

# Distributed semantic sensor web architecture

Hoan-Suk Choi

Dept of Multimedia Engineering  
Hanbat National University  
Daejeon, Korea  
hkrock7904@gmail.com

Woo-Seop Rhee

Dept of Multimedia Engineering  
Hanbat National University  
Daejeon, Korea  
wsrhee@hanbat.ac.kr

**Abstract**—Recently, data on the web has been increased by interactive service, social media and the pervasive computing technology. To utilize these huge data include sensor data, semantic sensor web service platform is needed. And it can provide the context-aware semantic web service. But, because sensor generated data is so dynamic, it can't always update and store. Therefore, we propose the distributed semantic sensor web architecture to separate processing of context information and service information. This paper analyzes the requirements of semantic web of things and proposes the semantic sensor web service environment to provide user centric context-aware semantic web service through the semantic sensor web platform.

**Keywords**—*Semantic sensor web; Semantic Web of Things; service platform; Context-aware semantic web; Pervasive service.*

## I. INTRODUCTION

Due to the spread of mobile smart device more than 5 billion, we can access anywhere, anytime to the Internet. Thus, social media data and multimedia data have grown rapidly by improving accessibility of the Internet [1]. Also, real time location service and augmented reality services (e.g., layer, foursquare, facebook's places) using smart device sensors (e.g., multiple axis gyro sensor, accelerometer sensor, proximity sensor, light sensor, GPS and digital compass) have been widely used. Therefore, people became more interested in the real world status information. And the research of pervasive computing for real world status information through sensors has continued in WSN (Wireless Sensor Network) and IoT (Internet of Things). Recently, to do this easily, web technologies need to be integrating sensors and device on the pervasive computing such as WoT (Web of Things). The open web standards are supported for information sharing and device interoperation. By penetrating smart things into existing web, the conventional web services are enriched with physical world services. This WoT vision enables a new way of narrowing the barrier between virtual and physical worlds [2]. It will accommodate trillions of sensors and devices on the web, and will greatly increase the size and scope of the web [3]. Therefore, the semantic approach was proposed because these huge data should be managed efficiently and provided to user systematically. It can enhance information processing, such as data search, data analysis and mining, situational awareness and question answering. In case of web searching, the major web search engines are applied semantic web to use these data on the web effectively [1]. The semantic web is an

extension of the World Wide Web by giving a semantics meaning to information on the web. And the data is formally defined in the form of ontology to interpret for machines. Also, the SWoT (Semantic Web of Things) is an emerging new paradigm that provides new experience by convergence of existing service, content and data generated by sensors through semantic and lightweight alternative protocols with REST principle [2], [4]. But, the semantic approach is difficult to provide an appropriate reasoning. Because the area of semantic knowledge is too broad to define of all condition and knowledge. And if all raw data of sensors forward to the web, it will be cause explosion of data. Also, updating and managing of sensor raw data increase the cost of network and service platform. So, the distributed architecture of semantic service environment to separate the sensor service domain and the semantic service platform is needed.

Therefore, we propose the SSW (Semantic Sensor Web) service environment for minimizing of the sensor data handling. It has the distributed SSW architecture, which is consisted of the SSWP (Semantic Sensor Web Platform) and the SG (Smart Gateway), to separate service domains, minimize and simplify the information process. This environment deploys the SG in each service domain to aggregate sensor data. The SG generates context information by the reasoning function and forwards only this context information to the SSWP. The SSWP aggregates service and context information to provide user centric context aware semantic web service. Through this proposed environment, user can acquire context information of other service domains regardless of location.

The rest of paper is organized as follows. Section 2 and 3 discuss the related works and the requirement of SWoT. Section 4 describes the SSW service environment. And section 5 describes the proposed functional architecture and service procedures about the SSWP and SG. Finally, the results of this research and some possible future works discuss in section 6.

## II. RELATED WORKS

For the smart environment society which is provided by the context aware smart services, SWoT connects things to the Internet, and provides whole information which are machine readable on the web. So, we need reasoning mechanism to convert raw data to context information.

---

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(grant number: 2012007452)

Many related projects and researches are actively progressed. As a European project, the WoO (Web of Objects) project [5] is a network and services infrastructure in ITEA 2 Call 5 project. The WoO aims to develop a network and services infrastructure for smart objects simplifying the development and the deployment of the distributed applications independent of proprietary protocols. Based on DPWS (Devices Profile for Web Services) and a fine grained semantic modeling of devices and services, security and privacy mechanisms will be assured at the device level. The WoO provide a reasoning-based objects adaptation to context. It includes the object reactivity in case of vital resources decrease, but also includes the autonomic collaboration with other smart objects for accomplishing complex tasks.

The DiYSE (Do-it-Yourself Smart Experiences project) project [6] was enabled people to direct their everyday environment into a highly personalized interaction experience that can span the home and city domains. The project provided sustainable marketplace for user-generated application for an IoT world, in which user can participate, reusing well abstracted components, capabilities and devices. As such, it goes beyond web, mobile or multimedia applications. Driven by a user-centered design methodology and concrete proof-of-concept demonstrators, the DiYSE project envisions innovating on new valuable interactive user experiences based on aware services, sensors, actuators and media devices.

The SPITFIRE (Semantic Service Provisioning for the IoT using Future Internet Research by Experimentation) [7] is an integrated project in the EU's FP7. The SPITFIRE provide vocabularies to integrate descriptions of sensors and things with the LOD cloud, semantic entities as an abstraction for things with high-level states inferred from embedded sensors, semi-automatic generation of semantic sensor descriptions and search for sensors and. On top of this infrastructure, applications are assembled by issuing search requests for matching sensor services and invoking found services directly.

As research papers, the SIoT (Social Structure to the IoT) [8] proposed to exploit human social network relationships to share the resources offered by a given smart thing, such as smart things enabled to support web services usable by friends of their owner. The Semantic Profiles to Model the "Web of Things" [9] proposes the use of semantic web principles to establish a set of models building the basis of a fine grained and extensible object description document. It proposes a semantic-based framework to process such profiles, in order to enable connected objects to be used in web applications. And it provides support for composing connected objects together by proposing to share a data model between different object providers, enabling then same format to be used for further content generation. The Discovery Mechanisms for the Sensor Web [10] is to search for sensors, exploit basic semantic relationships, harvest sensor metadata and integrate sensor discovery into already existing catalogues. Sensor.Network [11] is a web-based infrastructure for storing, sharing, searching, visualizing and analyzing data from heterogeneous devices. And it discuss data centric collaborations and social

interactions made possible by features such as tagging, searching and sharing with fine-grained privacy controls.

A semantic enhanced service exposure model [12] propose a converged service exposure framework for user generated application creation in a more open and sustainable marketplace. This framework is intended to be one in which even non-professional users can easily reuse existing services, capabilities, and resources to create new services, and thereby profit from a highly personalized, meaningful communication and interaction experience. It includes telecom, web, device and user generated service exposure. Also, it proposes service description, service discovery, semantic annotation and publication component.

However, the [4], [9], [10] are too much focus to a semantic knowledge. And the [3] is full connection of sensor approach. The semantic approach is difficult to provide an appropriate reasoning because the area of semantic knowledge is too broad to define of all condition or knowledge. The full connection of sensor approach generates data so dynamically to manage and store. And it leads to an unbearable level of network traffic [12]. The middleware approach [11] provides the centralized functionality. It doesn't matter about centralized problem when the service environment and the amount of data which generated by sensor and device is smaller scale of application. But, when it comes to a larger scale, the processing model of data should be distributed and simplifying [14].

### III. REQUIREMENTS OF SEMANTIC WEB OF THINGS

In this chapter, we describe the requirements of SWoT for SSW environment to support a user-centric context web service and describe how to apply these requirements to the proposed architecture.

#### A. Connecting sensor to Internet

Existing Internet protocols such as HTTP, TCP are too complex and require resources for use in the constrained environment such as WSN. And resource-constrained sensors are difficult to apply these protocols. Therefore, lightweight alternatives are being proposed. The Representative protocols are 6LoWPAN and the CoAP. 6LoWPAN is a lightweight IPv6 adaptation layer allowing sensors to exchange IPv6 packets in the Internet. CoAP provides subset of HTTP methods (GET, PUT, POST, and DELETE) and lightweight alternative of HTTP [3]. In order to achieve it, we provide the connectivity function which is an independent physical interface and REST APIs on the SG.

#### B. Semantic description about sensor and device

The context information is represented in the form of machine readable description on the web (ex: RDF triples, XML, JSON). In order to achieve it, we provide convert function and generate context information through the context descriptor on the SG and the semantic data translator on the SSWP.

### C. Search for sensors and abstraction algorithm

Assume that the device information was comprised in the context description, the service is able to searching devices using special query language [8]. If we find the necessary device and get information, we need to convert this information to status information, such as “The printer1 is working” and “Highway 33 will be icing” by sub-zero temperature value from multiple temperature sensors. In order to achieve it, we provide the RRF (Rule base Reasoning Function) on the SG and aggregate this information using the information crawler on the SSWP.

### D. Integrated with static and dynamic information

The status information for the context based service should be integrated. This status information consists of the static information such as web page and the dynamic information that is generated from device to provide context aware service. In order to achieve it, we provide abstraction, registration, management of related services using the SBF (Service Brokering Functions) on the SSWP. When it receives service request, acquires necessary static information from the service and content provider and devices through the SG. Using this information, the context based service information can be generated.

## IV. SEMANTIC SENSOR WEB SERVICE ENVIRONMENT

The SSW service environment is able to provide a user centric context-aware web service which provides service publishing using context information about real world entities according to user’s information such as context, tastes, history and reasoning about relevant sensors. Figure 1 shows the overview of the proposed SSW service environment.

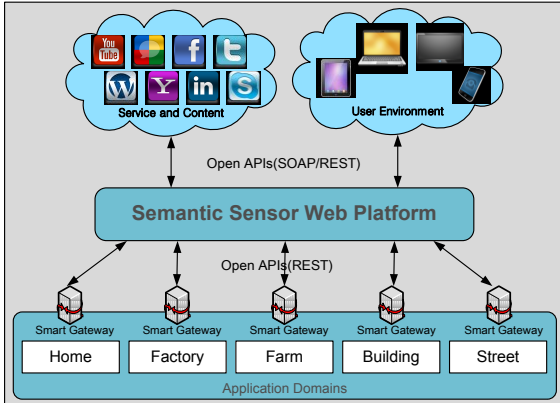


Figure 1. Overview of the semantic sensor web service environment

This environment includes various SG according to each application domain such as smart home, factory, farm, building and street. Also, the user can provided service of other application domains because the SSWP aggregates and provides the context information of other application domains. So when you stay at home, you can monitor state of farm, or receive notification message of factory state.

On the other hand, the service provider requires the context information about real world entity that needs to provide the service through the SSWP. The SSWP integrates and

processes the context information and service information to provide a user centric context-aware web service using open APIs (SOAP/REST). Also, if a user owns multiple devices and wants the user device independent service, the SSWP manages the user device profile and preference for the device-specific service. For example, when a user requests by tablet pc, the SSWP generates service UI (user interface) according to specification of device. So, the service UI provides a proper tablet interface such as, touch interface, show more detail information than mobile phone. As a result, the SSWP provides same environment to various user devices and get user information such as context, tastes, history. And it provides the context information to the service and the content provider and publishing a user-centric service to user. For a detail explanation of the SG and the SSWP is in section 5.

### A. Information processing levels

The information processing about the SSW service environment is consists of three levels as shown in figure 2.

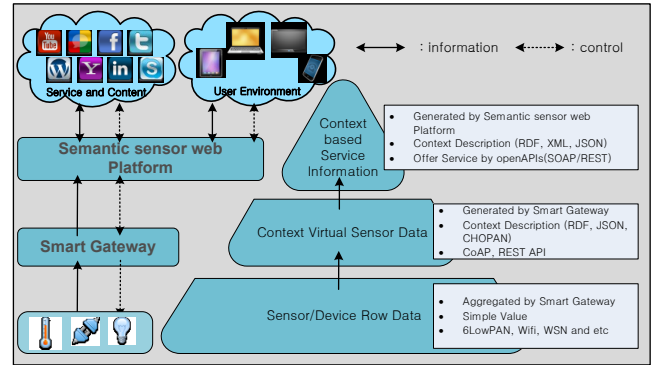


Figure 2. Information processing levels

First, the sensor/device raw data level has a simple value which aggregated by the SG using multiple interfaces. It is a real-time data such as temperature value C. If these simple values are transmitted to the SSWP without any processing, the processing cost of networks and SSWP will be increased. Therefore, the simple value information is transmitted only to the SG. Second, the context virtual sensor data level has a context information which generated by the SG using the RRF. For example, the service rules have some threshold value about special event. Through this rule, a decision can be made about status such as “The printer1 is working” by energy consumption of appliances. In addition, it makes a context decision by the values from several devices such as “Highway 33 will be icing” by sub-zero temperature value from multiple temperature sensors. Third, context based service information level has a high-level state information which generated by the SSWP using multiple context information that was received from multiple services or content providers and SGs. For example, the SSWP gets the information from the weather service, user profile, history and location and sends “wash recommended” as a push message with the location about nearest DIY car washing place to user and several information such as “last car wash date”, “the weather is sunny until next Saturday”, “list of car wash location” and “Tom like DIY car wash”, etc.

## V. FUNCTIONAL ARCHITECTURE FOR THE SEMANTIC SENSOR WEB SERVICE ENVIRONMENT

The data which generated by sensor or device is highly dynamic. If all data are transferred to SSWP and updated always, the overhead and utilization of networks are too increased [12]. Therefore, we propose the two separated agents, the SG and the SSWP. The SG doesn't forward sensor data to the SSWP except the context information generated by the simple RRF. And the SSWP provides the SBF for user centric service environment according to context, user, service, content information. In this chapter, we propose the functional architecture of two agents.

### A. Functional architecture of the smart gateway

The SG is installed in each service domain. And it aggregates the data which generated by sensors or devices in its domains. Also, it provides rule based reasoning and context description and sends the context information to the SSWP. The SG is consists of the multiple interface drivers, the connectivity function, the MF (Management Functions), the context descriptor, the RRF, repositories and the light weight web server as shown in Figure 3.

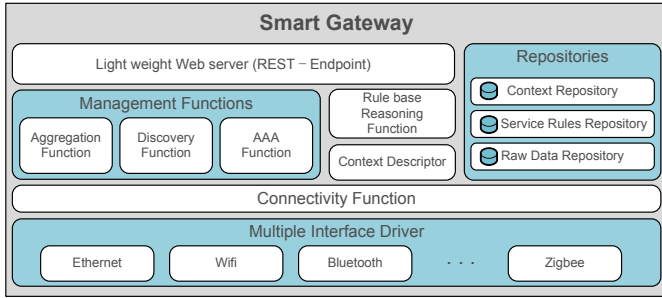


Figure 3. Functional architecture of the smart gateway

The multiple interface drivers support various interfaces of sensor and device, such as Ethernet, Bluetooth, RF, ZigBee, RFID, WiFi. The connectivity function provides a management of the multiple interface drivers. And it provides connectivity between the MF and the multiple devices. Therefore, it is able to provide connectivity to manage physical device regardless of interface type.

The MF are consists of the aggregation function, the discovery function and the AAA function. It stores the service requirements and service rule information at repositories by the sub functions. And it orchestrates sub functions and other functions of the SG and communicates with the SSWP using light weight web server which support APIs based on the REST principle. The aggregation function collects the raw data from the connected devices periodically, and stores the aggregated data to the RDR (Raw Data Repository). The aggregation function operates in pull or push mode. The pull mode is triggered by the aggregation function to satisfy the requirement of the SSWP. And the push mode is triggered by the RRF when the particular event is detected. For a detail explanation of the information procedure and aggregation function's operation is in Figure 4. The discovery function searches the device which is able to generate the required

information according to the information request message. And it makes a connection with the target device to acquire service required information using the connectivity function. The AAA function determines the ownership and permissions about devices that required by the MF and located in own range. The context descriptor generates context information. It converts aggregated raw data from devices, such as temperature value C or movement detection time T to the context description type. Also, it annotates the additional metadata information of device (ownership, deploy date, location, etc) to context information. The SRR (Service Rules Repository) has some threshold value about special event which defined by the service abstraction function of SSWP. Through this simple rule, the RRF makes a decision about real-world entities' state such as "The printer1 is working" by stored data in the RDR about energy consumption of appliances. In addition, it make a context decision by the values from several devices such as "Highway 33 will be icing" by sub-zero temperature value from multiple temperature sensors that are located near Highway 33.

The repositories are consists of the RDR, the SRR and the CR (Context Repository). The RDR stores the raw data that acquired from devices, such as temperature value C or movement detection time T. And the SRR stores the service rules acquired from devices web platform and offers the information to the RRF. This information includes some threshold value about target device of required service, event type, target service information and context value of the event such as "icing", "working", "open" and etc. The CR stores the context information which generated by the RRF. It includes context value, target service information, appropriate device information. The light weight web server provides REST API to connect between the SG and the SSWP. So, the SG can communicate with the SSWP using REST API.

Figure 4 shows service procedure of the SG. The service procedure is consists of service request and event detection procedure.

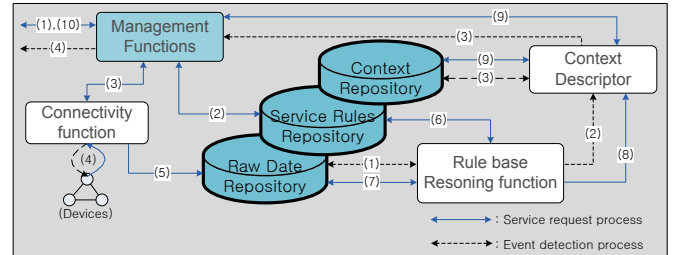


Figure 4. Service procedure of the smart gateway

The service request procedure is as follows:

- 1) The SG receives the service information request message from the SSWP using REST APIs. The service information request message includes the service rule and required information.
- 2) The MF search devices which are able to provide appropriate context through the discovery function. And, it stores service rules information in the SRR.

- 3) The MF request to collect information to the connectivity function through the aggregation function.
- 4) The connectivity function aggregates the required information from appropriate devices.
- 5) When aggregation is finished, the connectivity function stores the aggregated information to the RDR.
- 6) ,7) The RRF gets service rule information and aggregated raw data from the SRR and the RDR.
- 8) The RRF generates context information and forwards it to the context descriptor.
- 9) The context descriptor converts the context information to the form of semantic description. Also, sends the converted information to the CR and the MF.
- 10) The MF provide the required information to the SSWP. If these procedures are done and same service information is needed in next time, the SSWP able to get this historical information from the CR through the MF of SG.

The event detection procedure is as follows:

- 1) When the service request procedure is done, the aggregation function refreshes same service information. Therefore, the steps 3) ~ 5) of service request procedure are repeated and updates the data to the RDR. The RRF checks the data and detects the rule base event.
- 2) If the event was detected, the RRF forwards this event which includes urgent status to the context descriptor.
- 3) The context descriptor converts context information, includes urgent status, to the form of semantic description. Also, sends the converted information to the CR and the MF.
- 4) If the received context information includes urgent status, the MF of the SG forward it to the SSWP to provide the push notification service to user.

#### B. Functional architecture of the semantic sensor web platform

The SSWP aggregates and stores the context information which aggregated by the SGs. It provides service & content discovery and generates summarized information using the service abstraction function and service information in the SR (Service Repository). As a result, this platform provides the user-centric SSW service to proper user environment through the user information. The SSWP is consists of the MF, the information crawler, the SBF, the semantic data translator, the reasoning function, the repositories and the web server as shown in Figure 5.

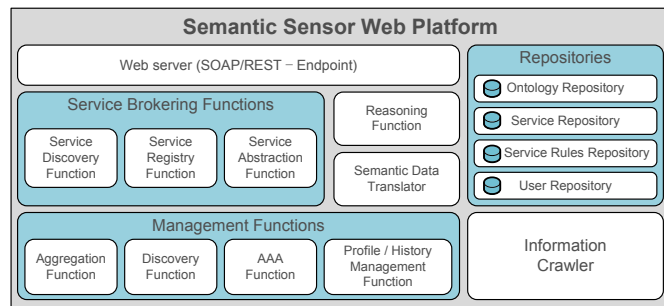


Figure 5. Functional architecture of the semantic sensor web platform

The MF are consists of the aggregation function, the discovery function, the AAA function and the profile/history MF. It provides discovery, selection and aggregation of necessary information according to request of the SBF. And it orchestrates sub functions and other SSWP. The aggregation function collects the context information from the SGs and other content or service provider through the information crawler. If the particular service needs more information of other content or service provider, the aggregation function requests to the information crawler according to service location that acquires by the service discovery function of the SBF. And the aggregated information is stored at each repository. The discovery function searches a SG to acquire the required context information and sends the location information of the SG to the aggregation function. The AAA function determines the ownership and permission about the SG, service and user information that required by the MF and provides security. The profile / history management function manages the user profile and history of service information and stores it to the user repository. Then, the SSWP can offer user-centric service. The information crawler provides aggregation the context information from many SGs and service or content provider periodically. And, it forwards to reasoning function. Sometimes it triggered by a request from aggregation function when receive service request.

The SBF are consists of the service discovery function, the service registry function and the service abstraction function. It communicates with the service and content provider using the web server and provides user service environment and service entry point through SOAP or REST API. The service discovery function searches the related service and content according to the service request. The service registry function registers and manages SSW environment service which generated by this platform and stores this service information in the SR. The service abstraction function summarizes the service information such as requirement, type, service rules and stores the summarized information in the SR. The semantic data translator converts summarized service and user information to the context description type and stores it at the each repository. The reasoning function generates high-level state information using multiple context information that was received from multiple service/content providers and the SGs.

The repositories are consists of the OR (Ontology Repository), the SR, the SRR and the user repository. The OR stores the high-level state information which is described in context description type. This information was generated by the reasoning function according to multiple context information that was received from multiple service, content providers and the SGs. When receives service request from user, the SBF provides this information from the OR. The SR stores the service information acquired from the service abstraction function. The SRR stores the service rules acquired from the service abstraction function and offers the information to the SG for rule based reasoning. This information includes some threshold value about target device of required service, event type, target service information and



context value of the event. The user repository store the user information acquired from the profile / history management function. User information includes service authorization, service history, user preference, service contract, user device and current status information. The web server provides SOAP and REST API to user and service and content provider. Also, provide REST API to communicate between the SG and the SSWP. Figure 6 shows service procedure of the SSWP. The service procedure is consists of service publication procedure and service request procedure.

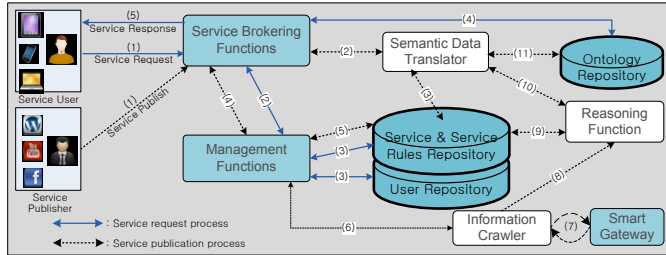


Figure 6. Service procedure of the semantic sensor web platform

The service publication procedure is as follows:

- 1) The service publisher sends service information to the SBF.
- 2) The SBF summarize the information through the service abstraction function and sends abstracted information and service rule information to the semantic data translator.
- 3) The semantic data translator converts abstracted information and service rule information in the form of semantic description. And, stores the converted information in the SR & SRR.
- 4) The SBF requests required information to MF.
- 5) The MF acquire the service requirements and service rules from the SR & SRR.
- 6) The MF find the SGs which have devices that can acquire the necessary information. And, it sends request message includes related gateway location and necessary context information to the information crawler to aggregate related information.
- 7) The information crawler aggregates required information.
- 8) When aggregation is finished, the information crawler sends the aggregated information to the reasoning function.
- 9) The reasoning function gets service information from the SR & SRR. And, generates high-level state information according to the service and aggregated information.
- 10) The semantic data translator gets the high-level state information from the reasoning function and converts to the proper form of semantic description.
- 11) The semantic data translator stores the converted information to the OR.

The service request procedure is as follows:

- 1) The service user sends service request to the SBF.
- 2) The SBF searches the required service through the service discovery function and forwards the service and user information to the MF.

- 3) The MF get the searched service information and user history from appropriate repositories.
- 4) The SBF get high-level state information from the OR.
- 5) The SBF provide high-level state information according to user environment using SOAP or REST APIs.

## VI. CONCLUSION AND FUTURE WORKS

Pervasive computing will increase the size and scope of the Internet through many sensors and devices. And the semantic web technology applies to managing and processing of these huge data effectively. To utilize data on the web generated by sensors of devices or other services efficiently, we proposed the distributed SSW service environment to separate the sensor service domain and the semantic service platform. In addition, the raw output of sensor has to provide appropriate abstractions to the SSWP for minimization of the handling of sensor data. In this paper, we described requirement of SWoT, the distributed SSW service environment, information processing levels. And we proposed functional architecture and service procedures of the SSWP and SG. Our future research will focus on service abstraction and rule generation mechanism. Service abstraction generates service requirement and metadata. And rule generation mechanism provides a simple rule for reasoning and context generation.

## REFERENCES

- [1] A. Sheth, "Semantics scales up: beyond search in web 3.0," IEEE Internet Computing, vol. 15, pp. 3-6, December 2011.
- [2] D. Zeng, S. Guo and Z. Cheng, "The web of things: a survey," Journal of Communications, vol. 6, pp. 424-438, September 2011.
- [3] Z. Shelby, "Embedded web services," IEEE Wireless Communications, vol. 17, pp. 52-57, December 2010.
- [4] A. Sheth, C. Henson and S. S. Sahoo, "Semantic sensor web," IEEE Internet Computing, vol. 12, pp. 78-83, August 2008.
- [5] "WoO: web of objects," ITEA2-10028, <http://web-of-objects.com>, 2009-2011.
- [6] "DiYSE : do-it-yourself smart experiences project," ITEA2-08005, <http://www.dyse.org>, 2009-2011.
- [7] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, T. Cuong, H. Hasemann, A. Kröller, M. Pagel, M. Hauswirth, M. Karnstedt, M. Leggieri, A. Passant and R. Richardson, "SPITFIRE: toward a semantic web of things," IEEE Communications Magazine, vol. 49, pp. 40-48, November 2011.
- [8] L. Atzori, A. Iera and G. Morabito, "Slot: giving a social structure to the Internet of things," IEEE Communications Letters, vol. 15, pp. 1193-1195, November 2011.
- [9] B. Christophe, "Semantic profiles to model the web of things," IEEE SKG, pp. 51-58, October 2011.
- [10] J. Simon, B. Arne and S. Christoph, "Discovery mechanisms for the sensor web," Sensors, vol. 9, pp. 2661-2681, April 2009.
- [11] V. Gupta, A. Poursohi and P. Udupi, "Sensor.Network: an open data exchange for the web of things," IEEE PERCOM Workshops, pp. 753-755, April 2010.
- [12] H. Cuiting, G. M. Lee and C. Noël, "A semantic enhanced service exposure model for a converged service environment," IEEE Communications Magazine, vol. 50, pp. 32-40, March 2012.
- [13] G. Lei, Z. Chunhong and S. Li, "RESTful web of things API in sharing sensor data," IEEE iTAP, pp. 1-4, August 2011.
- [14] H. Ning and Z. Wang, "Future Internet of things architecture: like mankind neural system or social organization framework?," IEEE Communications Letters, vol. 15, pp. 461-463, April 2011.