

Towards WoT-based Smart Home Middleware Platform

Jeong-Sik Kim¹, Eung-Sung Kim² and Jin-Hong Kim³

^{1,2}*Department of Computer & Mobile Convergence, Gyeonggi College of Science and Technology*

³*Department of Information and Communication Engineering, Seoil University*
{¹arius70, ²imagecap}@gtec.ac.kr, ³jinhkm@seoil.ac.kr

Abstract

Today, Sensors/wireless sensor networks are being deployed around the world, measuring local and global environmental conditions. Their advanced sensing functionalities enable context-aware smart platforms, middleware and applications to proliferate. Residences are transformed into smart home, incorporating embedded sensors and pervasive technology. In an idealized vision of a fully integrated smart home, all the operations of a house can be efficiently controlled by a unified smart ubiquitous application. However, we are far from realizing this scenario. A main barrier is the proliferation of incompatible standards and protocols used by various device manufacturers, which makes the smooth integration of appliances from different vendors a complex process. In this paper, we exploit these new technological possibilities to bring smart home towards the Web, achieving high interoperability and flexibility. By employing standardized, well-known, reliable and scalable Middleware Platform, between the embedded devices of the smart home can be addressed. We present the development of a smart home service, supporting concurrent interaction.

Keywords: *Web of Things, Smart Platform, Smart Home Service, Middleware Platform*

1. Introduction

Embedded physical devices, such as household appliances are becoming smarter and smarter. They are equipped with embedded microprocessors and wireless transceivers, offering limited communication capabilities and providing smart behavior [1][2][3]. Everyday objects are fitted with small, cheap mobile processors, sensors and actuators. Sensors and wireless sensor networks are being deployed in smart home solutions, measuring with precision the environmental conditions inside the home environment. Their advanced sensing functionalities and their increasing accuracy enable the development of smart home applications that offer advanced automation [4]. Residences are transformed into smart home, incorporating embedded sensors and actuators, and pervasive technology. This merging of computing with physical objects introduces the concept of information appliances, defined as devices or machines, designed to perform some specific functionality but are usable, at the same time, for the purposes of computing [5][6][7]. Typical examples of computing devices include smart phones, embedded sensors and actuators, radio-frequency identification (RFID) chips and smart cards. In the near future, home will offer new automation possibilities to their residents, increasing their comfort level. Changes will happen to the way people live and interact with their home environments, when technology recedes into the background of their lives [8], when information processing is thoroughly integrated into everyday objects and activities. By now, residential smart meters have been introduced in our lives as sensor devices

Received (October 28, 2017), Review Result (December 7, 2017), Accepted (December 11, 2017)

that measure in small time intervals the energy consumption of a house [9][10]. Accordingly, this paper proposes Web of Things based Smart Home Service (SHS) for Middleware Platform.

1.1. Problem Statement and Motivation

In an idealized vision of a fully integrated smart home, all the operations of the house can be efficiently controlled by a smart ubiquitous application. However, we are far from realizing this scenario [11]. A main barrier is the proliferation of incompatible standards and protocols used by various device manufacturers, which makes the smooth integration of appliances from different vendors a very complex process. Embedded technology, based on multi-hop wireless communications, is being integrated lately in smart environments such as smart home [12][13]. However, even though this technology is becoming mature and highly reliable, it still faces issues such as device failures, transmission losses and unexpected delays. Home residents and family members have advanced needs, demanding direct access to their home devices easily and fast. These people may need to interact simultaneously with their smart home environment. Moreover, smart home applications that target home automation and energy conservation cannot tolerate faults and errors. Hence, application frameworks for smart home need to compensate problems at the physical, ad hoc, resource-constrained environment and also to support multiple users who may interact concurrently with their smart home. Furthermore, application development from third parties and end users for automating home environments is currently a complicated procedure, since the user needs to study and understand the vendor-specific proprietary protocols and technologies employed in each smart home solution [14][15]. Web-enabled smart home have a crucial role to play in current and future technological and societal challenges. This important role of smart home in relation to the physical and urban environment needs to be investigated, highlighted and promoted. Technologies, protocols and architectures need to be defined, which permit the smooth and flexible adaptation of smart home solutions to these challenges, ensuring satisfactory levels of security and user privacy.

1.2. Approach

Smart home operating with Web technologies could be adapted with little effort, to support interaction with social networking sites, for sharing the status of the house as well as individual home devices/services, between family members and other tenants. Social networking could be also utilized for establishing energy competitions between neighboring houses and flats [16]. In addition, it could be used for electricity-related comparisons, to help people understand their electricity device. In addition, specified service by all smart home, facilitating in-home and home to communication. Also in this case, the Web, as a scalable, pervasive and flexible platform, could be an appropriate solution for such a wide-scale interconnection. Eventually, in order to enable the vision of interoperable smart home operating with Web technologies, reliability and performance need to be ensured. Especially when considering specific smart home applications dealing with safety, health, advanced automation.

2. Related Works

2.1. PMPT (Pervasive Middleware Platform Technology)

Home appliances are no longer isolated devices. They are interacting and communicating with each other at home. This brings many challenges to integrate various devices and to make them smarter for the users. This kind of smart home can help disabled or elderly people to live life with more independence and comfort. Home automation and assistive living are possibly the eminent cases of context-aware pervasive

systems [17]. A home with these types of appliances and smart intelligence to combine the devices' services in smart ways is either called a smart home or house with home automation. Home automation can comprise centralized control of lighting, heating, ventilation, and air conditioning, home security, and other systems. The typical purpose of these systems is to deliver better convenience, ease, energy efficiency, and security. Smart home systems provide automation capabilities that allow home owners to have more complete control over their home. These systems integrate many devices having different capabilities, such as TV, doors, lights, oven, windows, and multimedia players. Smart home systems also include sensors and actuators for different complex systems, such as intrusion detection, video surveillance, fire detection, patient health monitoring, and entertainment. These sensors and actuators use different communication protocols that are mostly incompatible with each other [18]. Usually, current home are equipped with smart devices with wired and wireless technologies. In our reviewing of the literature, we found a number of networking protocols to interconnect devices in a smart home. These include Java intelligent network infrastructure (Jini), UPnP, and Device Profile Web Services (DPWS). Technologies based on service orientation are not new for pervasive systems. UPnP and Jini have been proposed as protocols and architectures for dynamic device and functionality discovery, based on describing services in terms of WSDL and Java interfaces respectively. These are excellent starting points for our study, as they provide support for basic interoperation, but to realize genuinely smart home, more aspects need to be designed in terms of home sensing and composition. There is some standard protocols for exchanging the information between the devices and the components of a middleware platform, such as Simple Object Access Protocol (SOAP), REpresentational State Transfer (REST) and Web Services Description Language (WSDL). We will present the most prominent protocols and transmission styles that are important for home networking and service application [19][20].

2.2. OSGi

OSGi is an open specification for the delivery of managed services to networked environments, such as home and automobiles. The OSGi service platform consists of the OSGi framework and a set of standard service definitions. The OSGi framework is the execution environment for services, which is based on Java Virtual Machine. It can accommodate many heterogeneous frameworks in a generic environment. The OSGi framework is able to integrate heterogeneous devices and services, which appear as local entities. An OSGi-based smart home architecture is presented by which uses the OSGi gateway as a central coordination point for managing the home network and covers numerous heterogeneous communication technologies. Smart home usually require the integration of many heterogeneous devices and service applications in deployment so it is hard and expensive to implement system tests. An OSGi based context-aware framework is developed to provide dynamically appropriate information and services for users in a smart home environment. The developed middleware integrates services through a reasoning system [21]. Further, OSGi-based Home system is presented in, which provides integrated user interface and an event-driven, multi-threaded service development platform thorough adopting message oriented middleware and OSGi technologies. It offers reliable automatic operations, fault tolerant and configurable home automation, high extensibility and large scalability. The OSGi platform offers the means to manage all services and computation tasks at the service gateway in a centralized manner, and is therefore used as a fundamental component at the pervasive layer in the architecture we propose herein. We propose to describe all devices as UPnP devices, a protocol that is easily integrated as an OSGi bundle into the common registry. Several approaches propose extending the OSGi framework so as to integrate devices that use alternative protocols, so that they can communicate to other devices.

2.3. Middleware Layer

The advancement of home network technologies provides new opportunities to the user to control and monitor devices in home such as lights, doors, TVs, air conditioners, and entertainment systems. However, these opportunities have major issues that are preventing to extensive use of these appliances embedded into smart spaces [22][23]. The devices have different communication interfaces and manufacturers, restricted to a specific service and technology, and also the home network application is unable to avoid the problem of dynamically integration of devices into the system because it often happens that devices are dynamically added/removed. It means that available services of devices in the smart home are dynamic in their availability [24]. There are several kind of devices at physical layer, such as sensors and actuators. The devices sense the environment and execute some actions to satisfy the user instead of having an explicit request from a user. To have smarter solutions for the home network, we need to have communication among the multiple networked devices to support complex services for the users [25][26]. The home network needs a middleware that connects different appliances and handles communication among them. Middleware is a software layer between the application and hardware layer, supporting integration among numerous products and platforms, while preserving the integrity of the overall solution in terms of robustness and reliability [27].

3. Proposed System

3.1. Middleware Layer Specification for SHS

Middle layer application is primarily designed for an ordinary PC. The actual hardware, where middle layer application runs, depends on particular system architecture. It runs at all the times and hosts both bottom layer and middle layer software. This is a robust low-maintenance machine typically placed in a utility room. We can take advantage of operating system that allows multiple partitions on one CPU. Middle layer can be accessed via remote interface from any computer connected to the network / Internet. However, most of the current systems do not require a computer based machine running at all the times. Instead they use PC for configuration only and the system behavior is then downloaded into specialized hardware. In the latter case, middle layer can be used just as the bottom layer software for configuration only. This solution is generally more stable, but has limited functionality. The middleware layer represents the device layer and software layer architecture and the software components needed to abstract it. It is responsible for letting physical devices interact with the upper layer components by publishing available service descriptors, executing service instances, decoupling interactions from the specific communication mediums, *etc.* The pervasive layer is a service-oriented event-driven software layer. Device and service discovery, as well as interoperability, are important requirements for pervasive home networking. This component realizes an abstraction layer wrapping all of the devices currently in the house. This network has been designed in to adapt the complex home network into a transparent one. The substantial advantages of this abstraction cope with the unreliability of networks, as devices are added and removed randomly; the problem of application communication, due to the heterogeneity, and the lack of scalability. The pervasive layer aims to interconnect devices in heterogeneous networks in the home environment and to provide a common mechanism for accessing the services they offer for the rest of the middleware layers and applications. Figure 1 provides an overview of the architecture of the pervasive layer.

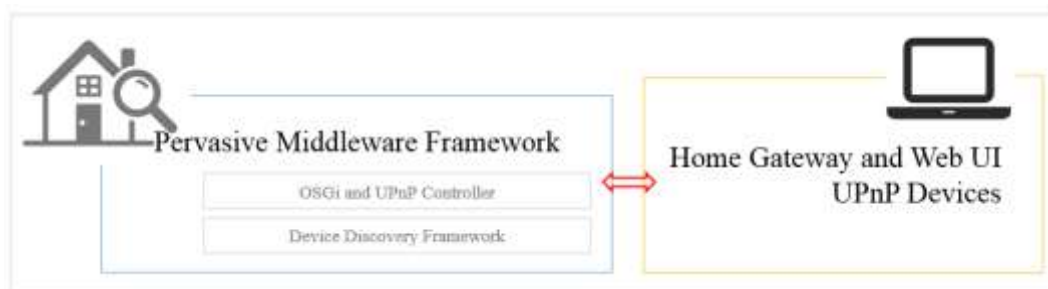


Figure 1. Middleware Framework Network

OSGi provides the framework for the wrapping of devices (UPnP or non-UPnP through the use of a proxy), providing a standard interface for interacting with them. All components participating in the OSGi framework are deployed as so-called bundles. The controller is a special OSGi bundle that is responsible for handling events and controlling the services available in the framework, functioning as a bridge between the OSGi layer and the Web Service (WS) layer, which provides a standardized API to the upper layers. On top are the clients, which can invoke services that are exposed by the available devices. Thanks to an event propagation mechanism, registered clients are notified when UPnP services become available or unavailable, and can subscribe to change events concerning the state variables that they are interested in. A client may be a visualization and simulation tool, a BPEL orchestration engine a composition layer, or various user targeted interfaces. Figure2 provides an overview of the architecture of the pervasive layer. The network layer, where physical devices can dock, is at the bottom. UPnP devices use TCP/IP and UDP as basic networking protocols.

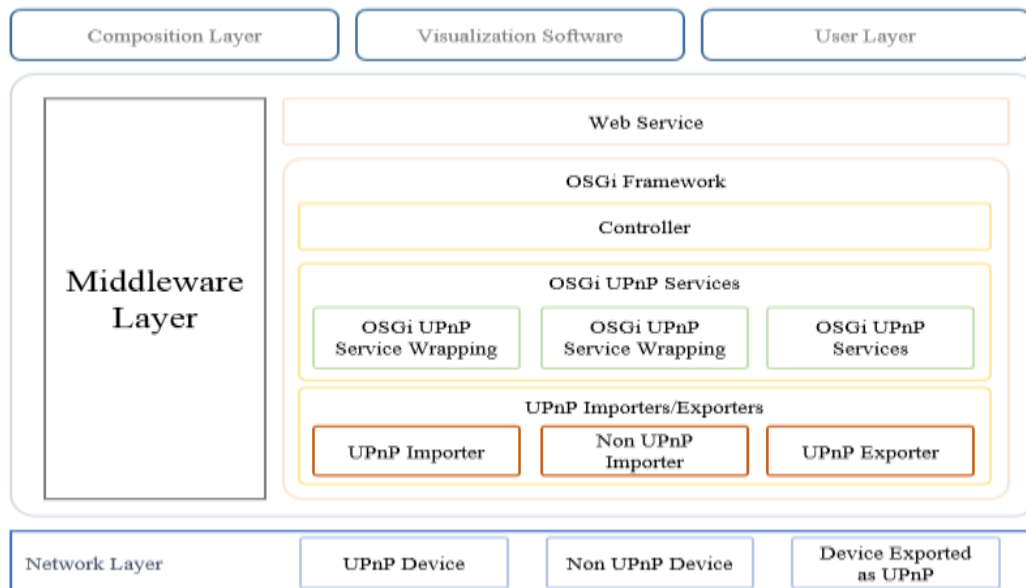


Figure 2. Middleware Layer Architecture

3.2. Requirements for SHS

SHS have a generations of embedded devices will be much more ubiquitous, highly integrated in our everyday lives, massively deployed in home environments. This would create the need for a greater degree of flexibility in order to be manageable. Our of the envision three distinct interaction possibilities with home

devices that cover the most important use cases that should be supported in smart home: 1) Ad hoc interaction. Users directly access sensors and actuators on demand, 2) Continuous monitoring. Devices stream data at regular intervals, 3) Event-based messaging. Events are sent sporadically when something important happens. Devices must send data only if the phenomenon occurs and this as quickly as possible. Smart home shall be open and accessible for simultaneous users, who could pull easily the data they need directly through the Web, and use them right away in their own applications. We propose a list of requirements that most of the existing smart home application frameworks for embedded devices lack, and will be needed in large-scale, open, and heterogeneous smart home environments.

3.3. Application Framework Architecture

Application framework for smart home is described, based on the requirements defined. This framework achieves a Web-oriented interaction with the home environment, following the principles of REST and resource-oriented architectures. The reasoning for selecting these design guidelines is discussed that explain below the general design and implementation of the framework, assuming the presence of Web-enabled household appliances, and Web clients, who may interact simultaneously with their smart home environment. The application framework follows a modular architecture and it is composed of three principal layers: device layer is responsible for interaction and management of embedded devices; control layer is the central processing unit of the system; and presentation layer has the duty of presenting the available devices and their corresponding services to Web users, through their Web browser. In device layer, the Driver module holds the technology-specific drivers that are used to enable the interaction with embedded devices. As soon as this module discovers a new local device, a new thread is automatically created, dedicated to the task of representing the newly discovered device. Each thread keeps track of the static and dynamic properties of the device it represents, inside a Resource Registry. Upon reception of a response from a device through the driver module, this response is forwarded to the appropriate thread, in order to be further forwarded to the Web client, who created the respective request. Using threads for devices simplifies their management, increases their performance and keeps the implementation logic simple. Resource-constrained devices are generally unreliable, a Request Queue is integrated to each device thread. This queue supports multiple concurrent users, allowing simultaneous Web requests to target the same physical device. Every new request made for a service supported by the device, should be temporarily stored inside that queue. Requests are stored in a FIFO manner, and are transmitted sequentially. As soon as a response arrives in the queue for a previous-made request, a new request is returned from the queue and is transmitted for execution at the device. The framework is implemented in Java because of the versatility and portability of the language, which allows the framework to run on virtually any device that has a Java Virtual Machine (JVM). It can be considered lightweight as it needs only 2.7 megabytes for its full installation. The most common system architecture consists of an application framework covering a smart home or a flat. However, in larger-scale indoor scenarios such as a smart building, perhaps an application framework would be responsible for the whole building, in case it could reach effectively all the embedded devices of the building.

4. Implementation and Experiments

To examine the capabilities of the application framework in terms of supporting and managing numerous embedded home devices, we created an emulated scenario, in which multiple virtual devices are assumed to be available inside some smart home environment. Each device is discovered by the driver module of the

application framework, and a new thread dedicated to this newly-found device is created. We experimented with different numbers of devices, considering that the framework has been installed on a typical laptop. This is actually the computer type on which the framework runs, for the purpose of the experiments performed in the thesis. For better experimentation, we also installed the framework on a small server (Intel Core i5-3570K Quad Core 3.4 GHz), which is of course a more powerful computing machine. We considered 4-different performance metrics to assess the scalability of the framework in regard to supporting multiple devices as shown in abstract methods that need to be implemented by each home device type, in order to be support by the application framework.

```

public class NewDeviceType extends Driver{
// the initializations of NewDeviceType are performed here
public abstract void startDevice();
// receive a new message from NewDeviceType
public abstract void messageReceived
(String message) throws IOException {}
// send a message in DISCOVERY and DESCRIPTION mode
public abstract void sendMessage(
String id, char type, String content) throws IOException;
// send a service request in NORMAL OPERATION mode
public abstract void sendMessage
(String id, char type, Request request) throws IOException;
}

```

These metrics are: 1) *Device Discovery and Information Parsing Time*. This includes the time needed to discover the devices and parse their service description information. 2) *Main Page Response Time*. The time needed to load an XML representation of the smart home environment inside a Web browser, listing all the available home devices. 3) *Response Time of Devices Page*. The time needed to load XML-based service description information of some specific, randomly-chosen home device. 4) *Memory*. The memory allocated to the application framework during its operation, to support the threads of the physical devices that have been discovered. And also, we shown by Table 1.

Table 1. Support of Multiple Home Devices on a Laptop

Device No.	Disc. Time	Resp. Time Main Page	Resp. Time Device Info	Memory
10	0.63s	50ms	92ms	10.2KB
50	5.20s	726ms	1119ms	65.5KB
100	13.29s	924ms	2262ms	196.6KB
200	38.17s	3455ms	4981ms	1.7MB
300	58.80s	5183ms	7183ms	3.8MB

The results of our experiment may be observed in Table. 1 for the laptop case and in Table. 2 for the server case, considering a number of devices ranging from 10 to 300.

Table 2. Support of Multiple Home Devices on Small Server

Device No.	Disc. Time	Resp. Time Main Page	Resp. Time Device Info	Memory
10	0.81s	32ms	55ms	196.6KB
50	3.75s	487ms	761ms	262KB
100	9.11s	665ms	1739ms	458KB
200	14.82s	1659ms	2470ms	2.3MB
300	29.50s	3507ms	5730ms	5.1MB

Discovery and main page response times are displayed also in Figure 3. Obviously, the server machine operates much better than the laptop, giving much better discovery and response times. This improvement in performance is around 30-35% when devices are up to 100 and reaches 50% in some cases, e.g. when devices are equal to 200. The application framework, when installed on the laptop, may support up to 300 devices, with few seconds response times.

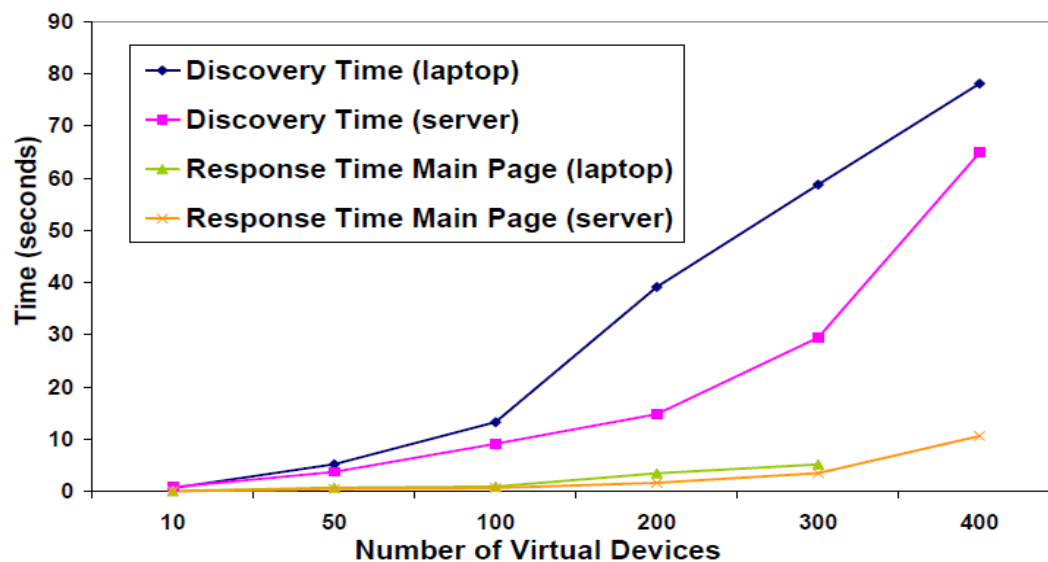


Figure 3. Discovery and Response Times as the Application Framework in the Presence of Multiple Home Devices

As the number of devices increases, the response times grow as well. As Figure 3 shows, the increase in discovery times is almost exponentially, both in the laptop and the server case, while response times grow linearly with an increasing gradient. This increase in response times is partly due to the thread environment, which has quite demanding computational needs in Java. By observing the response times, we consider that the framework installed on a laptop may effectively support around 100 devices, and installed on a small server around 200 devices. Larger numbers cause the system to malfunction. Memory requirements are increased as the number of devices grows, and a few megabytes are needed when 300-400 devices exist in the home environment. These memory requirements are rather small, considering the capabilities of computing devices nowadays. Finally, the time needed to discover the devices and parse their service description information is considered satisfactory, taking into account that around 1 minute time is enough to discover 300 devices in the laptop case and 400 devices in the server case. However, this time is only indicative, since other parameters would influence discovery time in a real-life

scenario, as for example the throughput and bandwidth capabilities of the device acting as the base station.

5. Physical Mashups for Smart Home in Case Study

The flexibility of the application framework, achieved by following REST principles, facilitates the development of smart applications, in the form of Web mashups. For example, with a few lines of code a database could be created and used, where all the data received from the devices would be automatically saved. Web mashups are Web resources that include content and application functionality through reuse and composition of existing resources. They can be parallelized with Business Process Execution Language, which is the formal language for service composition in WS-*. When Web mashups employ physical devices and services, they can be extended into physical mashups.

```
function getHttpRequest() {
  var xmlhttp = null;
  xmlhttp.open('GET', 'http://localhost:8080/tikitag/tags', true);

  xmlhttp.onreadystatechange = function() {
    if(xmlhttp.readyState == 4 && xmlhttp.status == 200) {
      var items = eval('(' + xmlhttp.responseText + ')');
      var secret_key = "04:BA:4A:B9:23:25:80";
      for (var i=0; i<items.length; i = i + 1) {
        if (items[i].t == secret_key)
          $('locked').innerHTML = "Door is unlocked!"; }
        if (!unlocked)
          $('locked').innerHTML = "Door still locked."; }
      }

    xmlhttp.send(null);
  }
}
```

Physical mashups exploit real-world Web services, offered by physical devices, combining them using the same tools and techniques of classic Web mashups. Our application framework adapts and applies the physical mashup practice in the smart home environment. Our framework supports the creation of physical mashups and advanced rules very simply, in any programming language that supports HTTP such as Perl, PHP, and JavaScript *etc.* They may be developed even by inexperienced home residents, who can reuse the functionality of their household information appliances in a uniform and easy way. Physical mashups may be combined easily with Web content, to offer a true Web experience to the real world. Web-enabled smart home have the capability to publicly expose the functionality of their sensor devices as openWeb API. Home owners could permit this, in order to help other citizens to get better informed about the environmental conditions that exist in the urban environment where they live. This environmental awareness would then assist people to have a better quality of life, by avoiding polluted or highly crowder or even dangerous areas. Thus, in order to adapt to these highly dynamic landscapes, urban mashups are proposed, defined as opportunistic physical mashups that are validated only when the local environmental conditions support the sensor-based Web services, which are declared by these mashups, and meaning that they focus on the services offered by sensor devices and not on the sensors themselves. This

happens because a large number of similar sensor devices may be deployed inside some area, offering the same service. To enable a true Web-based smart home application, the architecture of the application framework was extended to support a graphical application. The classical client-server model was followed, by placing the framework's functionality on the server side and the graphical application on the client side as shown in Figure 4.

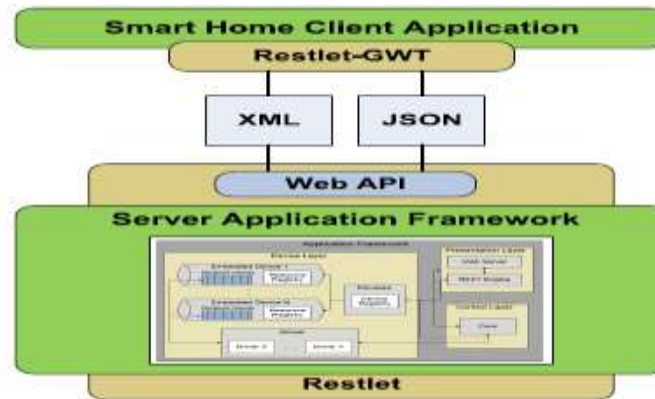


Figure 4. Web-based Smart Home Architecture in Case Study

6. Conclusions

In this paper, we investigated the general process needed in order to enable a smart home environment to the Web. This Web-enabling procedure was examined from various aspects, having in mind the relation of the smart home as an entity with important and influential actors of everyday life, such as online social networking. An important element in this paper was the development of a Web-based application framework for smart home, which can constitute a foundational pillar towards enabling a true Web enabled home environment. Merely, our proposal constitutes a flexible application-level solution for home automation, based on combining existing Web technologies and reliable, well-studied Web techniques. Our research work may be considered as a promising practice for home automation, supporting multiple simultaneous home residents. By following the physical mashup paradigm, our approach facilitates this procedure and it is characterized by high interoperability, supporting heterogeneous home devices and services. An important enhancement in the functionality of the application framework was the integration of request queues, enabling multiple concurrent users and efficient management of the communication with the physical devices of the smart home. Our technical evaluation efforts indicated that, by following the Web model, satisfactory performance of household embedded devices can be achieved, in terms of response times. Furthermore, various interesting Web applications, relevant to smart home were developed, in order to encounter technological and societal challenges such as energy awareness and conservation. These applications reused the functionality of Web-enabled smart home, offered as a Web API, with entities and resources available on the Internet/Web. Then, we explored how the Web-enabled smart home, as an important element in future digital cities and societies, may contribute for addressing various challenges. Some of these challenges include global discovery of environmental services in real-time through the Web and increased environmental awareness and knowledge inference in the urban area. Another aspect beyond the smart home research covers the extension of the application framework for smart home, as a powerful and flexible middleware for use in numerous pervasive scenarios. Definitely, a great challenge for future smart home is also security.

Security in smart home environments is a large topic and extended Web-based schemes need to be developed that involve end-to-end security mechanisms that guarantee authentication, data integrity and confidentiality, from the home devices to the application framework, and from the application framework to third party Web entities. We dedicated a small section discussing security inside the smart home and beyond, touching briefly only some general aspects.

Acknowledgments

The present research has been conducted by the Research Grant of Seoil University in 2017. Also, this research was financially supported by Gyeonggi College of Science and Technology.

References

- [1] J. Choi, D. Shin and D. Shin, "Research on Design and Implementaion of the artificial intelligence Agent for Smart home Based on support vector machine", Springer Berlin / Heidelberg, (2005).
- [2] I. Kastelan, M. Katona, G. Miljkovic, T. Maruna and M. Vucelja, "Cloud enhanced smart home technologies", Consumer Elec-tronics (ICCE), 2012 IEEE International Conference on, (2012).
- [3] C.-L. Wu, C.-F. Liao and L.-C. Fu, "Service-oriented smart home architecture based on OSGi and Mobile –Agent technology", IEEE Transactions on Systems, Man, and Cybernet-ics, Part C, vol.42, pp.193-205.
- [4] W. H. Tranter, "Wireless Personal Communications: Blu-etooth Tutorial and Other Technologies", Hingham, MA, USA: Kluwer Academic Publishers, (2000), p 262.
- [5] Y.C. Jung, Y.M. Park, H.J. Bae, B.S. Lee and J.S. Kim, "Employing collective intelligence for user driven service creation", IEEE Commun. Mag., vol. 1, (2011), pp. 76–83.
- [6] A.K. Dey, T. Sohn, S. Streng and J. Kodama, "iCAP: Interactive prototyping of context-aware applications", In Proceedings of the International Conference Pervasive 2006, Dublin, Ireland, (2006), pp. 254–271.
- [7] H.S. Choi, J.Y. Lee, N.R. Yang and W.S. Rhee, "User-centric service environment for context aware service mash-up", In Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea, (2014), pp. 388–393.
- [8] Z. Shelby, "Embedded web services", IEEE Wireless Communication, vol. 12, (2010), pp. 52–57.
- [9] D. Guinard and V. Trifa, "Towards the web of things: Web mashups for embedded devices", In Proceedings of the International World Wide Web Conferences (WWW), Madrid, Spain, (2009), pp. 15–22.
- [10] E. Avilés-López and J.A. García-Macías, "Mashing up the Internet of Things: A framework for smart environments", EURASIP J. Wirel. Commun. Netw. vol. 3, (2012), pp. 1–11.
- [11] S.N. Han, G.M. Lee and N. Crespi, "Semantic context-aware service composition for building automation system", IEEE Trans. Ind. Inform., vol. 10, (2010), pp. 752–761.
- [12] C.H. Quyet, "A Hybrid Semantic Energy Saving Service for Cloud-Based Smart Homes", Master's Thesis, School of Engineering and Technology Thailand, Pathumthani, Thailand, (2013).
- [13] T. Rodden, A. Crabtree, T. Hemmings, B. Koleva, J. Humble, K.P. Åkesson and P. Hansson, "Configuring the ubiquitous home", In Proceedings of the 6th International Conference on Designing Cooperative Systems, Riviera, France, (2004), pp. 227–241.
- [14] M. Chen and B. Shen, "A semantic unification approach for M2M applications based on ontology", In Proceedings of the 2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Wuhan, China, (2011), pp. 265–271.
- [15] B. Christophe, "Semantic profiles to model the web of things", In Proceedings of the 2011 IEEE 7th International Conference on Semantics Knowledge and Grid (SKG), Beijing, China, (2011), pp. 51–58.
- [16] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura and E. Jansen, "The Gator Tech Smart House: A Programmable Pervasive Space", Computer, vol. 38, no. 3, (2005), pp. 50–60.
- [17] A.J. B.Brush, B. Lee, R. Mahajan, S. Agarwal, S. Saroiu and C. Dixon, "Home automation in the wild: challenges and opportunities", In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, Vancouver, BC, Canada, (2011), pp. 2115–2124.
- [18] W. Keith Edwards, R. E. Grinter, R. Mahajan and D. Wetherall, "Advancing the State of Home Networking", Communications of the ACM, vol. 54, no 6, (2011), pp. 62–71.
- [19] C. Reinisch, M. J. Kofler and W. Kastner, "ThinkHome: A Smart Home as Digital Ecosystem", In Proceedings of 4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2010), (2010), pp. 256–261.
- [20] P. Fox, D. McGuinness, R. Raskin and K. Sinha, "A Volcano Erupts: Semantically Mediated Integration of Heterogeneous Volcanic and Atmospheric Data", In Proceedings of the ACM first workshop on

- CyberInfrastructure: information management in eScience (CIMS '07), Lisbon, Portugal, (2007), pp. 1–6.
- [21] K. Mohamed and S. Chambless, “Model Driven Ontology: A New Methodology for Ontology Development”, (2008).
- [22] J. Undercoffer, A. Joshi and J. Pinkston, “Modeling Computer Attacks: An Ontology for Intrusion Detection”, In Giovanni Vigna, Erland Jonsson, and Christopher Krügel, editors, 6th International Symposium on Recent Advances in Intrusion Detection, volume 2820 of Lecture Notes in Computer Science, (2003), pp. 113–135.
- [23] F. Freitas, Z. Candeias Jr, and H. Stuckenschmidt, “A new Usage for Semantic Technologies for eGovernment: Checking Official Documents’ Consistency”, Electronic Journal of e-Government, vol. 8, no. 2, (2010), pp. 121–134.
- [24] R. A. Smith, “Designing a cartographic ontology for use with expert systems”, In Proceedings: A special joint symposium of ISPRS Technical Commission IV & AutoCarto in conjunction with ASPRS/CaGIS 2010 Fall Specialty Conference, Orlando, Florida, (2010).
- [25] M. Silva, E. Endhe, E. Costa, I. I. Bittencourt, H. Barros, L. Dias da Silva, A. Pedro da Silva and D. Vêras, “Combining Methontology and a Ontology Driven Approach to Build an Educational Ontology”, IEEE Multidisciplinary Engineering Education Magazine, vol. 6, no. 3, (2011), pp. 11–18.
- [26] J. Bayer and D. Muthig, “A View-Based Approach for Improving Software Documentation Practices”, In Proceedings of the 13th Annual IEEE International Symposium and Workshop on, Engineering of Computer Based Systems (ECBS '06) , (2006), pages 278–287.
- [27] “Net!Works European Technology Platform”, Expert Working Group on Smart Cities Applications and Requirements. White Paper, (2011).

Authors



Jeong-Sig Kim, he received the Ph. D. degree in Electrical Electronic and Computer Engineering from SungKyunKwan University, Rep. of Korea in 2007. He worked the development of computing system in enterprise knowledge strategy consulting during 2005 ~ 2007 and the DRM(Digital Rights Management) development in DRM-inside during 2007 ~ 2012. Since 2012, He is currently an assistant professor in the Department of Computer and Mobile Convergence at Gyeonggi College of Science and Technology, Rep. of Korea. His research interests are Error Resilient Coding, Digital Signal Processing, DRM System, Multimedia Processing & Communications System, Big Data System and Software Engineering.



Eung-Sungg Kim, he received the Ph. D. degree in Electronic Engineering from SungKyun-Kwan University, Rep. of Korea in 1998. Since 2000, He is currently a professor in the Department of Computer and Mobile Convergence at Gyeonggi College of Science and Technology, Rep. of Korea. His research interests are Image and Video Processing, Digital Signal Processing, Multimedia Communications, and Neuron Network.



Jin-Hong Kim, he received the Ph. D. degree in Information and Communication Engineering from SungKyunKwan University, Rep. of Korea in 2006. He is currently an assistant professor in Department of Information and Communication Engineering. His research interests are Big Data Platform, Intelligent Vehicular Network, Software Architecture and Design, and Datamining.