

A proposal of Cloud-based Home Network System for Multi-Vendor Services

Satoshi Takatori, Shinsuke Matsumoto, Sachio Saiki and Masahide Nakamura

Graduate School of System Informatics, Kobe University

1-1 Rokkodai, Nada, Kobe, Hyogo 657-8501, Japan

Email: takatori@ws.cs.kobe-u.ac.jp, shinsuke@cs.kobe-u.ac.jp, sachio@carp.kobe-u.ac.jp, masa-n@cs.kobe-u.ac.jp

Abstract—A home network system (HNS) provides value-added services for home users by networking household appliances and sensors. In the conventional architecture, the HNS appliances and services are tightly coupled. It is therefore difficult for users to freely choose their favorite appliances and services. In this paper, we propose a new HNS architecture that accommodates multi-vendor services by extensively using cloud technologies. The new architecture manages individual HNS operations and data as standard services within the cloud. The vendor services must go through the cloud to access the HNS. Thus, loose coupling among the HNS and services can be achieved. As a result, the proposed architecture realizes more flexible HNS beneficial for both users and vendors.

Index Terms—home network system, cloud computing, smart city, multi-vendor system, HNS services

I. INTRODUCTION

Home Network System (HNS) is an emerging ubiquitous system in today's advanced information society[1]. HNS provides various value-added services by orchestrating networked home appliances and sensors. Some examples of HNS services are: energy-saving service[2], appliance control service[3] and context aware service[4]. Recently, some applications using HNS have appeared on the market. Furthermore, the Japanese government subsidizes to HEMS (House Energy Management System), which aims to optimize the energy consumption in home environment.

In spite of long-term research and development, HNS have not become common in our society. One of the major reasons is that HNS appliances and services are *tightly coupled* together. This is due to a business model, which is generally called "vendor lock-in". Because of the coupling, it is difficult to provide multi-vendor HNS services on a single HNS environment. For example, a customer, who has a plan to introduce a new HNS service into his/her house, is forced to buy a new home appliances and devices instead of using some existing appliances in the house. He/she cannot use their favorite devices and forced to pay expensive costs. In the result, the spread of HNS does not proceed.

In our previous work[5][6], we have researched a HNS architecture for providing multi-vendor HNS service. This architecture resolves the tight coupling by applying a concept of service-oriented architecture (SOA)[7]. However, in our research, we have assumed that the HNS architecture is deployed only within a home area network (HAN). Therefore, users of HNS services are needed to introduce a home server in their

home and needed to manage the HNS services in themselves. This problem also causes a problem to pay extra efforts for continuous server maintenance and service management. Therefore, there is no advantages compared to current HNS in respect of cost.

In this research, we propose a novel HNS architecture, named Cloud HNS, that accommodates multi-vendor services by building on the concept of cloud computing. The new architecture manages individual operation for HNS appliances and sensed data as a standard and abstracted service within the cloud. Vendor-provided services access to HNS and obtain basic HNS operations through the cloud.

Specifically, Cloud HNS consists of three components: *HNSLite*, *HouseCloud*, and *Application Service Provider (ASP)*. *HNSLite* is a lightweight HNS that removes a home server from conventional HNS. *HouseCloud* manages registration information and devices of each household based on the SOA in the cloud. *ASP* provides various HNS services. Each component provides HNS services by sharing the role of as follows: *HNSLite* performs information collection and physical manipulation of the home devices. *HouseCloud* standardize information that customer premises and appliance control, provide as services. *ASP* develops and operates value-added services.

In this paper, first we explain the proposed Cloud HNS architecture, and then we discuss about on the three layers. Finally, we consider the way of providing HEMS on Cloud HNS, as a case study.

II. PRELIMINARIES

A. Home Network System (HNS)

HNS is a IT-enabled system which provides value-added services by networking household appliances (e.g., TV, air conditioner, fan and light) and sensors (e.g., thermometer and illuminometer). HNS provides application services include personal home controllers, remote monitoring/controlling, appliance orchestration, energy saving, context-aware services. There are existing HNS services, such as remote controlling by smart phone[8] or energy saving by control air conditioners automatically[9]. Currently, HNS receives some attention in wide research areas (e.g., smart city, energy-saving, crime prevention and health-care). Home appliance companies have begun providing actual HNS services and specialized appliances.

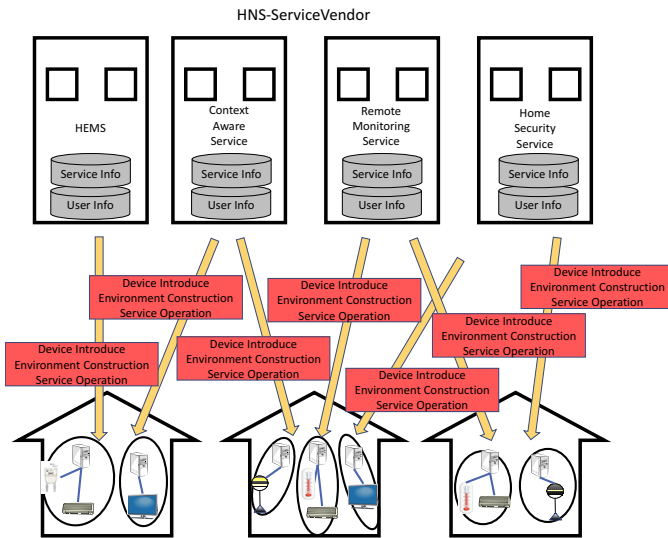


Fig. 1. Conventional architecture of HNS

Figure 1 shows an overview of the architecture of conventional HNS. Each house has built a HNS environment and has installed appliances, sensors, meters and a home server. The home server collects sensor data such as room temperature, humidity and a number of existing people. The server also collects historical data about appliance controlling. These collected data can be regarded as “house data”. Moreover, the server provides HNS services by control appliances based on real-world contexts by using house data.

Service vendors introduce HNS environment and provide HNS services in individual user’s home. HNS service information and user information, which are used by HNS services, are managed by the service vendors. A home server connects these database and individual HNS devices. Basically, service vendors conduct initial setup and issue maintenance of HNS environment. So the operation of the home server is delegated to home users. In some cases, service vendor directly provides services which can’t realize only the home server. For example, when a home security service detects a malicious intruder by using sensor data, a service vendor dispatch a security guard. Each vendor have databases about user information(e.g., name, address) necessary for contract and offer HNS services.

In the conventional architecture, if we want to use multi-vendor HNS service in a single house, we need to introduce different HNS environments and devices.

B. Cloud Computing

Cloud computing [10] is new computing paradigm that enables to provide various IT resources in on-demand through worldwide network without being conscious of substances. One of the well-known cloud computing models is IaaS (Infrastructure as a Service) which offers various computing infrastructure components such as physical server, storage and network. The others are PaaS (Platform as a Service) and SaaS

(Software as a Service). They offer computing platform and on-demand softwares as networked services, respectively.

C. SmartCity

Smart city is a next-generation city planning that aims to achieve sustainable society with ICT technologies [11][12]. In a smart city, various kinds of data are collected from sensors and devices. The data include, road traffic, power consumption, health status of inhabitants, and other any type of environmental measures. These data are analyzed with advanced and real-time data processing technologies to achieve affluent society.

III. CHALLENGE

In this paper, we address the following three challenges.

A. Challenge C1: Tight coupling between HNS device and service

In the conventional HNS architecture, HNS service requires specific household devices. These devices are developed with specialized features and industry specific protocols. Therefore, HNS user must install a complete set of specialized devices, communication protocols and services. This means it is difficult for user to freely choose their own favorite HNS services and devices.

Moreover, this tight coupling leads a lack of interoperability in terms of cross-vendor device migration. If a service vendor decide to terminate their providing HNS service, HNS user is needed to install an another complete set of HNS environment. Another case is that when a HNS specialized TV device get out of order, a user cannot use an alternative TV device as a HNS dedicated TV.

B. Challenge C2: Costs for Deployment and Management

User need initial costs for home servers and devices when introducing HNS in his/her house. In order to use HNS service, user must purchase vendor specific devices such as a home server that does not exist in the original home. Moreover, as mentioned in the challenge C1, even if other HNS has already been constructed in the house, it is not possible to reuse existing appliances and sensors. If user want to introduce new HNS service, it is necessary to purchase some devices that meets the HNS service.

Maintenance costs of home server occurs even later the introduction of HNS. Dealing with trouble in continuous operation of the home server gives a tremendous burden on user. Deploying a home server to each house may also forces spending expensive costs for maintenance and operation. If a home server breaks then an individual service setting is going to be lost.

C. Challenge C3: Distribution of information

In the conventional architecture, it is difficult for cross-vendor usage of each piece of information which managed by an individual service vendor. Thus, user is required to register their information (e.g., installed devices, family configuration, and their preferences) to all service vendors.

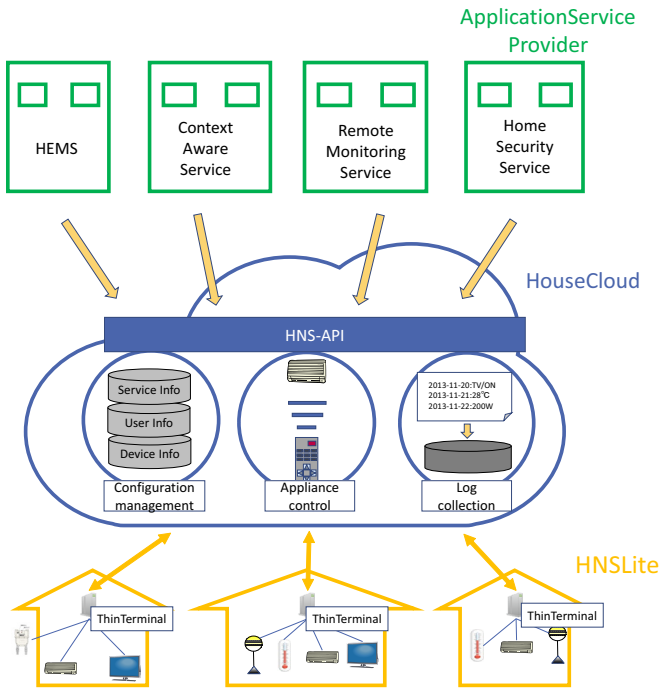


Fig. 2. A architecture of proposed Cloud HNS

Furthermore, in the situation where the information is distributed to each vendor, the development of service of smart city is also difficult. Service for smart city need to collect a wide variety of information such as energy usage and appliance status for each house. However, at present, it is necessary to refer to various information for all vendors.

IV. CLOUD HNS

A. overview

The goal of this research is to provide a framework which enables multi-vendor HNS service regardless of configuration of HNS. As an approach for the purpose, we propose a novel HNS architecture, named Cloud HNS, that accommodates multi-vendor services by building on the concept of cloud computing. To cope with challenge C1, we provide HNS service as a cloud service by isolating HNS services from HNS environment. Software process for HNS service conducted in home server is alternatively provided by Cloud HNS via internet. In addition, Cloud HNS centrally manages users' information in cloud. These approaches could be solutions to challenge C2 and C3, respectively.

B. Architecture

Figure 2 shows an overview of the architecture of Cloud HNS. The proposal architecture consists of three layers, *HNSLite*, *HouseCloud*, *ASP (Application Service Provider)*.

HNSLite is a lightweight HNS that removes a home server from conventional HNS. *HNSLite* obtains *house log data*, such as sensor data and device status of each household. House log data is stored in databases of following *HouseCloud* through

a network. Furthermore, it performs device control, according to the instruction from *HouseCloud*.

HouseCloud provides mainly the following three functions: providing services to control home devices, managing static master information required for service provision, accumulation of house logs data collected from each household. It is determined by user that how much publish these functions of *HouseCloud*.

ASP provides HNS services for *HNSLite*. *ASP* obtains all information necessary for HNS service from *HouseCloud*. If *ASP* wants to control devices in the user house, request to *HouseCloud*. There is no possibility that the *ASP* operate devices in the user house directly.

This layered architecture follows a design principle of loose coupling. Each of the three component keeps generality and independence of vendor specific protocols and features.

C. *HNSLite*

HNSLite is a lightweight version of HNS by removing a home server from the conventional HNS. In *HNSLite*, *ThinTerminal* is installed instead of conventional home server. *ThinTerminal* works as a gateway of home network and internet. So, *ThinTerminal* have only minimal functions to communicate with home devices. *ThinTerminal* does not provide high functionality service such as data collection and or cooperative control of home devices. *ThinTerminal* plays as an adapter that connects *HouseCloud* and *HNSLite*. All devices installed in home are connected to cloud server through *ThinTerminal*. A high functionality service, which provided on home server in the conventional architecture, is provided by *HouseCloud*.

User can suppress initial installation cost by removing the home server from home. In addition, user can suppress the maintenance cost for home server.

In order to provide HNS service without being bounded by vendor specific devices, *ThinTerminal* covers wide variety of communication protocols. In this way, *ThinTerminal* can communicate using a DLAN for TV, and in *ECHONETLite* for an air conditioner. User will be able to choose HNS devices in home without being aware of the communication protocol. It is also possible to cope without additional effort even when the communication protocol is unified in the future.

D. *HouseCloud*

HouseCloud is a central layer of Cloud HNS. *HouseCloud* provides *ASP* appliance controlling and house data (e.g., log and configuration information) as a standard HNS service that is independent from particular purpose or usage.

HouseCloud has the following three functions.

- Management of the configuration information
- Collection of house log data
- Appliance controlling

1) *Management of the configuration information*: *HouseCloud* manages various configuration information about *HNSLite* to be connected *HouseCloud*. Most representative

of configuration information is *home information*. Home information consists of the following information: house information (e.g., room layout and address), resident's personal information (e.g., family structure and name of residents), device configuration information (e.g., device location, type and name), and service information (list of registered services and its settings).

If there are any changes in home environment, such as change of configuration of the device or family structure, user updates home information managed by HouseCloud. By centralizing all information in HouseCloud, some changes are reflected in all services by one update. It is not necessary for user to update the information for each service.

For security reasons, it is necessary to completely hide raw data of home information from other layer. HouseCloud must strictly control the home information. HouseCloud publishes home information to ASP only if user gives permission to access to their data.

In addition, HouseCloud manages *device information*. Device information includes all house devices such as sensor and appliance. The device information consists of the type of device, name, model number, specifications and the available functions. These information is used by device controlling in a house. Device vendor registers device information to HouseCloud. As a result, the device will be available at HNS.

Moreover, HouseCloud manages *service information* so that users can search for available HNS services. Service information is information about the resources needed for service provision, such as service name, processing details, the type of using devices and log data. In addition, service information include contract information, such as ASP name, service rates and period. Service information is used (registration, update, delete) to service management by the user. User can choose a service that suits user self by compare and evaluate HNS services provided by ASP from different aspects based on service information.

2) *Collection of house log data*: HouseCloud accumulates and manages *house log data*. House log data consists of environment log and device log. Environment logs are retrieved by environmental sensors. Various type of logs are included as environment log; temperature, humidity, number of residents and so on. Device logs includes state and operation history of each device. Some examples are switching history of a room light and current channel of TV.

HouseCloud periodically obtains operation history and sensor values by accessing appliances and sensors in HNSLite. Retrieved data, named house log, is processed and stored in a database managed by House Cloud. ASP accesses house log through an API that HouseCloud provide and use as information add value to HNS services.

HouseCloud also accumulates infrastructure usage information (e.g., water, gas and electricity) that is published from each infrastructure company. These data is also referred by HNS services.

3) *ApplianceControl*: HouseCloud abstracts appliance control in HNSLite and provides it as a service to ASP. If there

is a request for appliance control from the HNS service, HouseCloud converts the request into operation command and send it to HNSLite. Appliance control service is available from HNS services through APIs that HouseCloud provide. If these APIs are called, HouseCloud determines whether or not to run the appliance control on the basis of the service setting information. When a user does not gives permission to control the specified home appliance, HouseCloud does not control it.

E. *ApplicationServiceProvider (ASP)*

ASP developments and provisions various HNS services using the HNS. Specific examples of service include HEMS, home security, remote monitoring/controlling, and smart city service. ASP registers information of their developed services to HouseCloud. User can freely choose HNS services which are suitable for own HNS environment from various services that are registered in HouseCloud.

When performing provision of services, ASP treads following three phases. service development, service registration to HouseCloud, operation of service. In conventional architecture, ASP is required to develop a specialized device for their service along with the service development. However, in Cloud HNS, ASP is only required to built software because use the existing devices in the house.

Next, ASP registers developed HNS services to HouseCloud. Specifically, registered information includes service name, description, price, required minimum environment and default configuration settings. A HNS user selects some listed HNS services with reference to registered service information.

When registration to HouseCloud is completed, operation phase of HNS service begins. Because home information is centrally managed on HouseCloud, ASP is possible to reduce costs of security measures. A HNS service in operation phase obtains from HouseCloud information such as room temperature and power consumption of the user's home utilizing the service. At this time, information to be obtained is limited to information that the user permit. By using the acquired information, ASP performs value-added services corresponding to the situation of each home. If there is a need to control devices in the home, ASP operates the device through API that HouseCloud provides.

V. CASE STUDY

In this section, we consider how to achieve HEMS (House Energy Management System) on Cloud HNS as a case study. Figure 3 shows implementation of HEMS on Cloud HNS.

At first, user picks his/her favorite service from various types of HEMS services provided by ASP in accordance with home environment and purpose of their own. Examples of HEMS service are; visualization service of power consumption, optimization service of appliance control for energy saving, creating energy service using storage batteries and solar power system.

Type of HEMS services that users are using is managed by HouseCloud. In addition, information about the configuration of HNS service, such as functions of home devices to expose

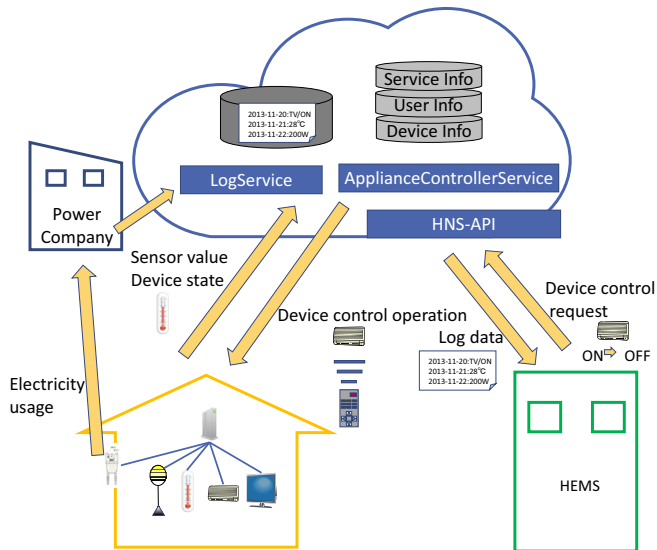


Fig. 3. HEMS on Cloud HNS

to service, reference authority of the home information, is also managed. For example, in the above energy saving service, it is necessary that HouseCloud permit the service to control appliances in the house for energy-saving and publish power consumption. Authority of the appliance control can be freely controlled by the user. If user does not want to operate TV by remote, service control TV when using the energy-saving services, the user can set not to allow control TV on HouseCloud.

Provision of services begins after user select and register the service. HouseCloud acquires the log data of appliances and sensors in HNSLite and store it in an own database. This is collected from the time regardless of the service the user chooses, and start using HouseCloud. If the battery or solar power generation system exists, HouseCloud also accumulate state their logs. In this implementation, HouseCloud also collects power usage that is obtained from the power company.

HEMS service get the configuration information and house log data through APIs of HouseCloud. Service perform the processing on the basis of these information. In the case of the energy saving service, if the lighting is lit, the service makes a policy that turns off the lighting. Then, the service requests appliances control HouseCloud by calling APIs. In response to the call of APIs, HouseCloud issues a control command to the home device. In this case, HouseCloud issue a command to turn off the lighting.

As you can see in this case study, the proposed architecture has a structure that users can select services freely by loose coupling HNS in the user house and HNS services that is provided by ASP.

Further, it is not necessary for ASP to be aware of the details of each HNS. Standard process that needs to access the HNS(e.g., appliance control or acquisition of home information) can be performed easily by delegate to HouseCloud.

TABLE I
COMPARISON BETWEEN THE CONVENTIONAL HNS AND CLOUD HNS
FROM USER PERSPECTIVE

Perspectives	Conventional HNS	Cloud HNS
Flexibility of selection for HNS services	Low(only service that a vendor who user construct provided)	High(all services on cloud)
Reusability of devices	Low	High
Introduction costs	High (need home-server and devices for the HNS service)	Low (only need Thin Terminal)
Maintenance costs of services	High (users must manage services)	Low (HouseCloud manage all service)
Availability of services when network disconnected	Possible	Impossible
Response speed of service invocation	Fast	Slow (depends on network latency)

VI. DISCUSSION

A. comparison

Table I and Table II show comparison with Cloud HNS and conventional HNS.

First, we describe the comparison from the perspective of HNS user. There are some advantages that can be enjoyed by user in Cloud HNS, such as flexibility of service selection and reusability of devices are high. User can choose wide variety of HNS service provided on Cloud HNS. In addition, Cloud HNS has distinct advantage of maintenance costs and initial costs. On the other hand, there is a problem that user can not use services when network disconnected. In order to provide the service using resources on the internet in Cloud HNS, availability of service depends on the state of the network. Therefore, there is also a problem that the network congestion occurs, the response speed is slow.

Table II describes comparison from the perspective of vendors. In Cloud HNS, maintenance costs of personal information and service operation costs can significantly be reduced. Each ASP only bears responsibility for the service that was developed internally. Thus, it is not necessary for ASP to manage the user's information and devices. ASP also can reduce the cost to cope with device failure. Multi-vendor services linked services from different vendors is facilitated, risk of business expansion is also reduced. Because of the architecture that HouseCloud aggregate distributed information of HNS in the Cloud, the development of Smart City services can be facilitated. In the CloudHNS, on the other hand, there

TABLE II
COMPARISON BETWEEN THE CONVENTIONAL HNS AND CLOUD HNS
FROM HNS VENDOR PERSPECTIVE

Perspectives	Conventional HNS	Cloud HNS
Running costs of services	High (it is necessary to manage devices of a user's house)	Low (vendors only manage software of services)
Costs of managing personal information	High	Low (HouseCloud centrally manages information)
Service cooperation among services which different vendors provide	Difficult	Easy
Risk of business expansion	High (risk is greater because vendors must produce device with the service)	Low (only development costs of services)
Development of service for smart city	Difficult (because of distributed information)	Easy (vendor can use centralized information that is consolidated in HouseCloud)
Flexibility of service development	High	Low (depends on APIs provided by HouseCloud)

is a possibility that the flexibility of service development is reduced. Each service provided by ASP must be realized in the extent of APIs HouseCloud provided and using only the data that the user has permission. Therefore, ASP will not be able to customize service flexibly than traditional HNS. Not only HNS service but also general cloud service has this limit.

VII. RELATED WORK

Smart Serve is one of the similar approach to Cloud HNS in terms of remote operation of home appliances through the internet. Smart Serve is a networked service that can remotely control home appliances through VPN connection. Home appliances are connected to LAN and controlled by a specialized device, called service adapter. The difference between smart serve and Cloud HNS is whether HNS available from multi-vendor services. Smart serve only available device remote control service of vendors who provide this. However, in the proposed architecture, it is possible to use a wide variety of HNS services, including services which are provided by the different vendor that construct HNS in the user's home.

VIII. CONCLUSION

In this paper, we proposed a novel architecture, named Cloud HNS. Cloud HNS enables loose-coupling between HNS service and HNS environment. Cloud HNS regards basic HNS functions, such as device controlling and information

management, as a cloud service. As a case study, we discussed how one of the most typical HNS service, energy saving service, is realized on Cloud HNS. In addition, we considered benefits and limitations of Cloud HNS from the perspectives of user and vendor. Our future work includes design and implementation of a service API and specific data models for implementing the proposed architecture.

IX. ACKNOWLEDGMENTS

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (C) (No.24500079, No.12877795), Scientific Research (B) (No.23300009)] and Sekisui House, Ltd.

REFERENCES

- [1] M. Nakamura, A. Tanaka, H. Igaki, H. Tamada, and K. Matsumoto, "Constructing home network systems and integrated services using legacy home appliances and web services," *International Journal of Web Services Research*, vol. 5, no. 1, pp. 82–98, Jan 2008.
- [2] M. A. A. Pedrasa, T. D. Spooner, and I. F. MacGill, "Coordinated scheduling of residential distributed energy resources to optimize smart home energy services," *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 134–143, Sept 2010.
- [3] Y. Tajika, T. Saito, K. Teramoto, N. Oosaka, and M. Isshiki, "Networked home appliance system using bluetooth technology integrating appliance control/monitoring with internet service," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1043–1048, Nov 2003.
- [4] N. Cohen, J. Black, P. Castro, M. Ebling, B. Leiba, A. Misra, and W. Segmuller, "Building context-aware applications with context weaver," pp. 1–12, 2004.
- [5] M. Nakamura, Y. Fukuoka, H. Igaki, and K. Matsumoto, "Implementing multi-vendor home network system with vendor-neutral services and dynamic service binding," in *Proc. IEEE International Conference on Services Computing (SCC 2008)*, Jul 2008, pp. 275–282.
- [6] T. Okamura, M. Nakamura, and H. Igaki, "Finding optimal energy-saving operations in home network system based on effects between appliances and environment," in *Proc. Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT2010)*, Jun 2010, p. CDROM.
- [7] T. Erl, *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2004.
- [8] "iTamaHome," <http://www.tamahome.jp/i-tamahome>, accessed: 2014-01-04.
- [9] "SmartHEMS," <http://www2.panasonic.biz/es/densetsu/aiseg/features/index.html>, accessed: 2014-01-04.
- [10] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599 – 616, 2009.
- [11] R. G. Hollands, "Will the real smart city please stand up?" *City: analysis of urban trends, culture, theory, policy, action*, vol. 12, no. 3, pp. 303–320, 2008.
- [12] A. Mahizhnan, "Smart cities: The singapore case," *Cities*, vol. 16, pp. 13–18, 1999.