International Journal of u- and e- Service, Science and Technology
Vol.9, No. 1 (2016), pp.161-170
http://dx.doi.org/10.14257/ijunesst.2016.9.1.18

# An Ontology Model for Smart Service in Vertical Farms – An OWL-S Approach

Saraswathi Sivamani, Hong-geun Kim, Myeongbae Lee, Jangwoo Park, Changsun Shin and Yongyun Cho*

*Dept. of Information and Communication Engineering,*
*Sunchon National University, Suncheon, 57922, Republic of Korea*
*{saraswathi, khg_david, lmb, jwpark, csshin, yycho}@sunchon.ac.kr*

***Abstract***

*Recently, the evolution of ubiquitous computing has brought a breakthrough in network access and web based services including the agricultural field, which is integral to human living. In a ubiquitous vertical farm environment, context aware services display the context information in selecting the appropriate web service by identifying the state of the user and its surroundings. In this web enabled environment, establishing a context aware system for the vertical farm without the understanding of domains gets complicated especially when shaping with the avails. The semantic web should enable users to locate, select, compose and monitor web-based services automatically. For a successful execution of such services in semantic web, the service needs to be grounded with the corresponding WSDL. To resolve these issues, our work includes the development of an OWL-S based ontology model to define the relationship between the domains and add classes needed for the model in every aspect of the service oriented system. Compared with any other semantic web service, OWL-S is more suitable for the Vertical Farming System in the ubiquitous computing.*

***Keywords****: Ubiquitous Computing, Semantic Web, OWL-S, Ontology*

## 1. Introduction

Information across the web has grown enormously in this emerging pervasive environment, which makes it difficult for a effective communication. Web service brings aid for such situations. The programming language running on various platforms can use web services for interoperable machine-to-machine interaction [1] over a network, thanks to the use of XML. Web service description language (WSDL) [2] is a well-known XML-based service that helps service providers to access the system in a simpler form, regardless of the underlying runtime implementation. For non- technical people or in an automated environment, web services serve as a unit handling business processes without thinking about the application.

Alongside this, semantic web technologies have been proving the means of annotating resources and services resulting in the new standards that enable true semantic interoperability for services [3]. Semantic web should provide greater access to the services on the web. In order to achieve a high degree of automation in the semantic web services, an Ontology Web Language for Services (OWL-S) [4] specification, which defines a set of ontologies through a semantic description of services, are created.

The service description consists of three ontologies such as Service Profile, Service Model,and Service Grounding. A lot of research is underway in the service-oriented technique in the agricultural ubiquitous environment [5, 6]. As the vertical farm system is acclaimed for automation, the inclusion of semantic web services assures an effective

---

* Corresponding Author

result. This paper presents an ontology for enabling semantic web services in a vertical farm system with the OWL-S approach. The concepts are identified in the vertical farm environment, keeping the services as a key.

## 2. Background

Agriculture in urban areas [7] have become a new trend, owing to vertical farming with pervasive computing. As the main significance of vertical farming is a complete automation without human interaction, the relationships are well defined with the help of Ontology. As the relationship revolves around the concepts, Gruber et al. [8] termed Ontology as a "formal, explicit specification of a shared conceptualization". Compared to the other context models [9], the ontology model has more advantage in building a context model, which is more popular for the semantic interoperability to exchange and share the context knowledge between the interfaces.

Many existing research in vertical farming have introduced the ontologies for a better communication of the context information. Most of the research focus on the automation of services, most of which are focused on the context model [10, 11]. Others are primarily centered along the discovery enhancement in service annotations [12]. Web service discovery can be considered as the main focus in locating a suitable web service to accomplish the given purpose. More or less of the popular framework used in semantic web services is WSMO, OWL-S and WSDL-S. Compared to other web services [13], we find OWL-S to be more suited to the Vertical farm's ubiquitous environment.

OWL-S is the ontology web language for semantic services. OWL-S is an enhancement of the DAML-S [14]. A Major advantageous feature of the OWL-S is that it has automatic web service discovery, invocation, composition and interoperability. The OWL-S ontology has three sub ontologies, service profile for advertising and discovering the service, service process that explains the service and grounding, and provides the integration of services. Despite these normal functions, to perform the service discovery, certain input, output, precondition, and effects are needed in the OWL-S. For this reason, similar parameters can be created in the OWL and directly linked with the Service class in the OWL-S ontology. Many such successful approaches have succeeded at achieving semantic interoperability seamlessly [15]. Adopting the overture, the domain specific ontology is made with the main concepts categorized together, which is described in more detail in the succeeding segment.

## 3. Ontology Model for Vertical Farm

The Ontology for vertical farm was designed mainly focusing on the service-based vertical farm without any human intervention. The OWL-S based approach, which helps the services in providing richer semantic specifications, is used in vertical farm ontology. To build a context aware model for the services in the vertical farm environment, the first step taken was to identify the concepts. The concepts are identified using statistical measures [16].

 

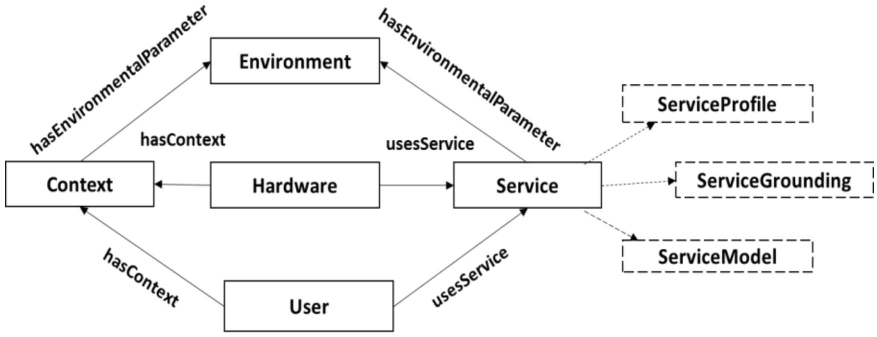


**Figure 1. Top Level Ontology for Vertical Farm Extending the Upper Ontology of service Class**

After analyzing the common dependency among the domain, the concepts are grouped as follows. The concepts are Context-Based, Service-Based, Environment-Based, Hardware-Based, and User-Based concepts. Figure 1 describes the top-level model of the Vertical Farm Ontology, in which Service-Based concepts extend the upper ontology services of OWL-S.

### 3.1. Context-Based Concepts

The Context-Based Concepts are related to the set of environmental parameters and time for a specific location. As the User is interrelated with the context, the decision made by the User is always depending on the context. Context awareness is most significant in the ubiquitous environment. The user receives the instance of the context class in a particular interval to make accurate decisions on the services. When environmental conditions such as temperature, humidity, luminance and carbon dioxide are received, location is used as an element to identify the control service and select the control equipment. Corresponding control equipment can be selected with the range of the building. As the sector belongs to the room and room belongs to the building, so does the location of the corresponding control device.

```
<owl:Class rdf:ID="Context">
      <owl:disjointWith rdf:resource="#User"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="hasContext">
      <rdf:type rdf:resource="&owl;FunctionalProperty"/>
      <rdfs:domain rdf:resource="#User"/>
      <rdfs:range rdf:resource="#Context"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasEnvironmentalParameter">
      <rdfs:domain rdf:resource="#Context"/>
      <rdfs:range rdf:resource="#Parameter"/>
   </owl:ObjectProperty>
…
```

The code snippet above reads the object property *hasContext* and *hasEnvironmentalParameter* with their respective domain and range. As mentioned in Figure 1, the context holds the information of environmental parameter that is used by user class.

 

### 3.2. Environment-Based Concept

The Environmental-Based concepts hold environmental parameter such as Temperature, Humidity, Illumination and CO2. In the Vertical Farm Ontology Model, autonomous services are achieved through the sensors and actuators, which deals with the environmental parameters. Even though the crops are cultivated in the vertical farm, both the indoor and outdoor weather conditions play a role in the crop's growth condition. During the ventilation, there are many possible outdoor weather conditions which evolve a disease in the harvests. The environmental parameters sensed by such smart devices are described in the code below. The class *Multisenso*r holds four sensed values, which are interpreted in a union form.

```
<owl:Class rdf:ID="EXSENSOR">
    <rdfs:subClassOf rdf:resource="#MultiSensor"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#returnsEnvironmentCondition"/>
        <owl:allValuesFrom>
          <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
              <owl:Class rdf:about="#Humidity"/>
              <owl:Class rdf:about="#Luminance"/>
              <owl:Class rdf:about="#Temperature"/>
              <owl:Class rdf:about="#Carbondioxide"/>
            </owl:unionOf>
          </owl:Class>
        </owl:allValuesFrom>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
```

### 3.3. Hardware-Based Concept

In the Hardware-Based Concepts, both the devices and the control equipment are utilized to extend services. A hardware-based concept supports the service- orientation in the system. Each sector has its own control device implanted in the vertical farm. When an environmental change is noticed in the particular sector, with the help of smart service selection, the control device is decided. Since both the environmental condition and control devices are connected to the Location, the relationship of the sector and the device can be discovered and the action is performed accordingly. Another important class is System that mainly helps in the communication of resources. The property "*isControlledBy*" is linked between the Server (subclass of System) and ControlServices (subclass of Service). OWL axioms in code below briefs the class "AirCondition" and its properties. Similarly, all the other control devices are interrelated using the object property.

 

```
<owl:Class rdf:ID="AirCondition">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#affectsEnvironmentalCondition"/>
      <owl:allValuesFrom rdf:resource="#Temperature"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#ControlEquipment"/>
</owl:Class>
<owl:Class rdf:ID="Humidifier">
  <rdfs:subClassOf>
…
  <rdfs:subClassOf rdf:resource="#ControlEquipment"/>
</owl:Class>
…
```

### 3.4. User-Based Concept:

User-Based Concept consists of users such as administrator, researcher, and system manager, all of whom make use of the context information. All authorized users are afforded their own User ID for access to the services (monitoring and controlling services) as shown in the code below. As all the services are automatic, the user generally plays the role of monitoring the crop’s growth. With the help of automatic web service discovery, the suggestion and the flow of the process are easily obtained for the user.

```
<owl:Class rdf:ID="Administrator">
  <rdfs:subClassOf rdf:resource="#User"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="phone">
  <rdfs:domain rdf:resource="#User"/>
  <rdfs:range rdf:resource="&xsd;int"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Id">
  <rdfs:domain rdf:resource="#User"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
…
```

### 3.5. Service-Based Concept:

Service Class plays a major role in Vertical Farming. The Service based Concept makes use of context information which in turn is used by the User Class. To achieve considerable benefit, the upper ontology of services is inherited from the OWL-S as represented in the OWL code below. The Class Service contains Service Profile, Service Model, and Service Grounding. Many service functions are used in the Vertical Farm Ontology. Each function has defined Input, Output, Precondition, and Effects (IOPE). Semantic Services in the OWL-S works with Input and Output services. Since it is an external procedure, pre-conditions are demanded to satisfy and execute the mapping to provide the result.

```
<owl:Class rdf:ID="Service">
      <rdfs:subClassOf>
         <owl:Restriction>
            <owl:onProperty rdf:resource="#hasServiceProcess"/>
            <owl:maxCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:maxCardinality>
         </owl:Restriction>
      </rdfs:subClassOf>
      <rdfs:subClassOf rdf:resource="&owl-s;Service"/>
   </owl:Class>
   <rdf:Description rdf:about="&owl-s;describedBy">
      <rdfs:domain rdf:resource="&owl-s;Service"/>
      <owl:equivalentProperty rdf:resource="#hasServiceProcess"/>
   </rdf:Description>
   …
```

## 4. Instantiation of Vertical Farm Ontology

The Service-Oriented Ontology for Vertical Farm System is designed with consideration of future pervasive computing. Ontology was implemented in the OWL at Protégé. Ontology was verified by retrieving the desired information with SPARQL Query Language. The semantic interoperability to exchange and share knowledge between the systems is achieved using the ontology model.

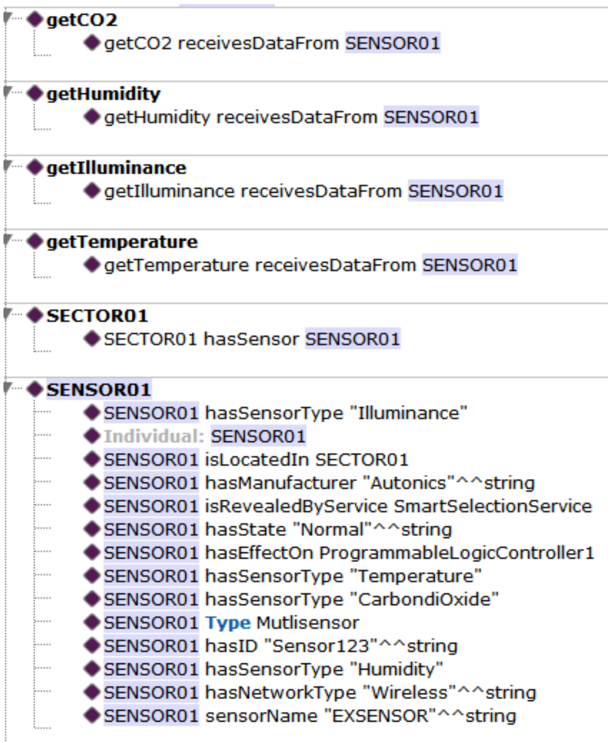


**Figure 2. Instance of Multi-Sensor**

Figure 2 shows the instance of an individual regarding the sensor and its object properties. The location of the sensor is mentioned, which is distinct for each cultivation. Sensor is of the *Multisensor* type that can sense many of context information such as Temperature, Humidity, Illumination and Carbon-di-oxide. Many services make use of the Sensor instance, which includes *getTemperature*, *getHumidity*, *getIlluminance* and *getCO2*. This device is also revealed by the *SmartSelectionService*. The data properties are also shown like *hasID, hasState, hasManufacturer, hasNetworkType* and *hasSensorType*. For a better understanding of the relationships, the domain and range are specified accordingly. The ontology is made to parse the information and invoke the rightful service.

In OWL-S, the three classes, such as Service Profile, Service Process, and Service Grounding, help to connect the owl files with the web services automatically. The service profile has a detailed explanation of the service, while the Service process helps to draw the flow of the process and finally, service grounding helps the ground with the correct web services.

#### 4.1.1. Service Profile

Service Profile is commonly described as “What the Service does”. The common parameters for the profile are Input, Output, precondition, and effects, but it can also use some other parameter necessary for the service discovery of the application.

```
<profile:hasInput>
        <profile:hasDescription rdf:ID= “Control Humidity”>
          <profile:parameterName> humidityValue
          </profile:parameterName>
         <profile:hasController rdf:resource= “&VerticalFarm; #Humidifier”/>
         <profile:referTo rdf:resource=            “&VerticalFarmProcess;
#humidifierController”/>
         </profile:ParameterDescription>
</profile:hasInput>
```

The VerticalFarmProfile helps in understanding the detailed explanation of the service and gives the services to the VerticalFarmProcess. The above code explains the input of humidifierController and their basic information about the service. The service is directed to the VerticalFarmProcess for a particular service named “humidifierController”. According to the change in the environmental condition, the control device selection differs. Therefore, <profile:hasPrecondition> is defined to filter the accurate service.

#### 4.1.2. Service Process

The Service Profile plays a principal part in the service discovery by drawing service in the workflow form. Service profile is used to describe “how the service works”; in other words, the service presents profile. As mentioned earlier, basic functional parameters (IOPE) are linked to the process. According to the complexity of the service, it can be categorized in one of the three subclasses of process, which are simple, atomic, and composite process.

The atomic process in the service process is a single interaction, such as login service. The following example of Crop monitoring also belongs to the atomic process. When the input on the location and crop is given, the environmental conditions are obtained for the particular sector. The composite process involves a workflow of the service in a step-by-step form. When a user signs in, the order of the loading page can also be considered as a simple example of the composite process. A partial atomic process of Crop monitoring is presented in the code below, in which the data property input such as Location, CropName, and CropType are given with the precondition “CropExist”. All the monitoring environmental conditions are obtained as the result of the service.

```
<process:AtomicProcess rdf:ID="CropMonitoring"/>
  <process:Input rdf:ID="Location"
        <process:parameterType    rdf:resource="&VerticalFarm;#Location"/>
  </process:Input>
    <process:Input rdf:ID="CropName">
    <process:parameterType rdf:resource="&xsd;String"/>
  </process:Input>
  <process:Input rdf:ID="CropType">
    <process:parameterType rdf:resource="&xsd;String"/>
  </process:Input>
  <process:Condition rdf:ID="CropExists"/>

  …
    <process:Output rdf:ID="EnvironementalCondition">
            <process:parameterType rdf:resource="#EnvironementalConditionOutput"/>
  </process:Output>
```

#### 4.1.3. Service Grounding

Service Grounding explains the interaction of the service with the WSDL. Simply put, it describes "How to access the service". Grounding maps all the atomic processes in the VerticalFarmProcess to the WSDL, mapping all the input and output to XML input and output messages. According to the needs and IOPE, the web service is selected and linked to the corresponding WSDL, serving a successful automation of web service discovery.

## 5. Conclusions

This paper introduces a smart service ontology for a vertical farm system that includes the concepts for the services, enabling the context awareness to the user, with the hardware-based concepts. Data Integration and interoperability is always a critical issue in an automated environment. For this, OWL-S is used to integrate various web services automatically. The researchers of this paper are broadening their interest and studies to design and implement the service in the test bed environment.

## Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the C-ITRC (Convergence Information Technology Research Center) (IITP-2015-H8601-15-1007) supervised by the IITP (Institute for Information & communications Technology Promotion) and this research was supported by IPET (Korea Institute of Planning and Evaluation for Technology in Food, Agriculture, Forestry and Fisheries) through Agri-Bio-industry Technology Development Program, funded by MAFRA (Ministry of Agriculture, Food and Rural Affairs) (No. 315001-5)

## References

[1] H. Kreger, "Fulfilling the Web services promise", Communications of the ACM, vol.46, no.6 (**2003)**, pp. 29-ff.
[2] R. Chinnici, J. J. Moreau, A. Ryman, & S. Weerawarana, "Web services description language (wsdl) version 2.0 part 1: Core language", W3C recommendation, vol. 26, no. 19, (**2007**).
[3] F. Bobillo, J. Gomez-Gomero, & R. Pérez-Pérez, "Towards Semantic Web Services: A brief Overview", Proceedings of the IADIS international conference, (**2005**).
[4] D. Martin, M. Burstein, D. Mcdermott, S. Mcilraith, M. Paolucci, K. Sycara, and N. Srinivasan, "Bringing semantics to web services with OWL-S" World Wide Web, vol. 10, no. 3, (**2007**), pp. 243-277.

[5] Y. Cho, j. Moon, I. Kim, J. Choi, & H. Yoe, “Towards a smart service based on a context-aware workflow model in u-agriculture”, International Journal of Web and Grid Services, vol. 7, no.2, (**2011**) pp. 117-133.
[6] Y. Cho, S. Park, J. Lee, & J. Moon, “An OWL-based context model for U-agricultural environments”, Computational Science and Its Applications-ICCSA **(2011)**, pp. 452-461.
[7] F. H. Besthorn, “Deep Ecology's contributions to social work: A ten-year retrospective”, International Journal of Social Welfare, vol. 21, no. 3, (**2012**), pp. 248-259.
[8] T. R. Gruber, “The acquisition of strategic knowledge”, Elsevier, (**2013**).
[9] C. Bettini, O, Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, & D. Riboni, “A survey of context modelling and reasoning techniques”, Pervasive and Mobile Computing, vol. 6, no.2, (**2010**), pp. 161-180.
[10] Y. Cho, J. Moon, & H. Yoe, “A context-aware service model based on workflows for u-agriculture”, Computational Science and Its Applications–ICCSA**, (2010)**, pp. 258-268.
[11] T. Kim, N. Bae, M. Lee, C. Shin, J. Park, & Y. Cho, “A study of an agricultural ontology model for an intelligent service in a vertical farm”, International Journal of Smart Homes, vol. 7, no. 4, (**2013**).
[12] Y. Cho, J. Moon, & H. Yoe, “A context-aware service model based on workflows for u-agriculture” In Computational Science and Its Applications–ICCSA **2010** (pp. 258-268). Springer Berlin Heidelberg.
[13] L. A. Kamaruddin, J. Shen, & G. Beydoun, “Evaluating usage of WSMO and OWL-S in semantic web services”, Proceedings of the Eighth Asia-Pacific Conference on Conceptual Modelling, **(2012)**.
[14] A. Ankolekar, M. Burstein, J. R. Hobbs, O. Lassila, D. Martin, D. McDermott and K. Sycara, “ DAML-S: Web service description for the semantic web”, The Semantic Web—ISWC **(2002)**,pp. 348-363.
[15] Z. Maamar, N. C. Narendra, and S. Sattanathan, “Towards an ontology-based approach for specifying and securing Web services” Information and Software Technology, vol. 48, no. 7, **(2006)**, pp. 441-455.
[16] S. Peroni, E. Motta, & M. d’Aquin, “Identifying key concepts in an ontology, through the integration of cognitive principles with statistical and topological measures”, The Semantic Web, **(2008)**, pp. 242-256.

## Authors

**Saraswathi Sivamani**, She completed her Bachelor’s degree in Information Technology from India. She received her Master’s degree on Information and Communication Engineering in South Korea and is currently pursuing a Doctorate degree in the same department. Her area of interest includes Ontology model, Ubiquitous Computing and Big data processing. She has also worked as a Senior Engineer, Java Developer for three years in the Tata Consultancy Services, India.

**Hong-geun Kim**, He received his BS and MS degrees in Information & Communication Engineering from Sunchon National University in 2011 and 2013, respectively. Currently, he is a PhD student in Sunchon National University. His research focuses on Localization and renewable energy harvesting and Embedded System and RFID/USN technologies.

**Myeongbae Lee**, He is a PhD student of Sunchon National University. He is also a researcher for Go-Lab at Sunchon National University. His main research interests include Ubiquitous Computing and Middleware.

**Jangwoo Park**, He received his BS, MS, and PhD degrees in Electronic Engineering from Hanyang University, Seoul, Korea in 1987, 1989 and 1993, respectively. In 1995, he joined as a faculty member of Sunchon National University, where he is currently a professor in the Department of Information & Communication Engineering. His research focuses on Localization and SoC and system designs and RFID/USN technologies.

**Changsun Shin**, He received his PhD degree in Computer Engineering at Wonkwang University. Currently, he is an assistant professor of the Department of Information & Communication Engineering in Sunchon National University. His main research interests include Distributed Real-Time Computing, Distributed Object Modeling, Ubiquitous Agriculture, and Ubiquitous Sensor Network (USN).

**Yong-Yun Cho**, He received his PhD degree in Computer Engineering at Soongsil University. Currently, he is an assistant professor of the Department of Information & Communication Engineering in Sunchon National University. His main research interests include System Software, Embedded Software and Ubiquitous Computing.

 