2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.1.63

An Ontology Based Semantic Service Model in Smart Home Environment

Yi Yang[1+], Zhiliang Wang[1], Yang Yang[1] and Ming Li[1] and Yang Yang[2]

[1]School of Information Engineering, University of Science and Technology Beijing, Beijing, China

[2]College of Information Engineering, North China University of Technology Beijing, Beijing, China

Abstract. Although a great of achievements have been obtained in smart home fields in recent years, there are still two fatal drawbacks. First the smart home system cannot really understand itself; second the system is hard to modify if some of the devices have been changed. By given inspiration from the semantic web, we integrate the semantic character into our smart home system and create an ontology based semantic service model (OSSM) as the middleware of our system. Simulation in a virtual environment represents that OSSM can be used in a real time decision making circumstance and its semantic way of processing makes the whole system much smarter.

Keywords: Component;ontology;semantic service; smart home

1. Introduction

People spend more time in their homes than in any other space. The home ideally provides a safe, comfortable environment in which to relax, communicate, learn, and be entertained. Thanks to today's newly technology, the appliances, compared to their older ones, are much power-efficient, easily to use, can be remotely controlled and have many other benefits. Scientists and researchers are always not satisfied with the results, they still doing their best to make the home system much smarter and great achievements have been obtained already.

The University of Texas at Arlington have built a smart home prototype called "MavHome" [1~2]. MIT have developed a smart home architecture "Home of Future" [3]. Department of Computer Science and Institute of Cognitive Science University of Colorado have realized their smart home model "Neural Network House (ACHE)" [4]. University of Florida's Mobile and Pervasive Computing Laboratory have done some similar work on the "GatorTech Home" [5] platform.

But like all other systems nowadays, there are still complains among inhabitants. They are not intelligent enough as the word "smart" have claimed is the crux of the most of complains. By noticing the fact that all the smart home systems mentioned above are based by several static pre-defined rules and the processing logic after decision have made are quite simple. That mechanism causes two fatal drawbacks. First the system itself is unable to understand the real meaning of the decision and the consequences that after the decision has been operated. Second system is hard to modify if some of the devices have been changed because of the pre-defined rules.

Ontology based semantic web have achieved great success in SOC(Service Oriented Computing)[6] field in recent years. It is a key to conquer the drawbacks mentioned above by considering the ontology based semantic service is a more natural way to let the system understand what it is going to do and what it have done. This is where our research begins.

[+] Corresponding author.
E-mail address: yangyi0252087@163.com.

The research directions involve:

I. Introduction
II. Ontology Based Semantic Service Model
III. Simulation
IV. Conclusion

2. Ontology Based Semantic Service model

2.1 Definition

1) *Definition 1(Ontology Model):*

Ontology Model is defined as O<C,P,I>

where

- C Class
- P Property(including Object Property and Data Property)
- I Inherited relationship(including Class inheritance and Property inheritance)

2) *Definition 2(Semantic Similarity):*

Semantic Similarity is defined as M= $\sum_{i=0}^{n} F_i * W_i$

where

- Fi the ith Function Semantic value which is provided by physical certain device
- n the number of certain series of Functions
- Wi the weight of Function Semantic value which is associate with Fi

Semantic Similarity can be considered as an indicator which is used to determine whether to execute the service by the upper system

3) *Definition 3(Service)*

Service is defined as S<L,E(FD),M>

where

- L Location in home
- E(FD) a series of Functions which are provided by physical devices
- M Semantic Similarity

4) *Definition 4(Decision):*

Decision is defined as D<L,E>

where

- L Location in home
- E Effect On Environment

The Decision is made by Upper System that indicates which Environment factor should be effected.

5) *Definition 5(Ontology Based Semantic Service Model):*

Ontology Based Semantic Service Model is defined as OSSM<D,O,F,S>

where

- D Decision that the Upper system has made, this parameter can be considered as input
- O Ontology Model
- F Profiles(including Environment Profile and Inhabitant Profile)
- S Service, this parameter can be considered as output

2.2 Composition and hierarchy

OSSM can be considered as a middleware of the whole smart home system. It has an input "Decision" which is made by upper system and an output "Service" which is provided for lower system. We use OWL-DL(Web Ontology Language-Description Language)[7] as the language to describe Ontology as it is the recommended stander made by W3C(World Wide Web Consortium) and we choose Protégé[8] as our OWL entity creation tool. The hierarchy of OSSM is shown in figure 1.

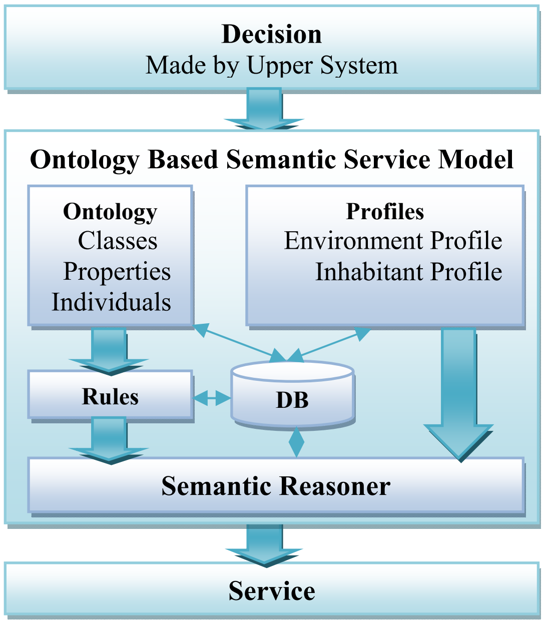

Figure 1. The Hierarchy of OSSM

The core of OSSM consists of four critical parts.

1) Ontology Part

This part contains a formal, explicit specification of a shared conceptualization of a smart home. It represents a set of concepts and the relationships between those concepts among Device, Environment, Location and other classes. By Now, this part contains 58 classes, 11 object properties and 3 datatype properties. Figure 2 shows the class tree of the ontology. Table 1 is the property list in detail which has ignored each reverse property.

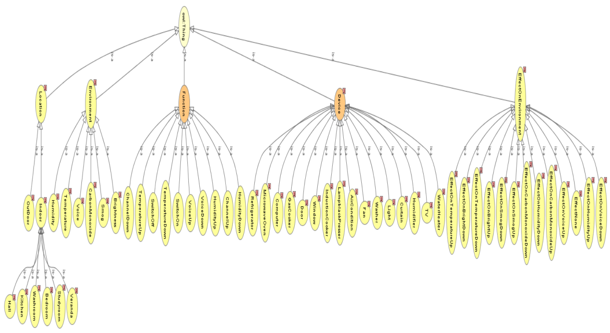
Figure 2. Class Tree of The Ontology

Table 1 Properties of The Ontology(Ignore The Reverse)

Porperty	Type	Domain	Range
hasEffect	Object	Function	EffectOnEnvironme nt
hasEnvironment	Object	Location	Environment
hasBrightness	Object	Location	Brightness
hasCarbonMonoxide	Object	Location	CarbonMonoxide
hasHumidity	Object	Location	Humidity
hasSmog	Object	Location	Smog
hasTemperature	Object	Location	Temperature
hasVoice	Object	Location	Voice
hasFunction	Object	Device	Function
hasLocation	Object	Device/ Environment	Location
hasSemanticValue	Data	Function	XSD: Double

Porperty	Type	Domain	Range
hasSemanticWeight	Data	Function	XSD: Double
isEffecting	Data	Device	XSD:Boolen
hasReverseFunction	Object	Function	Function

2) Profile Part

Profile part contains some meta rules and facts about environment and inhabitants. For example, the optimum temperature ranges in a certain location; The preference of inhabitants in using air-condition. It is used to provide suggestions to the reasoner.

3) Rules Part

Rule is a logical sequence and is used by the semantic reasoner. In our model, the Rules part composed of two such rules which are written in SWRL(Semantic Web Rule Language)[9] format as follows(individual(EffectOnEnvironment) means the parameter is an individual of class EffectOnEnvironment).

- Rule 1:

Function(?x)^hasEffect(?x,individual(EffectOnEnvironment))^hasSemanticValue(?x,?z)^swrlb:greaterThen(?z,0)^isFunctionOf(?x,?y)^Device(?y)^¬isEffecting(?y,individual(EffectOnEnvironment))^hasLocation(?y,individual(Location))->Service(?x)

- Rule 2:

Function(?x)^hasEffect(?x,individual(EffectOnEnvironment))^hasSemanticValue(?x,?z)^¬swrlb:greaterThen(0,?z)^isFunctionOf(?x,?y)^Device(?y)^isEffecting(?y,individual(EffectOnEnvironment))^hasLocation(?y,individual(Location))^hasReverseFunction(?x,?a)->Service(?a)

In more natural human language, rule 1 indicate the reasoner to get functions of devices which make positive contribution to the given effect on environment and are not effecting the environment now in certain location. While rule 2 tell the reasoner to get reverse functions of certain functions of devices which make negative contribution to the given effect on environment and are effecting the environment now in certain location.

4) Semantic Reasoner Part

A semantic reasoner is a software machine which can infer semantic logical results from a set of rules or axioms or preset facts. In our model, we use Jess(Java Expert System Shell) as core of our semantic reasoner. We choose Jess according to the following 3 reasons:

- Jess is a small, light and fastest rule engines available
- Jess is free for academic use
- Jess is written in Java language which means it is easier to integrate in our system

Semantic reasoner part first imports the rules, ontology instance(which is created by Ontology part) and then gathers the assistance information from database and profile part, at last it infers a triple < L,E(FD),M > which indicates a service.

3. Simulation

3.1 Simulation environment

In order to overcome the high cost of establishing the actual experimental environment, and poor controllability of the actual environment, but focus more on the model itself, we use virtual reality technology[10] to build a simulation platform, which we called virtual smart home.

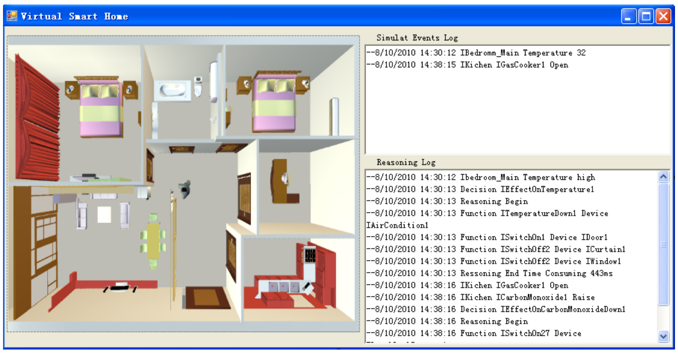

Figure 3. Screenshot of Virtual Smart Home

In fact, the environment simulates a ordinary department which contains two bedrooms, one washroom, two halls, one study room, one kitchen and one veranda.

3.2 Background environment parameter

In our scenario we only care about six environment parameters: temperature, humidity, brightness, voice, smog and carbon monoxide. Some of these parameters are only affected by indoor devices which we called “D Param” and others are affected by both indoor devices and outdoor environment which we called “G Param”. Table 2 shows the classification.

Table 2 Classification of Environment Parameter

Environment Parameter	Class
temperature	G Param
humidity	G Param
brightness	G Param
voice	D Param
smog	D Param
carbon monoxide	D Param

3.3 OSSM instance

Before start simulation, we must first instantiate the OSSM which means the ontology individual should be created. According to our simulation environment, we have created 205 individuals. Some of the individuals are shown in table 3.

Table 3 Individual Details (Only Portion)

Individual	Class	Property	Property Detail
IAirCondition1	AirCondition	hasFunction	IHumidityUp1 IHumidityDown1 ITemperatureUp1 ITemperatureDown1
		hasLocation	IBedroom_Main
		isEffecting	True/False
IEffectOnTemperatureDown1	EffectOnEnvironment	isEffectOf	ITemperatureDown1 ITemperatureDown2 ITemperatureDown3 ITemperatureDown4
ITemperature1	Temperature	hasLocation	IBedroom_Main
		isEnvironmentOf	IBedroom_Main
		isTempratureOf	IBedroom_Main
IHumidityUp1	HumidityUp	hasEffect	IEffectOnHumidityUp1
		isFunctionOf	IAirCondition1
		hasSemanticValue	1
		hasSenmanticWeight	1

Individual	Class	Property	Property Detail
		hasReverse Function	IHumidityDown1

3.4 Event examples

1) Event 1

The temperature of IBedroom_Main raise to 32 °C

System notices the temperature of IBedroom_Main is too high then IEffectOnTemperatureDown1is made as a decision to OSSM. OSSM asks ontology what to do according to the suggestion of the profile. Finally, OSSM outputs the Service as follows:

- Start the IAirCondition1's function ITemperatureDown1
- Close the IBedroom_Main's window
- Close the IBedroom_Main's curtain
- Close the IBedroom_Main's door

Figure 4 shows the output reasoning log of OSSM.

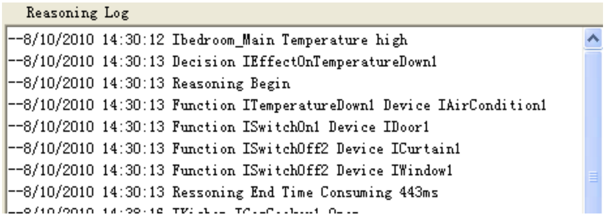
Reasoning Log
--8/10/2010 14:30:12 Ibedroom_Main Temperature high
--8/10/2010 14:30:13 Decision IEffectOnTemperatureDown1
--8/10/2010 14:30:13 Reasoning Begin
--8/10/2010 14:30:13 Function ITemperatureDown1 Device IAirCondition1
--8/10/2010 14:30:13 Function ISwitchOn1 Device IDoor1
--8/10/2010 14:30:13 Function ISwitchOff2 Device ICurtain1
--8/10/2010 14:30:13 Function ISwitchOff2 Device IWindow1
--8/10/2010 14:30:13 Reasoning End Time Consuming 443ms

Figure 4. Reasoning Log of Event 1

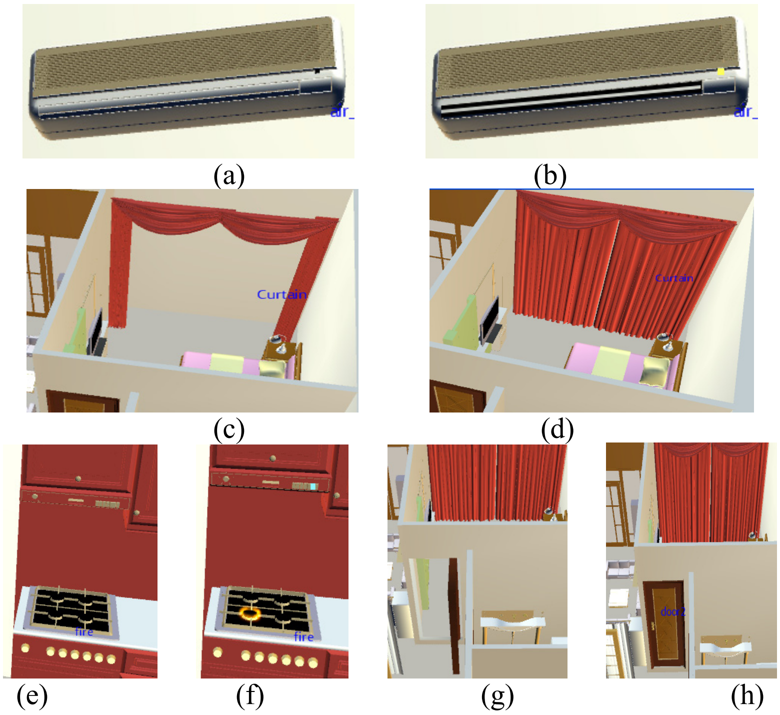

Figure 5. Some Visual Effect Created by Virtual Smart Home

2) Event 2

The temperature of IBedroom_Main raise to 32 °C and the lights are still lighting in the IBedroom_Main.

Quite similar to the event 1, OSSM output the Service as follows:

- Start the IAirCondition1's function ITemperatureDown1
- Close the IBedroom_Main's window
- Close the IBedroom_Main's curtain
- Close the IBedroom_Main's door
- Close the IBedroom_Main's lights

Consider the lighting lights can also produce hit in the room (we have assigned the negative hasSemanticValue to its "SwichOn" Function), so according to the ontology rule 2, the reverse function of lighting the lights should be activated (SwitchOff).

3) Event 3

The IGasCooker1in the IKitchen has been opened.

System detects the carbon monoxide level is raising in the IKitchen and it tells OSSM should do the effect ICarbonMonoxideDown1. The output of OSSM is as follows:

- Open the IKitchen's window
- Open the IKitchen's lampblack Presser

Fig.5 shows some visual effect created by virtual smart home. Fig.5(a) shows the status when the air-condition is close while fig.5(b) shows the status after it is opened; Fig.5(c)(d) illustrate how it looks like when the curtain is open and close; Fig.5(e)(f) express when the gas cooker is opened, the lampblack presser will open automatically; Fig.5(g)(h) show the behavior of the door.

3.5 Time consuming test

OSSM is supposed to use in a real time decision making situation so the time consuming character of OSSM is critical. We have done 100 times of consecutive tests based on random events. Fig.6 shows the result.

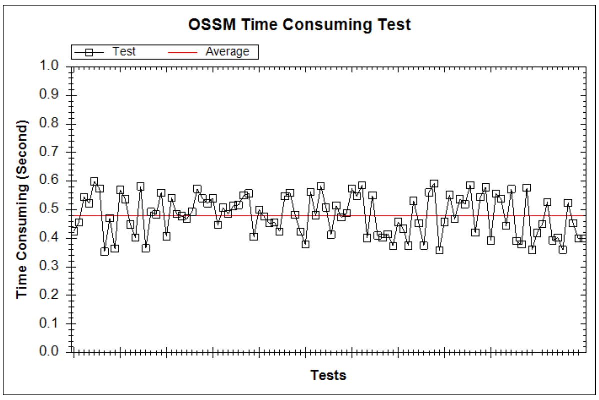

Figure 6. OSSM Time Consuming Test Result

The line with the rectangle symbols shows the result of each test while the line in the middle shows the average time consuming. It is glad to see that all the time consuming is less than 0.65 second, its average is 0.48 and variance is 0.025. The inhabitant can hardly feel any time delay if the decision has been made in less than 1 second.

4. Conclusion

The simulation in virtual smart home environment shows that OSSM can provide service quite properly according to the decision which is made by the upper system because of its semantic character. Although OSSM will cause a little time delaying it can be tolerated by inhabitant and so can be used in a real time decision making occasion. Though progress has been achieved much, we have a clear understanding that there are still disadvantages with OSSM. In the next stage of research, we will take follow factors into our consideration:

- The interactions of different rooms
- The interactions of indoor and outdoor environment
- The interferences of inhabitant

5. References

[1] Karthik Gopalratnam, Diane J. Cook, Online Sequential Prediction via Incremental Parsing: The Active LeZi Algorithm, IEEE Intelligent Systems, Vol.22, 52-58, 2007.

[2] Abhishek Roy, Sajal K. Das, Kalyan Basu, A Predictive Framework for Location-Aware Resource Management in Smart Homes, IEEE Transaction on Mobile Computing, Vol.6, 1270-1283, 2007.

[3] Stephen S. Intille, Designing a Home of the Future, IEEE Pervasive Computing, Vol.1, 80~86, 2002.

[4] M. Mozer, The Neural Network House: An Environment that Adapts to Its Inhabitants, Proc. AAAI Spring Symp, Intelligent Environments, SS-98-02, 1998.

[5] Helal A., Mann H., EI-Zabadani, The gator tech smart house: A programmable pervasive space. IEEE Computer, Vol.38, 50-60, 2003.

[6] Munindar Paul Singh, Michael N. Huhns, Service-oriented Computing: semantic, processes, agents, John Wiley & Sons Ltd, 2005, pp.3-15.

[7] W3C, OWL Web Ontology Language Guide, http://www.w3.org/TR/owl-guide/, 2004.

[8] Matthew Horridge, A Practical Guid To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.2, The University Of Manchester, 2009, pp.1-109.

[9] W3C, SWRL: A Semantic Web Rule Language Combining OWL and RuleML, http://www.w3.org/Submission/SWRL/, 2004.

[10] Stephen Cawood, Pat McGee, Microsoft XNA Game Studio Creator's Guide, McGraw-Hill, New York, USA Inc, 2007, pp.8-28.