

Research Article

Web-of-Objects Based User-Centric Semantic Service Composition Methodology in the Internet of Things

Safina Showkat Ara, Zia Ush Shamszaman, and Ilyoung Chong

Department of CICE, Hankuk University of Foreign Studies, Yongin 449-791, Republic of Korea

Correspondence should be addressed to Ilyoung Chong; ilychong@hufs.ac.kr

Received 4 November 2013; Revised 12 February 2014; Accepted 17 February 2014; Published 8 May 2014

Academic Editor: Mohammad Mehedi Hassan

Copyright © 2014 Safina Showkat Ara et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The general goal of the Web-of-Objects (WoO) is to simplify object and application deployment, maintenance, and operation of IoT infrastructures. WoO also aim to provide user-centric IoT service by enabling object virtualization and semantic ontology based service composition. In WoO, semantic modeling of objects plays a distinguished role in achieving interoperability of device and service through semantic ontology model. In this paper, we propose a semantic functional module for user centric service composition in WoO platform. We design an ontology model for virtual object (ViO); it is the physical representation of real world objects. Consequently, we present a user-demand based service composition by showing an use case of WoO. In the proposed semantic based service composition approach, virtual object and composite virtual object are the key entity; this class of objects allows composing services from heterogeneous objects by following a service composition algorithm. This algorithm creates composite service along with the semantic description of that service. The proposed approach of service composition enhances the collaboration of objects as well as services. In the proposed system, user can search a service using natural language through the user interface; on the other hand, the service composition module creates the service by selecting the required objects dynamically.

1. Introduction

The Internet of Things (IoT) [1] portrays a realm where small intelligent objects share data with each other or cooperate in groups to achieve complex goals. However, current devices and communication infrastructures characterized by proprietary protocols and a lack of common standards both on network and application level prevent the realization of this vision. Web-of-Objects (WoO) [2] is aiming a smart distributed application environment that aggregates data from different domains which are currently isolated from each other. In Figure 1, WoO show an overall architecture enabling the integration of any connected object into WoO solution. WoO architecture is divided into two tiers based on their functionality. Local level deals with object connection; at this level, capability discovery, logical execution, and exposure of features in shared formalisms and technologies are the main challenges. Backend level deals with infrastructure that supplies WAN connectivity. WoO backend will provide features enabling the design, execution, and support

of IoT based application [2]. Our implementation of WoO architecture does not meet all of those characteristics. We conceive a scenario where backend is already well structured where physical objects are related through a gateway to the application server.

Henceforth, we are proposing to design semantic ontology based service composition architecture for the Web-of-Objects (WoO), which enables a smart distributed application environment that integrates data from diverse territory currently isolated from each other. To overcome this isolation problem and enable simple development, deployment, and operation of smart distributed applications, an integrated design based on an identical resource-efficient infrastructure, uniform data, and service models and a comprehensive semantic description are needed. In such a world, a semantic ontology offers a practical framework to provide ways to process the vast amount of information produced. From an end-user point of view, the information provided is only meaningful within the scope of some end-user activity.

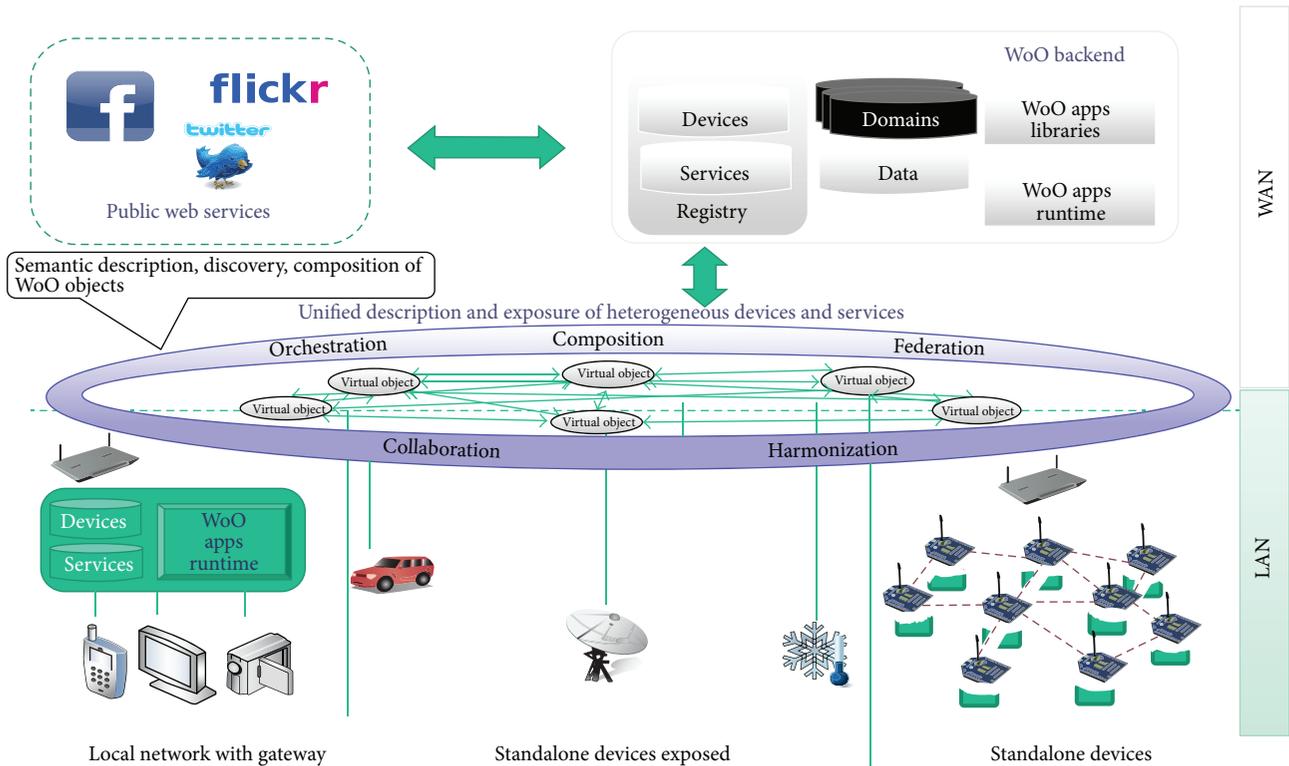


FIGURE 1: WoO overall vision. Virtual objects are the virtual representation of physical objects.

A number of schemes are available to facilitate the composition of services. Such programs enable a user to manually specify a composition of programs to perform some tasks [3]. WoO framework can benefit from structured models that detail various concepts and provide abstractions of the components and their attributes. The main abstractions concepts underlie the WoO domain and describes the relationships among device, object, resource, and service. The primary tenet of the WoO is an extension of the web in the physical world, to involve interaction with a physical entity in the ambient environment. Service composition and modelling in WoO facilitate for us dynamically selecting actor objects and service schedule and setting conditions for service execution [2].

In this work, service composition is accomplished by harmonization of objects; the main required agent to compose a service is the semantic description of objects. More specifically, through the use of in-depth semantic description, intelligent mechanisms can be imposed to dynamically composed service. For example, a service can be defined from the service description of another service which is done through collaboration and federation among multiple types of objects on the web. Objects will be defined with sensors, devices, contents, functional processes, underlying environments, media, and so forth. Objects are collaborated and federated in a semantic ontology to provide smart emergency at home and in a shopping mall. Here, service is offering all necessary functionalities to interact with the objects and related processes using the web interface. Device refers to real device in the physical universe and objects

refer to any virtual representation of physical entity in WoO domain. Resources are accessed by services which provide functionality to gather information about the entities they are affiliated with or manipulate physical properties of their associated objects. The object virtualization is attained with those various objects that are related to the model. With the objectification, new services are created according to user-centric right after the other services with the combination of various objects.

Web-of-Objects is the implementation approach of the Internet of Things and the Web of Things, where semantic ontology model is particularly well suited to improve interoperability among objects. In this paper, our aim is to propose a semantic ontology based service composition architecture along with a service composition algorithm in WoO. We also present a scenario based use case by considering the proposed architecture.

The remaining sections of this paper are organized as follows. Section 2 discusses related work in the sphere of web service composition and semantic ontology based service composition. Section 3 presents the proposed service composition approach along with detailed procedure, algorithm, and service composition rules. In Section 4 scenario study and prototype implementation is described and, finally, Section 5 concludes the paper with future direction.

2. Related Work

Web service composition can be categorized from manual to automated, where manual web service composition

involves user in every step of the process on the other hand automated web service composition allows very limited user interventions for defining user requirement for desired composite service [4–8]. Automation capability of web service discovery and the structure of the web service composition are the measure of this classification. Manual approaches follow standards, which define complex processes implemented through composite services like BPEL4WS [9]. Semiautomated approaches concern only parts of the process. For instance, they automate only the selection of certain web services that implement a manually defined composition scheme [10]. Finally, fully automated approaches entail automation of both the composition plan and the discovery of suitable web services. As the number of available services continuously increases over time, automation of the composition process constitutes the only solution able to efficiently manage the vast volume of this domain. Other advantages include scalability, flexibility to detect changes in atomic service definitions, and dynamic handling of service failure/unavailability.

Using the semantics may considerably facilitate the representation method and enrich engagement of intelligent techniques. Traditional works, such as the Causal Link Matrix [11], offer a solid experience for the semantic web service composition through Artificial Intelligence techniques. It includes all pivotal associations between semantic web services and uses them to articulate effective compositions.

Another approach for automated web service composition is attempted through planning as model checking, with the alteration of the MBP system [12]. It produces a description of the desired composite service in BPEL4WS. This approach copes with issues such as nondeterminism, partial observability, and extended goals. However, semantic information is not utilized during composition, while scalability is questionable. Another approach in [13] represents atomic services to perform service composition. However, current standards need to be extended to comply with this approach, though the result is not encouraging in terms of scalability.

Scalability and heterogeneity related challenges of IoT components and the extremely dynamic nature of the network topology have been addressed in [14, 15]. They have provided overview of a service-oriented middleware solution that addresses the mentioned challenges using semantic technologies to provide interoperability and edibility. They give emphasis on modeling a set of ontologies that describe devices and their functionalities and thoroughly model the domain of physics. The physics domain is indeed at the core of the IoT as it allows the approximation and estimation of functionalities usually provided by things. Those functionalities will be arrayed as services on suitable expedients through middleware. In this work, they have built a global ontology for the Internet of Things. Things is a real world concept and functionalities of things is real world approximation and these features are modeled as ontology; in this work, the suggested ontology is at the core of a service-oriented middleware for the Internet of Things which is scalable, edible, and interoperable between heterogeneous IoT components.

Though the current web service technology domain is not matured enough to properly own dynamic services, still there are some proposals to empower dynamic web service composition. Nevertheless, according to our knowledge, some of these proposals address both functional and nonfunctional properties of web services to improve the service composition method. Reference [16] defined the key challenges handled in semantic web services, that is, dynamic composition and optimization based on nonfunctional properties. To this end, we describe a methodology for the semantic service composition of web based services for pervasive computing.

A real world physical object can be described in a manner that facilitates its uses and accessibility in the virtual world, any object is the virtual representation of real world physical objects such as sensor or other physical things [17, 18]. A temperature sensor is an object on the other hand shoppers and guard are different kind of objects and all of the objects can be represented as virtual object. Here, they created virtual object which is related to each other by semantic ontology. Object formation information is stored in RDF. Creation of new virtual objects is done with tagging a new identity which is actually the combination of existing virtual objects for a specific service. Each new virtual object will generate a unique identity, that is, ViO-ID. Newly created virtual objects are semantically interoperable ViOs that are combined to offer the services requested by the end user. Features and function of source ViOs will be inherited; the new virtual object is created. This association and aggregation are made for the provision of composite services by preserving the orchestration with other related objects. These new objects are proficient to be reused. Lifetime of new virtual objects depends on the necessity of application. In this thesis work, they did not consider the new virtual object to be stored; that is, it will be destroyed after satisfying the application. Compared to our work, we are storing new virtual object information in an object repository. After execution of service, the object formation information, object mapping to create a service, information of newly created service will store in RDF file which is called object metadata file.

3. Proposed Semantic Service Composition Approach

In this section, we present a semantic-based dynamic service composition architecture and service composition procedure in the context of WoO. In the proposed approach, user's request for a service is not separately named by the user; the users write their own requirement in web interface as a normal language, like searching something in a web search engine. Service composition system comprises all the required services in the virtual object form according to a user's request. User query is represented to relevant semantic annotation and form a semantic graph. The semantic functional module provides a semantic description and ontology that are used to search services. Instead of using any service discovery mechanism, the service repository and its semantic description are used to detect a service.

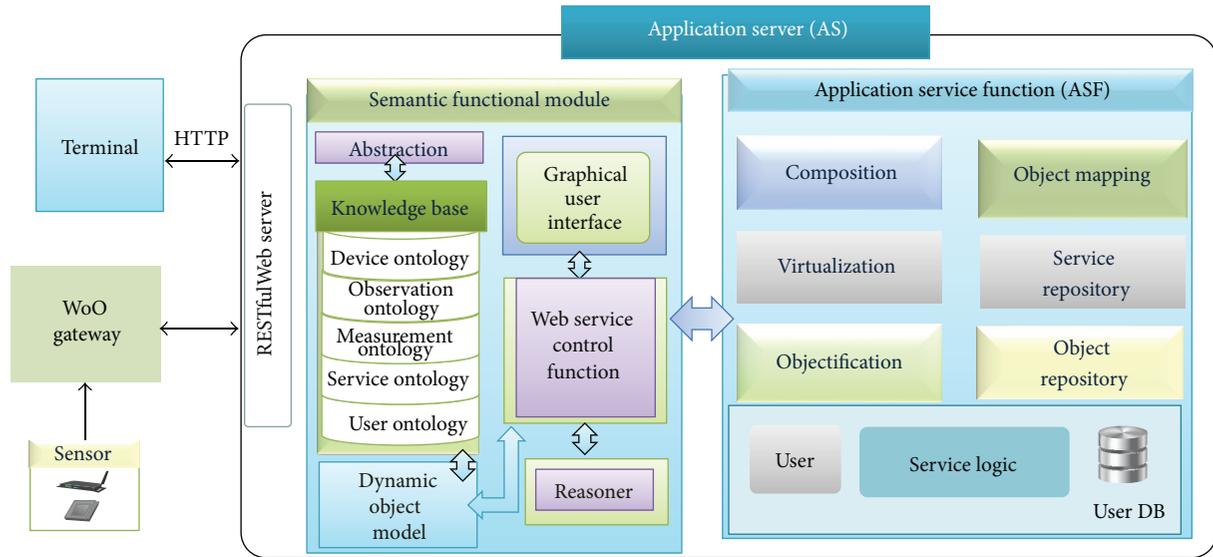


FIGURE 2: WoO based semantic service composition architecture. User can only connect to the AS through application or web.

Figure 2 depicts the components of the application server for service composition in WoO. Semantic Functional Module (SFM), and Application Service Function (ASF) are the main functional parts in Application Server (AS). Sensor devices collect data from the real world and send it to WoO gateway; WoO gateway, on the other hand, is capable of relaying environmental data to the application server (AS). The gateway maintains a device profile for sensors attached in the gateway. The terminal is the smart phone or web based client module which will contain application for user; the user application will help users to access the WoO service infrastructure and to get the notification and alert message. The semantic functional module is the core component to support the description of the physical and processing structure of objects. The overall architecture of our proposed system is shown here; our design consists of semantic functional module (SFM) and application service function (ASF). Once the sensor data are available through a gateway, we need to identify the semantics of this information. Therefore, our architecture consists of sensor and other related ontologies. Semantic ontology module maps raw data to the corresponding domain ontologies.

The XML data is renewed as RDF/XML file format and construct ontology applying low level observation and high level abstraction sequence. The data can be stored in database and it can be made accessible through the SPARQL interface. Web services can be built to accept SPARQL queries and return results in a standard format. The graphical user interface can be developed by using JSP/Servlets; here, web services can also be added to interact with the user. We have chosen JSP/Servlets based GUI, for example. The framework JENA that is also a component of this architectural module is also built in Java and this is the most advanced framework supporting semantic web technologies. The GUI will interact with the web service control function and send its request

to web service control function. The web service control function consists of business logics and specific template definition of the methods and variables in a peculiar kind of object to interact with the component of SFM. All the requests from the GUI will intermingle with this web service control functions only. These are responsible to obtain the request from the GUI and forward them to SFM; it also forwards the results sent from SFM to the GUI. The reason behind this service control function is to bring the benefits of multitier system. The GUI can be replaced with new technologies. The dynamic object model consists of JENA framework classes. All the control functions can act together through the model. It is the framework that helps to build semantic web applications. JENA also has a rule based inference engine. Execution of SPARQL query with JENA APIs is done here. Models can be made either in memory or in a file or in a database. In memory model means a model in RAM. It is not possible to construct a large memory model due to limitations of RAM size. A model can also build on physical storage like hard disk. A model can also be built into a database. JENA supports all major databases like Oracle, Microsoft SQL Server, MySQL, PostgreSQL, and so forth.

Reasoner helps to infer new RDF statements from existing base assertions and optional ontology definitions. With JENA, we can plug in different types of reasoners. We can say that reasoner is an inference engine of JENA. Ontologies capture the concepts and the relationships between the concepts describing the different resources available in the system. The knowledge base contains ontologies for the device, measurement, observation, service, and user; ontology concepts are related to the domain concepts. The developed web application, in this paper, is based on making use of semantic web technologies to demonstrate the benefits of such technologies by dynamic service composition in terms of different situations.

3.1. Proposed Semantic Service Composition Procedure. The service composition request is sent from the web interface to the service composition module. we can select predefined condition and relation among objects from the web interface; a service composer will define the service profile and model with the specific object; scheduler leads the model for scheduling of the service with special features. A repository will contain the dynamically composed service for execution. At the scheduled time the application server invokes the service for execution.

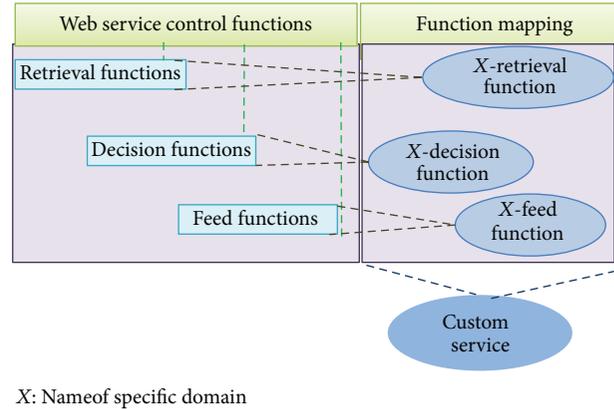
Figure 3 shows the functions involved in service composition. The web service control functions are categorized into three basic types on the basis of their main role, retrieving, decision making, and feeding. Each category contains distinguished functions for every domain in semantic ontology model. Though the participation of object in a custom service is dynamic, each object has its related control function to retrieve data and semantic information from the semantic module and impose decision which matches only with the domain.

Objects from virtual world take part in service composition. Depending on atomicity objects are mainly classified as a virtual object and a composite virtual object. The service is also classified as a service and composite service. Composite services are composed of service and other objects. Each individual object has its corresponding semantic description, whether it is service or any other object. Before going through the composition process, each object must qualify with basic semantic annotation specified by the conforming module. The object is the virtual representation of physical objects.

A component of the virtual world can be an atomic object which poses existence in virtual space such as services which are types of objects.

Hereafter, in Figure 4, we illustrate how service composition procedure accomplishes the proposed architecture. User make queries to the system using the user application (app/web browser); a semantic interoperable translation mechanism is used to translate user request by semantic reasoning.

Figure 7 shows WoO semantic ontology mapping for service composition in shopping mall. User requests for whatever service is recognized through query analysis and service selection. Once the petition is specified by query analysis, service selection and decision making will be done to compare query keyword with existing information of service; if any satisfactory percentage of similarity was found between keyword and service information, reuse of existing service is decided; in order to be executed, match service information is sent to object mapping. In case of failure to search a similar service, a list of object information is sent to object mapping to create a new service. From the object, repository object mapping function retrieves similar service. The service repository contains data for each service and the service itself; to interact with other components, an identifier is used for each service as foreign key. The service repository maintains a service context file, which contains identifier, relation, and list of other services that take part to compose a composite service, domain, and range of the service.



X: Name of specific domain

FIGURE 3: Web service control functions module and mapping for custom service.

The service context files, containing semantically enriched information about the service, are stored in resource description framework (RDF). Listing 1 shows a sample service context file with one reference service.

3.2. Service Composition Algorithm. The flow chart of service composition algorithm is shown in Figure 5, In this subsection, we propose a service composition algorithm which is the root of our proposed service composition approach (Listing 2). In case of failure to search suitable services, an optimal list of objects will be suggested by the algorithm. The object mapping function will follow the algorithm to search similar services for composition. The user will provide the keyword in natural language including the method of choosing similar service; there are two options to select service: by selecting either the most similar or the most liked one among the similar services. Finally, similar services will be set out after comparing with acceptance point. The user can include additional objects to compose service. The semantic functional module will contain service information. In the composition procedure, service repository will gather the semantic description to search for the most similar service. The keyword will match semantic descriptions of each service and with consider the matched one for temporary storage. The reduced list will store the matched services in temporary storage to calculate the similarity with the keywords. The most similar services will be considered and stored for future writings. Likewise, if the user wants to include more objects, then they can search in the object list of object mapping function following the same algorithm. From the semantic functional module object, information will be conveyed to map each object to search for more similar objects. The keyword will match with the semantic descriptions of each object and will consider the matched one for temporary storage, after matching, a reduced list of object is created and a calculation will be done to find the similarity with the keywords. The most similar services will be stored for future compositions.

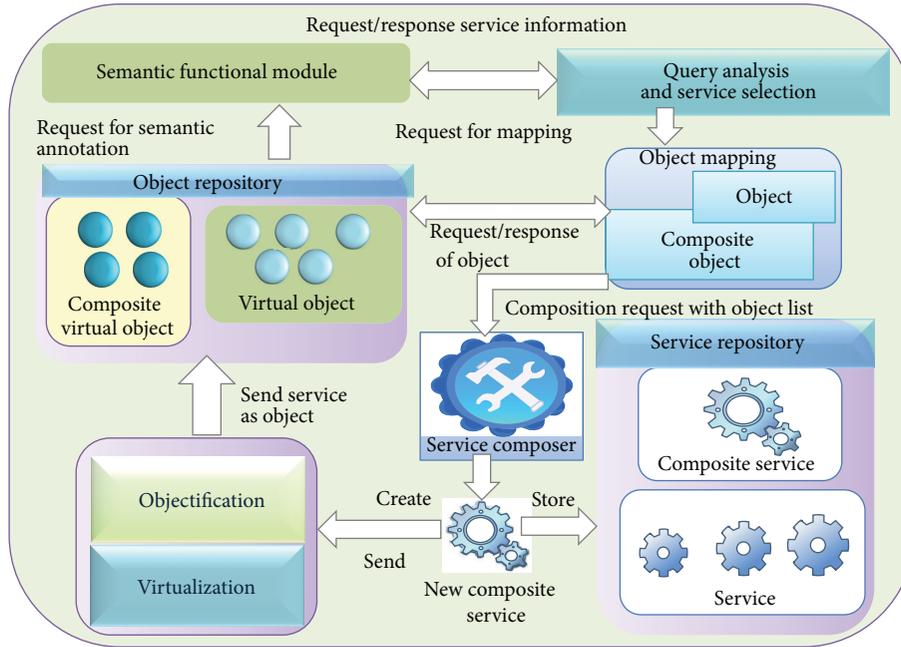


FIGURE 4: Dynamic semantic service composition procedure.

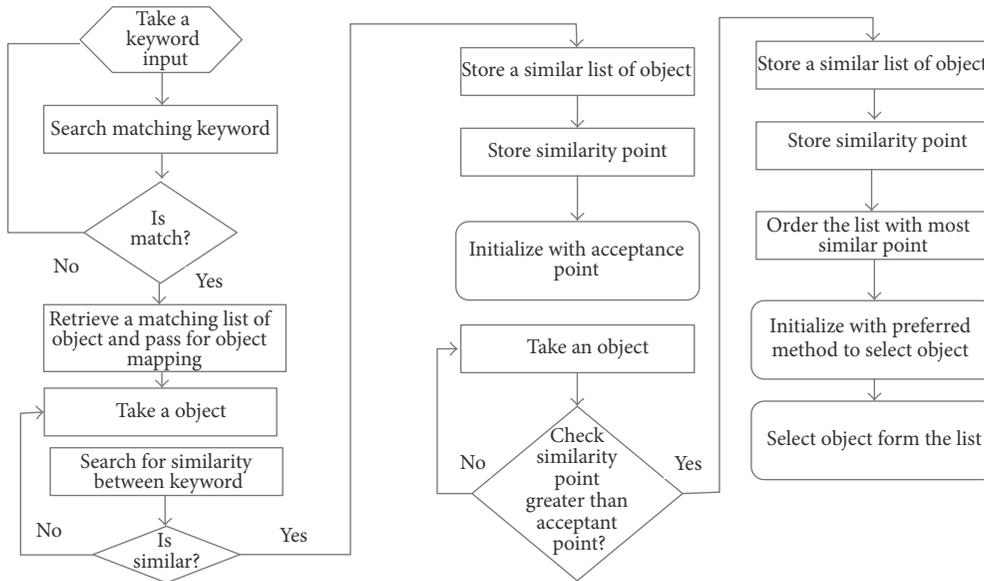


FIGURE 5: Dynamic object selection flow chart.

4. Scenario Study and Prototype Implementation

A semantic representation of both functionality and interface of a service allows for application scenarios in the Web-of-Object by enabling features such as service discovery and validation of interface consistency. The goal is to create ontologies and develop a concept to use this semantic representation in a Web-of-Objects. The WoO ontology is based on the concepts of systems, processes, and observations. It supports

the description of the physical and processing structure of objects. Objects are not constrained to physical sensing devices; rather an object is anything that can provide the value of an occurrence, so a device or computational process or combination could play the role of an object.

Figure 6 shows the WoO semantic ontology model. It is the conceptual model of virtual objects and their relation; it shows a relationship between the objects according to their action and behavior. The model is produced following the principle of minimal ontological assurances to make it

```

CompositeService>
<Service_Profile><compositeService Profile>
<Service_primary_ID><1.1.0>
<Service_Reference_ID><1.2.0>
<Object ID>1.1.4>
<Service_Name><baby_search_service>
<Service_Description><this service is composed to search baby inside the shopping mall>
</CompositeService>

```

LISTING 1: Sample service context file.

```

(1) Input: keywords for service  $S_1, S_2, \dots, S_n$ , choosing method (most similar one or most liked).
    Acceptance point, including additional object variable
(2) Output: composite service, In case of failure, suggest objects for optimal service composition.
(3) Initialize array A
(4)   For all service S do
(5)     If keyword matches in service
(6)       Store in A [with service name and keyword]
(7)   End for
(8) Initialize array sim
(9)   For each array do
(10)    If similar then
(11)      Store in sim [with service name and similarity point]
(12)    End for
(13) Initialize variable to acceptance point
(14)   For each array sim do
(15)     If service content greater than acceptance point then
(16)       Store in list with similarity point and name
(17)     End for
(18) Sort Sim Array
(19) Considering choosing method a or b
(20) Select services from sim
(21) Initialize array obj
(22)   For all object O do
(23)     If keyword match in object
(24)       Store in obj [with obj name and keyword]
(25)     End for
(26) Initialize array sim2
(27)   For each sim2 do
(28)     If similar then
(29)       Store in sim2 [with object name and similarity point]
(30)     End for
(31) Initialize variable to acceptance point
(32)   For each array sim2 do
(33)     If service content greater than acceptance point then
(34)       Store in list with similarity point and name
(35)     End for

```

LISTING 2: Pseudocode for service composition Algorithm.

reusable for a variety of application domains. It presents a set of classes and relations based on the whims of the sensor and observations.

Hence, we demonstrate a user case of the proposed service composition approach in WoO. In a semantic ontology base shopping mall system, there is no service discovery

function. Service discovery is accomplished through the semantic functional module. To search a service, the end user will make the query.

End users make query from the web interface to search a service in the system using general sentence. The user will not know any details about the services. Service details are hidden

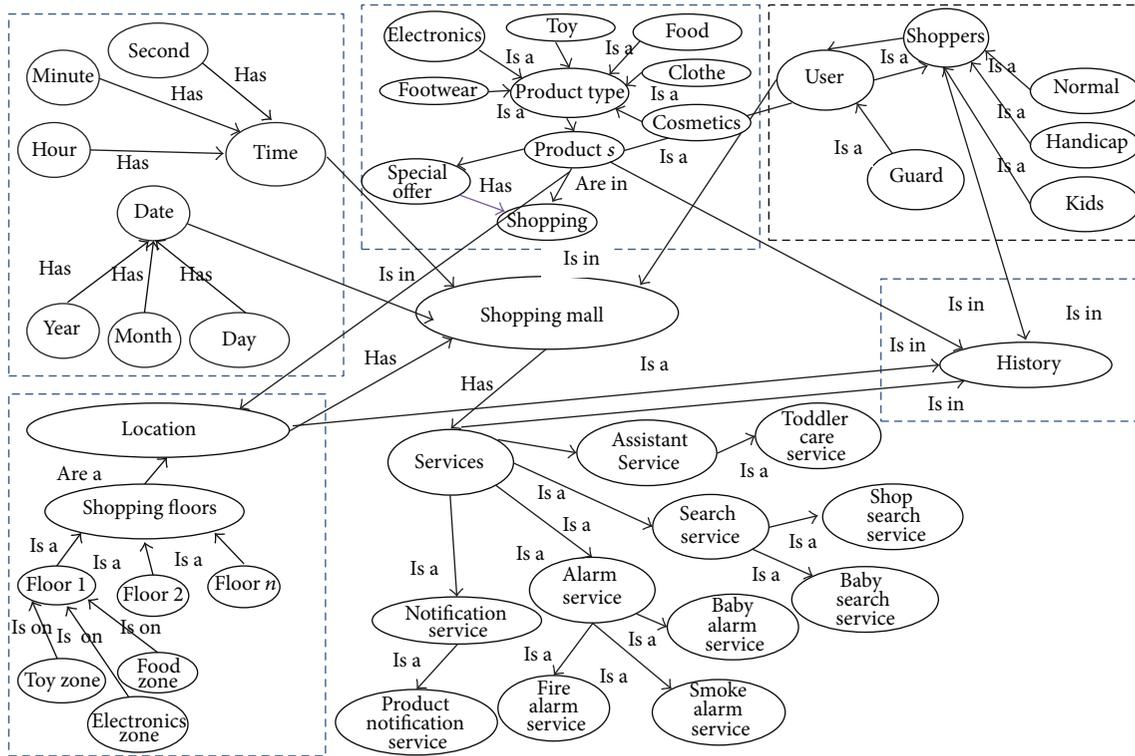


FIGURE 6: WoO semantic ontology model for shopping mall.

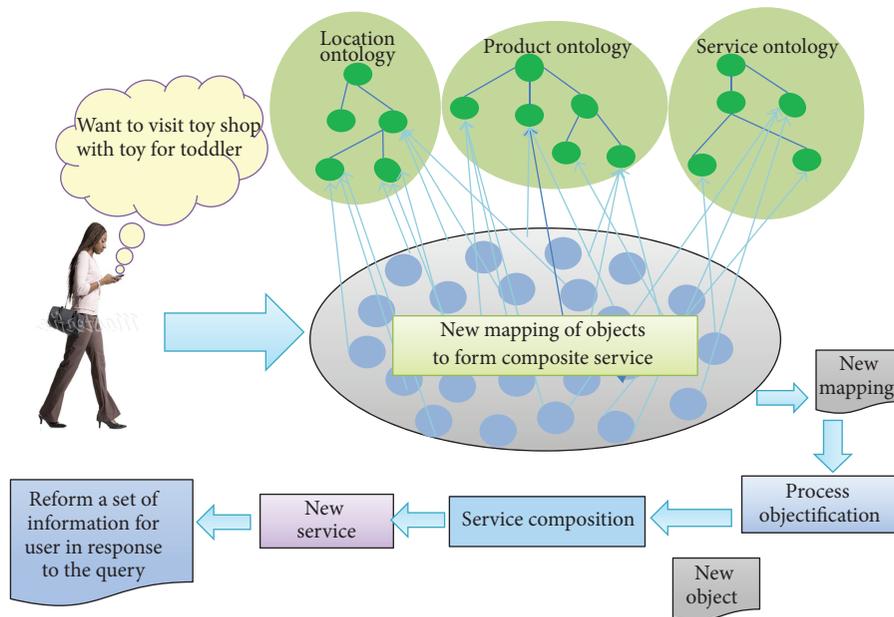
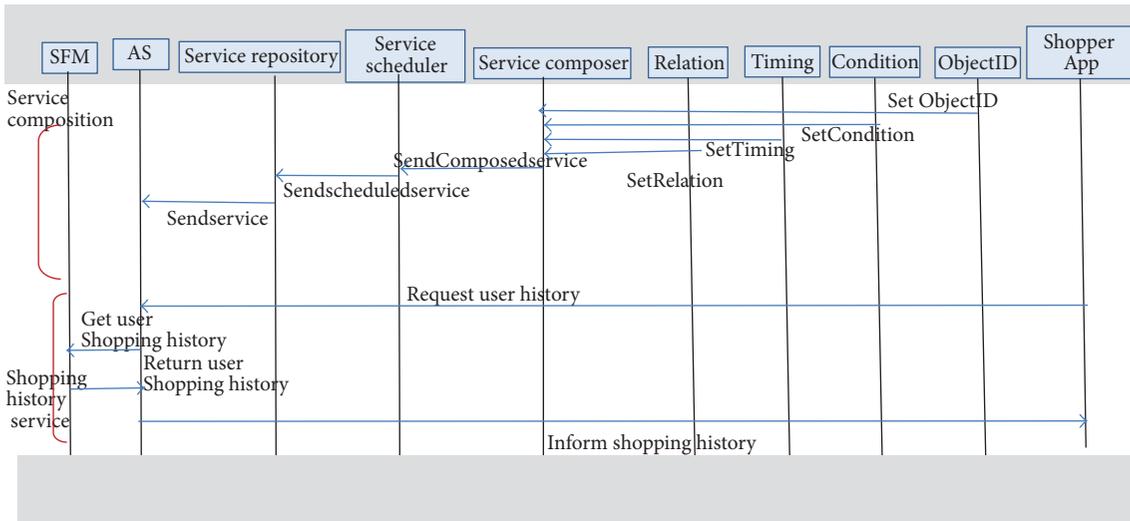


FIGURE 7: WoO semantic ontology mapping for shopping mall.

from the end user; semantic functional module will gather similar information related to the keyword. The semantic functional module will transmit the best match service to ASF. ASF will reform a new service with new mapping of objects. For example, When a shopper enters in a semantically enriched WoO based shopping mall and search toyshop to

buy toy for toddlers then shopping mall management-system always have the services ready to perform search operation to locate the desired toy for toddlers. After receiving the user query, analysis will be done to search the similar service related to the query. If the relevant service is not available to answer the purpose, from the semantic information of



AS: Application server
 SFM: Semantic functional module

FIGURE 8: Service composition sequence diagram for shopping mall scenario.

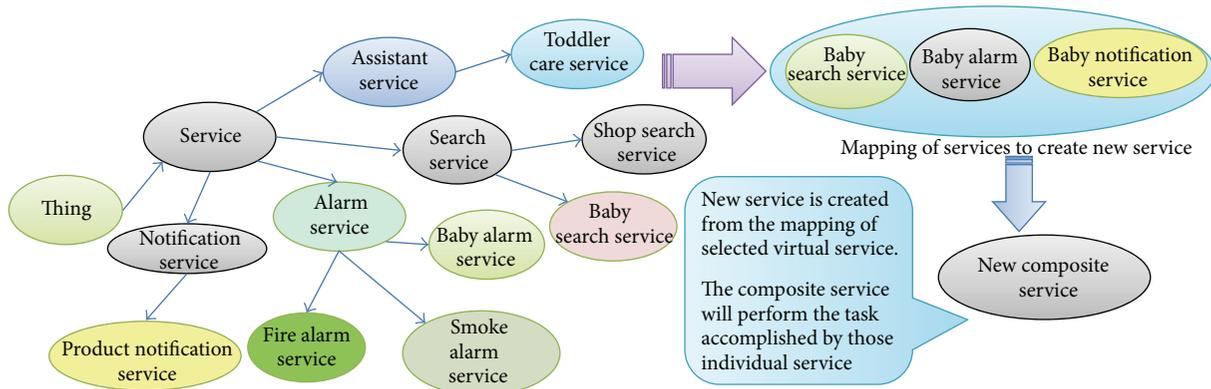


FIGURE 9: WoO semantic ontology mapping for service composition.

objects, a list of objects will be suggested to create new service and reform the service as a physical object. The detailed backend operation will be unknown to the user. The women will receive a suggestive place where they can find shops with toy for toddlers. This search information will be stored in history for any further query.

Figures 9 and 10 demonstrate another service composition scenario and the related web interface, where a mother lost her child in the mall. The mother queries or describes the situation using user application. To answer the query of the user, first list of existing services will be studied. Baby search service, baby alarm service, and baby notification service are the services related to baby lost and found queries. If the same user came in the shopping mall, the shopping mall system will provide some suggestions for the toddler. To provide a viable service to shoppers, there are some well-built services to support users. Among the services and their semantic information, a new form of service can be made. The newly

created service can work individually, aggregation of those services can perform a better service for the user. Thus, baby search service will search the location of the child. A hand band will be tied in baby’s hand which will auto ring when a mother queries for a lost child. A nearby person will be alerted from the ring that a lost baby is near them. The retrieving location of the baby notification service notifies the woman about the baby’s exact location within the shopping mall.

This way, a 3-step searching service is consummated. A system prototype was developed to explain the operation of the proposed system and to assess the feasibility of the scheme. The ASF and SFM comprise all the system components. Shopping mall ontology is built up using a protégé OWL editor; SFM and the ASF modules are developed in the Java programming language on an application server with Intel processor core i7 3.4 GHz, 8 GB RAM. JENA API is used for semantic data manipulation and JENA integrated reasoner for inference functionalities. SPARQL and MySQL

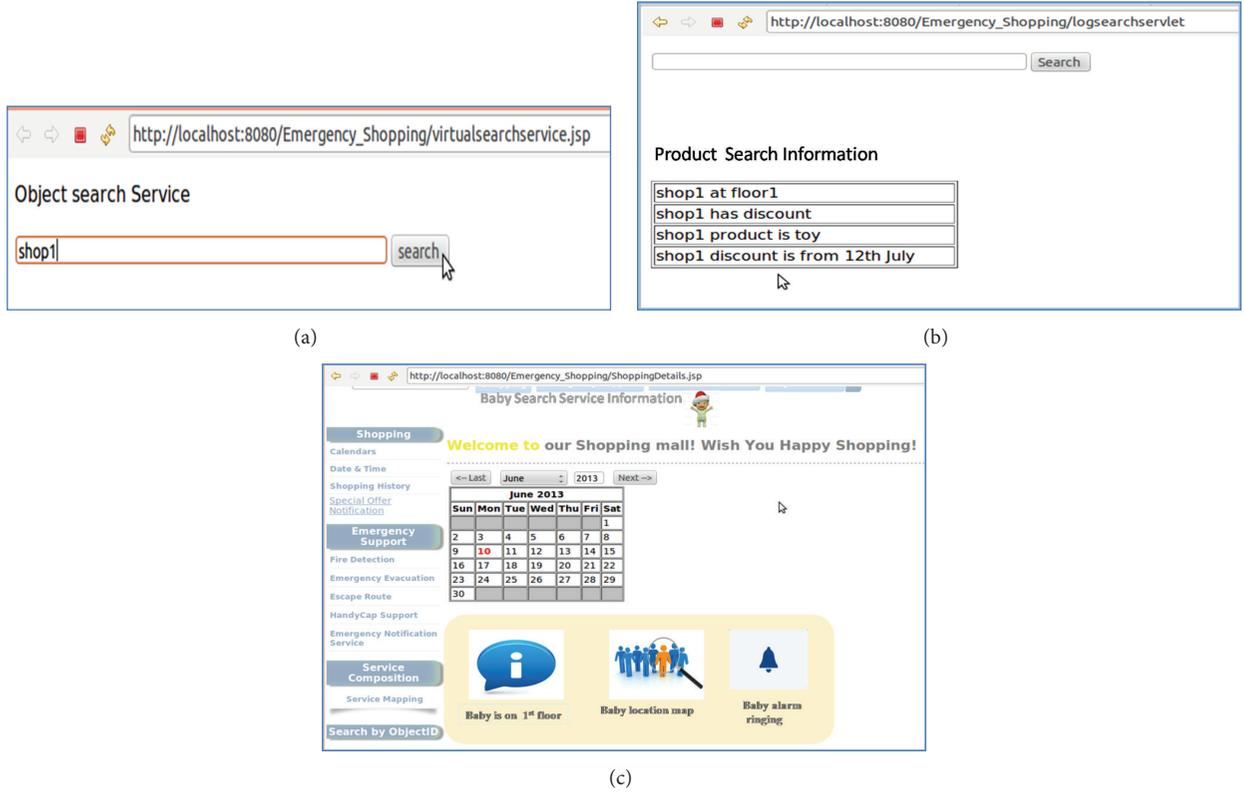


FIGURE 10: (a) Web interface to query for a lost child. (b) Web interface to query shopping history. (c) Web interface showing lost baby information.

TABLE 1: Service composition rules.

Rules	Scenario	Meaning
OP: $\text{Service}(?s) \wedge \text{hasdependency}(s?, \text{false}) \wedge \text{object}(?s) \rightarrow \text{compositeservice}(?s)$	Objects participation	Except composite object, in composition, an object can participate without any dependency.
CSP: $\text{Service}(?s) \wedge \text{hasdependency}(s?, \text{true}) \wedge \text{object}(?s) \rightarrow \text{compositeservice}(?s)$	Composite service participation	The vice versa composite object cannot participate without its base object.
OPR: $\text{isservice}(s?, \text{true}) \wedge \text{object}(?s) \rightarrow \text{compositeservice}(?s)$	Object participation restriction	To compose a service with objects, one object must be in the form of service.
AOP: $\text{Object}(?s) \wedge \text{hasadditionalObject}(?s, \text{true}) \rightarrow \text{compositeservice}(?s)$	Additional object participation	Other objects can participate in service composition with additional features.
SSC: $\text{hasAverageAcceptancepoint}(?s, \text{true}) \wedge \text{isservice}(?s) \rightarrow \text{isselected}(?s, \text{true})$	Service selection criteria	In case of service selection and searching, one service can be selected with an average acceptance point.

are also used for data manipulation. The ontology for the service composition was designed in protégé. To work with protégé, first, we have to specify the domain and range of the ontology and important terms that can be the domain. secondly, have to separate concepts, properties and relations. Concepts are considered as class and organized in class-subclass and relation diagram. For example, location node is a class which has subclasses of shopping floor. Defining properties like having measurements has sensor node, which is used for relating different class. We have to determine data properties to represent data attributes of the class.

5. Conclusions

We have presented a semantic service composition architecture, method, and algorithm in the context of WoO. In our proposed approach, ontologies describe the relation among objects, services, and rules to compose new services dynamically. Hence, semantically linked virtual objects are the virtual representations of real world physical objects; this object virtualization is required to ensure service orchestration and dynamic service composition by reusing and sharing virtual object's properties and attributes.

The proposed semantic module has layered structure analogous to semantic web stack, even though it emphasizes mainly on the ontology layer and logic layer. Services are described in the knowledge-base of the ontology layer, and rules are made to perform reasoning tasks for automatic service discovery and processing. We have considered shopping mall scenario as service composition use case in WoO infrastructure. The proposed system is capable of processing natural language of the user, which allows users to determine their requirements for inspection and repairs. Furthermore, we implemented an object assembler and a service composer for the application to match services. Figure 8 shows the sequence diagram for service composition. We have also construct rules using protégé, Table 1 contain rules and their detail description.

We are highly motivated to evaluate our work by comparing it with existing service composition approaches in our next study. We also intend to take the complete execution of all the functional entities of WoO that are planned but not yet implemented. Additionally we are developing user application in order to guarantee the system's complete functionality which is also needed to evaluate the performance in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by Korea Evaluation Institute of Industrial Technology (KEIT) funded by Ministry of Trade, Industry and Energy (MOTIE, Korea) [10047233, Development of Smart Home Web of Objects Architecture].

References

- [1] "The Internet-of-Things Architecture (IoT-A) project Sep 2010-Nov 2013," <http://www.iot-a.eu/public>.
- [2] "Web-of-Objects (WoO)-ITE2 project Jan 2012-Dec 2014," <http://www.itea2.org/project/index/view?project=10097>.
- [3] D. Pfisterer, K. Römer, D. Bimschas et al., "SPITFIRE: toward a semantic web of things," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 40–48, 2011.
- [4] J. Rao and X. Su, "A survey of automated Web service composition methods," in *Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC '04)*, pp. 43–54, San Diego, Calif, USA, July 2004.
- [5] B. Srivastava and J. Koehler, "Web service composition-current solutions and open problems," in *Proceedings of the Workshop Planning for Web Services*, 2003.
- [6] N. Milanovic and M. Malek, "Current solutions for Web service composition," *IEEE Internet Computing*, vol. 8, no. 6, pp. 51–59, 2004.
- [7] A. Bucchiarone and S. Gnesi, "A survey on service composition languages and models," in *Proceedings of the International Workshop Web Services Modelling and Testing (WsMaTe '06)*, 2006.
- [8] S. Dustdar and W. Schreiner, "A survey on web services composition," *International Journal of Web and Grid Services*, vol. 1, no. 1, pp. 1–30, 2005.
- [9] S. Thatte, Ed., "BPEL4WS (Version 1.1)," 2003, <http://www.ibm.com/developerworks/library/-specification/ws-bpel>.
- [10] F. Casati, S. Ilnicki, L. Jin et al., "Adaptive and dynamic service composition in eFlow," in *Proceedings of the 12th International Conference Advanced Information Systems Engineering (CAiSE '00)*, 2000.
- [11] F. Lecue and A. Leger, "A formal model for semantic web service composition," in *Proceedings of the Leading the Web in Concurrent Engineering: Next Generation Concurrent Engineering*, pp. 385–398, 2006.
- [12] M. Pistore, A. Marconi, P. Bertoli, and P. Traverso, "Automated composition of web services by planning at the knowledge level," in *Proceedings of the 19th International Joint Conference Artificial Intelligence (IJCAI '05)*, 2005.
- [13] D. McDermott, "Estimated-regression planning for interactions with web services," in *Proceedings of the 6th International AI Planning and Scheduling*, 2002.
- [14] S. H. Kang and S. K. Lau, "Developing ontology revision framework: a case study on the use of the coherence theory for semantic shopping mall," in *Proceedings of the 4th IEEE International Conference on Management of Innovation and Technology (ICMIT '08)*, pp. 508–513, Bangkok, Thailand, September 2008.
- [15] S. Hachem, T. Teixeira, and V. Issarny, "Ontologies for the internet of things," in *Proceedings of the 8th Middleware Doctoral Symposium (MDS '11)*, December 2011.
- [16] H. Knublauch, "Ontology-driven software development in the context of the semantic web:an example scenario with Protégé/OWL," in *Proceedings of the 1st International Workshop on the Model-Driven Semantic Web*, 2004.
- [17] P. Vlachas, R. Giaffreda, V. Stavroulaki et al., "Enabling smart cities through a cognitive management framework for the Internet of Things," *IEE Communications Magazine*, vol. 51, pp. 102–110, 2013.
- [18] Z. U. Shamszaman, *An implementation of web of objects (WoO) based context aware emergency management systems [M.S. thesis]*, CICE, HUFU, Seoul, South Korea, 2013.

