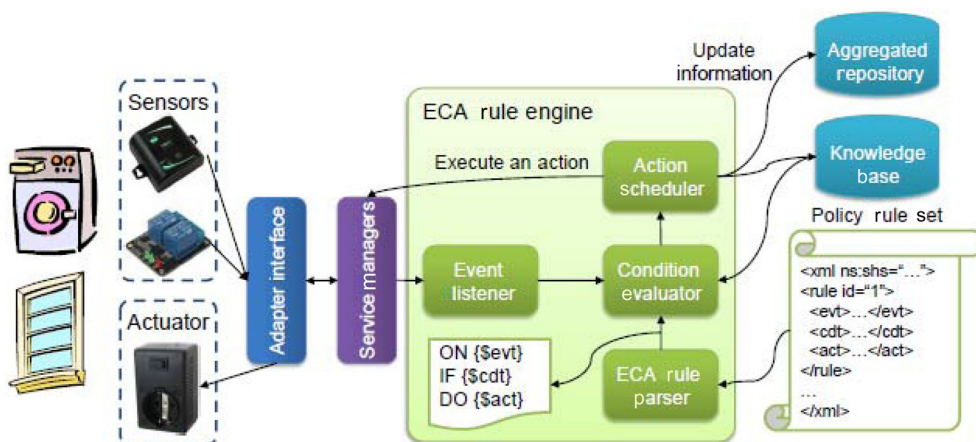


Fig. 3. Event-condition-action (ECA) rule engine.

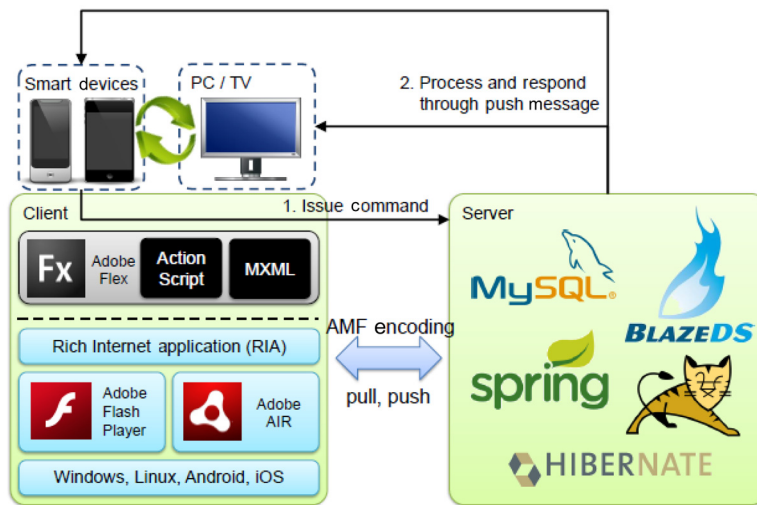


Fig. 4. Technologies supporting multi-modal user interface. MXML: magic extensible markup language.

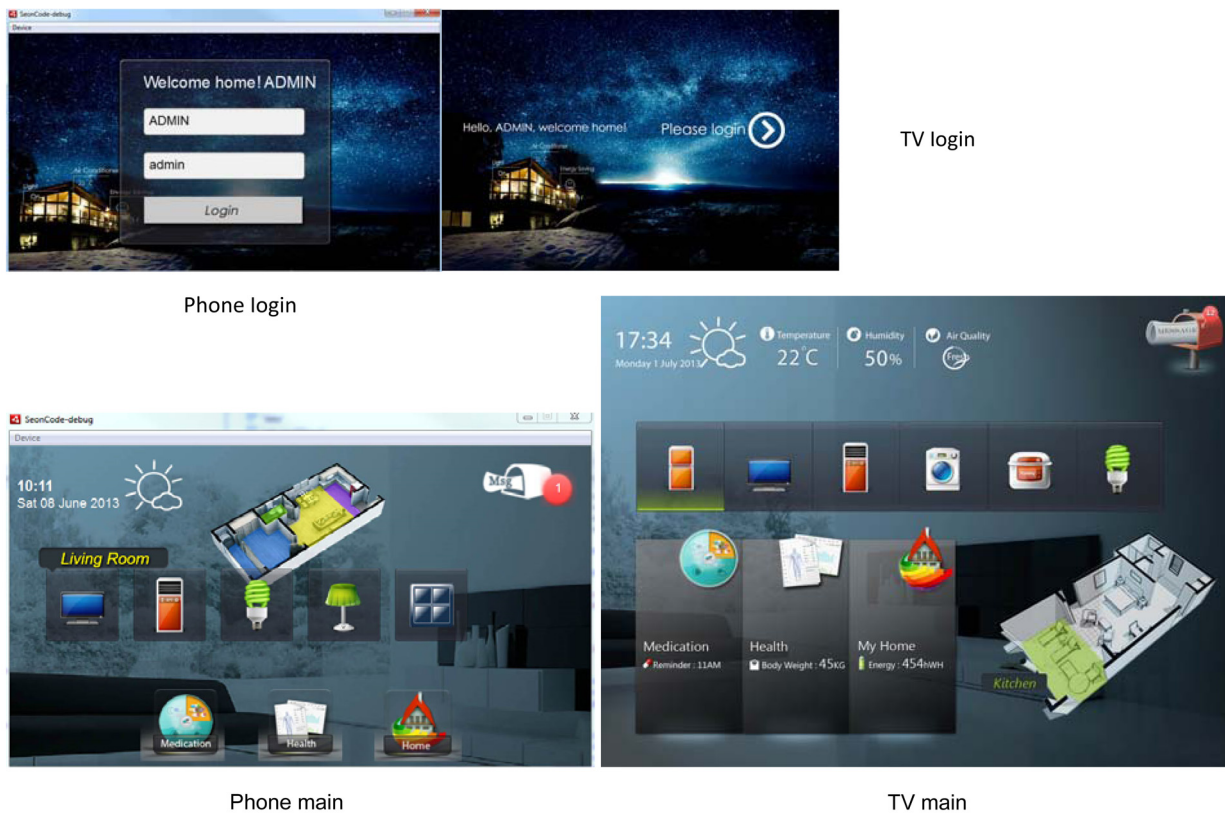


Fig. 5. Example user interfaces for smart home system.

pletely implemented through the smartphone, allowing the home user to control, view, and interact with their home anywhere within the home, or even outside. The home system currently also supports voice synthesis reporting of key messages and events, and in the future, will support voice commands. Security to system access

is reinforced through a TripleSafe design that employs user password login (Fig. 5), radio-frequency identification-based person tagging (at the home entrance), and smartphone-enabled facial recognition. We utilize all three safeguards to secure access to information, control, and device operations.

III. SMART ENERGY MANAGEMENT AT HOME

Fig. 5 shows the home screen of our home system, which contains the home layout and appliance icons. This is one of the essential UIs for information presentation and manipulation for home energy management. The user can freely select different rooms on the layout, and the controllable appliances in each room are automatically loaded and displayed. The user can then interact with each of the specific objects by interacting with their corresponding icons. In the backend, there are specific appliance object models that abstract the metering, control, and management capabilities of each appliance in the knowledge base. It is through these appliance object models that the energy manager (an instance of the service manager) can operate. The adaptors of the system actually interact with the hardware device, which consists of a suite of energy meters with built-in electric relays,

communicating via ZigBee network through a ZigBee coordinator and ZigBee sensor modules (Fig. 6).

The energy management is also responsible for providing historical information on the various aspects of energy consumption and cost. Two primary information presentations are currently supported: a graph view and an appliance schedule (Fig. 7). The graph view supports dynamic ranging of the data reporting granularity (e.g., hour, day, week, etc.), as well as smooth sliding of the reporting period (e.g., current week, past week, etc.). Dynamic animations of the value bars provide convenient visual cues to the user about the changes in energy consumption, selected appliance usage history, and energy cost. The appliance schedule is a calendar-style appliance usage chart that depicts the appliance usage data, and allows for user-directed appliance scheduling. Through simple, one-touch editing on smartphones, the users can modify appliance schedules, and the system will automatically turn appliances on or off according to these specific schedules. In addition, aggregate appliance control profiles are created for homes for even more dynamic and easier control over groups of appliances. Some example profiles are sleep mode, energy saving mode, study mode, and away mode.

One aspect of energy management where intelligent automation can help greatly is the scheduling of appliances. Although with the capability to create appliance schedule profiles, the home's task of appliance management is simplified, this capacity still does not provide users with scheduling intelligence where fine-grained control and adjustment are necessary (e.g., re-adjusting appliance usage so as not to exceed hourly or daily energy quotas, to minimize peak-time energy usage due to higher prices, etc.). Accordingly, our energy management implements a suite of smart appliance and environmental control algorithms called CODREX [6]. CODREX provides a powerful set of algorithms that allows for near-optimal appliance scheduling under a diverse set of objectives: user

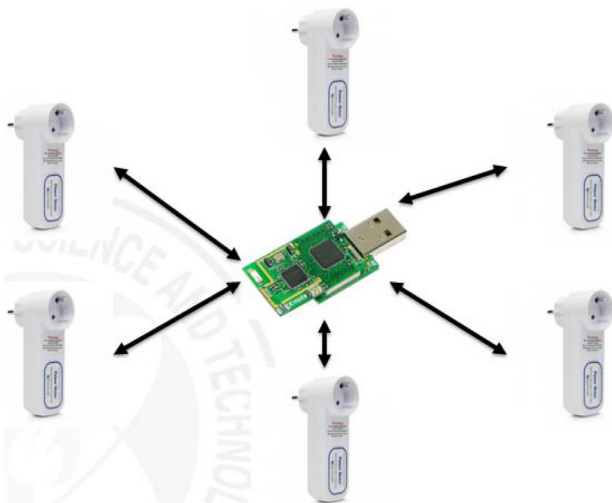


Fig. 6. Energy meter and relay deployment (ZigBee network).



Fig. 7. (a) Energy chart view and (b) energy schedule view.

comfort constraints, energy pricing, optimization objectives (energy and price). It also automatically adjusts home heating, ventilation, and cooling operations based on the same constraint sets.

IV. CONCLUSION

We have presented the design, architecture, and implementation of a smart home system for future home automation and comfortable living assistance. In particular, we have focused on a set of design criteria, and discussed how our system design reflects them. As the system design space is rather large, we focused on showcasing some of the energy management aspects. The first round of our smart home system development is near completion, and will be tested in real home usage scenarios in our POSTECH test home in the coming months.

REFERENCES

1. J. Y. Son, J. H. Park, K. D. Moon, and Y. H. Lee, "Resource-aware smart home management system by constructing resource relation graph," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1112-1119, 2011.
2. D. M. Han and J. H. Lim, "Smart home energy management system using IEEE 802.15.4 and ZigBee," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1403-1410, 2010.
3. Spring framework, <http://www.springframework.org/>.
4. Drools: the business logic integration platform, <http://www.jboss.org/drools/>.
5. BlazeDS: the server-based Java remoting and Web messaging framework, <http://sourceforge.net/adobe/blazeds/wiki/Home/>.
6. J. Xiao, J. Li, R. Boutaba, and J. W. Hong, "Comfort-aware home energy management under market-based demand-response," in *Proceedings of the 8th International Conference on Network and Service Management*, Las Vegas, NV, 2012, pp. 10-18.



Jin Xiao

Jin Xiao received the Ph.D. degree in computer science from the University of Waterloo, Canada, in 2010. He is currently a Research Professor at Pohang University of Science and Technology (POSTECH), Korea. His research interests include network and service management, service automation, network economics, smart grid, smart home and energy management of smartphones. He has received several best paper and poster awards such as the Fred W. Ellersick Prize in 2008.



Raouf Boutaba

Raouf Boutaba received the M.Sc. and Ph.D. degrees in computer science from the University Pierre & Marie Curie, Paris, in 1990 and 1994, respectively. He is currently a Professor of computer science at University of Waterloo, Canada and a Distinguished Visiting Professor at the Pohang University of Science and Technology (POSTECH), Korea. His research interests include control and management of networks and distributed systems. He has received several best paper awards and other recognitions such as the Premier's Research Excellence Award, the IEEE Hal Sobol Award in 2007, the Fred W. Ellersick Prize in 2008, the Joe LoCicero Award and the Dan Stokesbury Award in 2009. He is a fellow of the IEEE.