On the Dynamic Management of Information in Ubiquitous Systems Using Evolvable Software Components*

Syed Shariyar Murtaza[1], Bilal Ahmed [2], and Choong Seon Hong[3,**]

Dept. Of Computer Engineering Kyung Hee University, South Korea
shariyar@networking.khu.ac.kr, bilal@oslab.khu.ac.kr,
cshong@khu.ac.kr

Abstract. Rapid evolution in ubiquitous systems is complicating its management. In the realm of ubiquitous computing, the domain specific ontologies are constructed manually by experts, which is quite a toiling task. We proposed to extract ontologies dynamically for ubiquitous systems. We also argue that the components of the software middleware are the specialized research areas with different subject orientation e.g. context reasoning, service discovery etc[1] and following an evolvable component oriented approach would let the components of the software evolve independently of other software components, while making them interoperable irrespective of versions and vendors.

1 Introduction

With the ubiquity of emerging ubiquitous devices, our access to data would evolve exponentially and this continuous evolution in the information and interfaces would overwhelm the humans. Therefore, we need a mechanism which could evolve both in terms of information and software.

Different types of software infrastructures for the ubiquitous systems have been developed in the past years like Gaia [5], Solar System [8] and Context Toolkit [9] etc. But, they didn't consider the gradual evolution of the ubiquitous environments and the composition of ontologies is also done manually. To cope with evolution and interoperability, it is necessary to separate the overall environment into smaller logical groups or modules. Also, in these rapidly evolving ubiquitous environments, it may not even be desirable to build a comprehensive ontology, because the interfaces of appliances or devices are liable to change. Therefore, we seek to develop onotologies dynamically. Number of researchers has proposed different dynamic ontology extraction algorithms for text documents and web pages e.g. Chung Hee Hwang [2], Giovanni et al [3], and Zhan Cui et al. [4] etc. But, the dynamic extraction of ontologies in the ubiquitous environment to learn the information about the environment and user to provide him/her the seamless effect in the interaction with devices and services around him was never considered.

* This wok was supported by MIC and ITRC Project.
** Corresponding author.

Y.-T. Kim and M. Takano (Eds.): APNOMS 2006, LNCS 4238, pp. 513 – 516, 2006.

2 System Description

Our system contains two major applications: Server application and User application, along with several device applications for each home appliance (TV, Bulb etc).

The system employs UPnP [7] protocol and is developed using C# and Intel's UPnP SDK. A self explanatory description of the system is shown in figure 1.

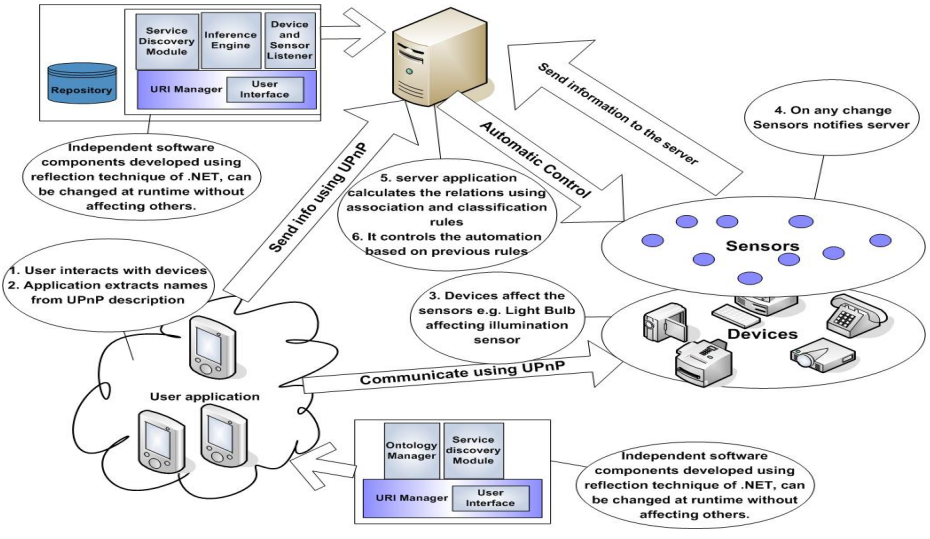

Fig. 1. System Overview

On user's interaction with devices, the user application extracts device name from the UPnP based device description by extracting the stop words from the manufacturer tag, then eliminating stop words[1] from the common string of the "Model Name" and "Model Description" tags of UPnP device description. This is shown in figure 2. The next step is to find the relations between devices. This is done at the server by finding the device and sensor relation using association rules [6]. The general template of the association rule is given below, where D manifests device and S manifests sensor. These association rules then become the basis of the device and schedule classification rules.

$$\mathbf{D_1 \cap D_2 \cap D_3 \cap \ldots . D_n \rightarrow S_i} \quad (1)$$

Device Classification Rules:

Let X={X1, X2….. Xn} be the devices and Xi, Xj $\in$ X where i,j=1 to n

1) If X1, X2 devices affect some common sensors and other uncommon sensors then they can be regarded as similar devices.
2) Devices always affecting same common sensors are same devices.
3) Same Device: If X set of devices is the 'Same Device' and if Xi has the request of use and it is not available then use Xj such that Xj value of effect

[1] Here, stop words represent those words which are not the part of actual name.

on the sensors (e.g. illumination intensity level) is less than or equal to Xi greater than all the value of effects in Y=X-Xv : Xv={Xi,Xj}

4) Environment Control: If X set of devices is similar and the request or priority of use of the value of effect from the user is c (like Noise level 30 of room) then sum of value of effect of all the devices in X should be less than equal to c.

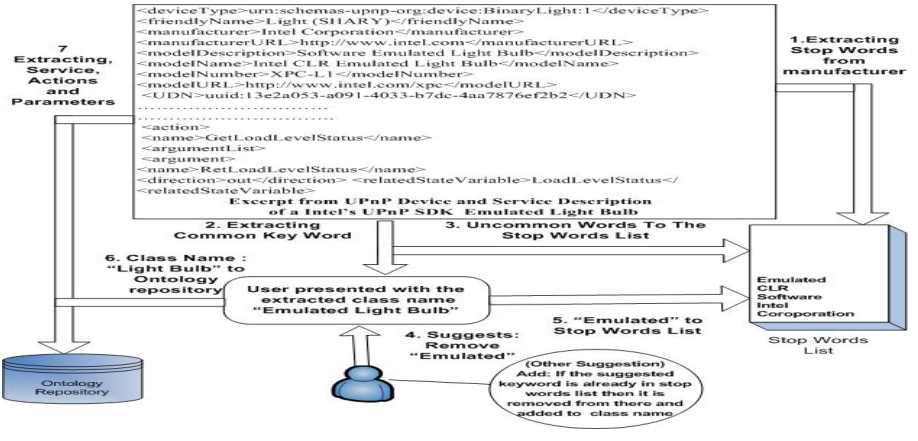

Fig. 2. Process of Extracting Device Level Concepts

Schedule Classification Rules:

1) Divide the time duration of a day into different slices of time.
2) For every slice of time calculate the highly used devices (devices having sup port greater than 50) and start them automatically.
3) Similarly, calculate low used devices (devices having sup port from 25 to 50) and leave them in their current state, while turn off all other devices
4) If the device throws error or not found then go for same device rule (1).

3 Evaluation

We considered a single room, a user and designed simple UPnP home appliances e.g. dimmable light bulb, medium size microwave oven, air conditioner with temperature control facility and etc. Similarly, we deployed different sensors like illumination sensor, temperature sensor and noise sensor. We performed the experiment for a week and divided the time durations into different slices according to the schedule rule for this experiment: Breakfast and wakeup timing, office timing, dinner and relaxation timings and sleep timing. Initially, the user application extracted the names then the server application calculated the relationships according to the association and classification rules of the previous section. We have shown these ontologies and their results in the form of a tree in the figure 3.These results allow the system to automate the environment and provide a seamless effect in a number of ways e.g. When the user gets up in the morning then according to the schedule rules of the previous sections the system will automatically turn on the Light bulb, but if it is not available then by same device rule it will turn on the lamp. Similarly, many other scenarios are possible.

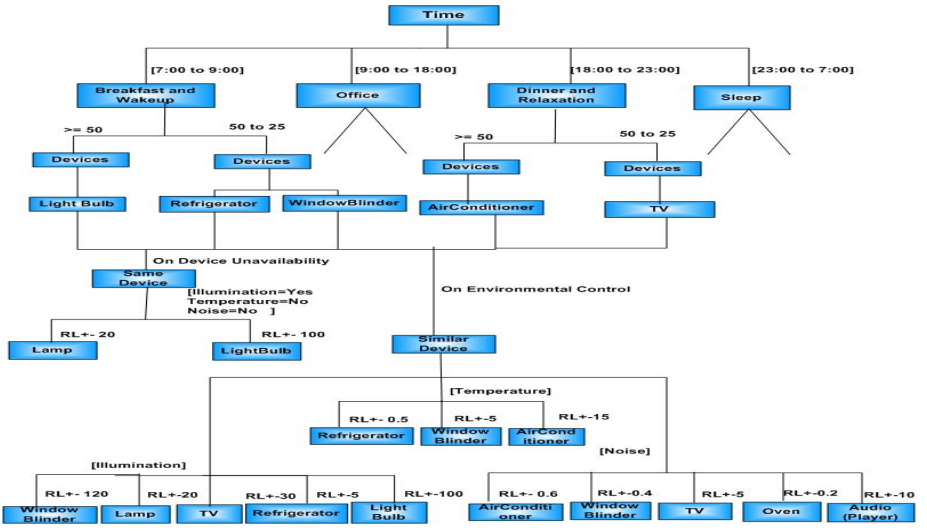

Fig. 2. Tree Diagram for Classification Relationship

In the future, we would like to publish these extracted ontologies in the form of RDF to reuse and share this work with other systems

References

1. Syed Shariyar Murtaza,Choong Seon Hong," An Evolvable Software Architec-ture for Managing Ubiquitous Systems",Proc. of 8th Asia Pacific Network Operation and Management Symposium (APNOMS), September 2005,pp 400-409
2. Chung Hee Hwang, "Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information", proc. of the 6th International Workshop on Knowledge Representation meets Databases, July ,1999.
3. Giovanni Modica, Avigdor Gal, and Hasan M. Jamil, "The Use of Machine-Generated Ontologies in Dynamic Information Seeking", Proc. of the 9th In ternational Conference on Cooperative Information Systems, 2001, 443- 448
4. Zhan Cui, Ernesto Damiani, Marcello Leida, Marco Viviani, "OntoExtractor A Fuzzy-Based Approach in Clustering Semi-structured Data Sources and Meta data Generation", proc. of Knowledge-Based Intelligent Information and Engineering Systems, LNCS, 05
5. A Ranganathan, Roy Campbell: A Middleware for Context -Aware Agents in Ubiquitous Computing Environments, proc. of ACM/IFIP/USENIX Middleware Conference,June,03
6. Han, Jiawei. Kamber, Micheline.: Data Mining: Concepts and Techniques, Morgan Kauf mann Publishers (2001): ISBN: 1558604898.
7. UPnP forum, http://ww.upnp.org
8. Guanling Chen and David Kotz.: Solar: An Open Platform for Context –Aware Mobile Applications, proc of the First International Conference on Pervasive Computing (Pervasive 2002), Switzerland, June, 2002.
9. Context Toolkit project http://www.cs.berkeley.edu/~dey/context.html