MDMA:A Modular Distributed Middleware Architecture via URI

Syed Shariyar Murtaza, Choong Seon Hong
Department Of Computer Engineering, Kyung Hee University
shariyar@networking.khu.ac.kr,cshong@khu.ac.kr

Abstract

This paper uses our proposed model for connecting ubiquitous physical objects over the web using URI, while utilizing the already developed frameworks, for ubiquitous service discovery like JINI, UPnP, and RDF/OWL for semantic web. By using this proposed scheme, we have presented architecture of a service oriented modular distributed ubiquitous middleware i.e. MDMA

I. Introduction

Web constitutes many resources. All the resources are present like a .gif, .html, .xml file etc and are represented by Universal Resource Identifier or URIs. If there is any thing on the web, then URIs can be used to specify them. Most common URIs that we encounter usually starts with http: which means that they belong to a space of objects which is accessed by http protocol. Metadata as every one knows is the data about data or information about information. RDF' s main purpose is the interoperability of metadata .It provides descriptions of Web resources or the description of any other object having a Uniform Resource Identifier (URI) as its address and this description is in machine understandable form. RDF uses ontology to define the meaning, characteristics, constraints and relationships of a set of properties. Anything that has a URI (i.e. resource), is used to represent ontology. Ontology could be reused, once created. The use of URIs for physical objects would promote the use of personal ontologies for individual devices by a user.

Currently, middleware for ubiquitous systems e.g. home network are deployed as a unit and usually a home gateway is used for his purpose. We envisage that by distributing the ubiquitous middleware into components on different devices, would help us leverage the computationally intensive devices, while keeping the energy preserve for the energy intensive devices. Further, the components of a middleware for ubiquitous systems vary widely in their subject orientation. We have specialized research areas and dedicated researchers for every component of the system e.g. context reasoning, service discovery, sensors and RFID are specialized research areas themselves. Therefore, by separating the middleware into modular services, these components could be developed separately and provided independently by different manufacturers. It would give the chance to manufacturers to improve or update their services independent of other services in the system; even they could charge for their services too.

II. URI Model for Devices

We extended the domain name system to one more domain, which could provide the network addresses of the physical resources in the world.

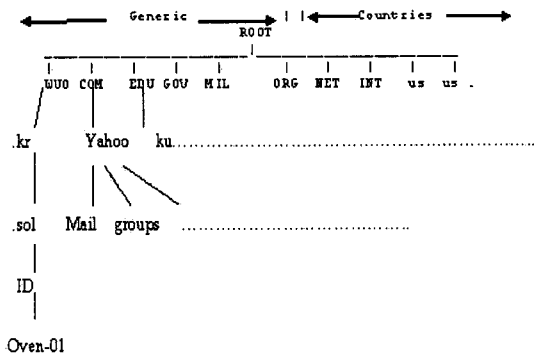

Fig 1.0 A n Extended portion of internet DNS

Here WUO stands for " A Web Of Ubiquitous Objects" , kr represents country Korea, sol represents

city Seoul, ID represents the social security number or national identity card number of a person and Oven-01 is identifier of the microwave-oven for that person.

A question could arise here, why have we chosen the National ID-Card number or Social Security Number? The answer is two fold; we need a unique identification while keeping it comprehendible for a person. Almost, every person remembers his social security or National identification number. Although, initially it is difficult to remember, just like a phone number, but it has been observed that after using it few times, people tends to remember it.

This URI will be used to access it over the web and its properties can be manipulated via Http and XML as URI scheme and communication language respectively.

III. Architecture

Based on our URI model, we also propose a modular distributed middleware architecture, for the development of the ubiquitous systems. Our architecture follows a service oriented modular approach. By modular we mean that our middleware is distributed in components or modules and it is not necessary for each module to be on the same system, they could be distributed on different systems. e.g. A microwave oven does not need to posses a context reasoning mechanism, always, to make inferences. Only specified ontology represented in owl or RDF would be enough for microwave oven to keep with it. While, the context reasoning service, when needed, could be retrieved (or data could be sent), via URI, from some web server or any PDA etc, available in the ubiquitous environment. We have also not used any home or residential gateway, instead we employed http services in all the devices, due to which these device could be operated directly from the web .Also, if every person keeps his PDA/mobile phone with him then he would be able to manipulate the devices with his personnel configuration either in his own home or any other network. This is manifested in the figure 2.0.

Currently, we have employed the OSGi based architecture for our middleware, because it is evolving as a standard in ubiquitous environment for interoperability .Secondly, developing a separate service discovery interoperability middleware is a separate research area and is out of scope of our work. But as mentioned, to prove the feasibility of our work we employed OSGi and currently different service discovery consortiums don' t agree on any common standard.

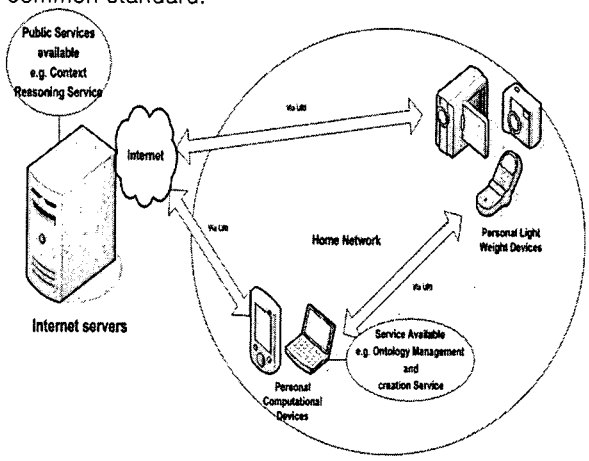

Figure2.0 Overview of the MDAM

Our framework architecture is shown in the figure 3.

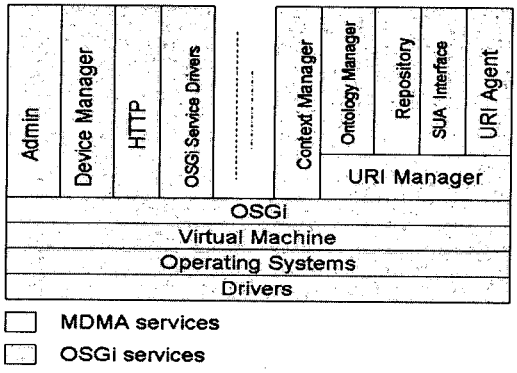

Figure 3.0, Software Component Architecture

Our architecture is divided into two parts pure OSGi based services and MDMA services. Some of the OSGi services include, administration, device management, HTTP and service discovery drivers for UPNP, JINI, Bluetooth etc. Figure 3.0, shows the software component architecture. This architecture could be deployed as a unit on single system, but we propose a modular approach to utilize the computational device. In case of a modular approach, as we proposed a URI for every device; we'll use that URI scheme to identify different components lying on different devices. To achieve the modularity, we distribute the components according to their tasks and needs. We'll briefly elaborate on each

URI Manager: It is the main component which interacts with all the modules of the system. URI

manager not only manages but also keeps record of the URIs of the components used. Particularly in home environment, when the service is available on the near by device, it doesn't use the internet name service, instead query the native service discovery protocol.

URI Agent module contains the two code mobility approaches, "Code On Demand" and "Mobile Agent".

Ontology Manager: Ontology module provides the mechanism to retrieve ontology for a particular service via URI Agent module and the methods to create, save and delete the ontology and the schema for it. This ontology manger would not be needed by every device. Therefore, we would rather keep it in a separate device, like laptop or PDA etc.

Service User Agent Interface: This module acts as a presentation module, which interacts with URI agent to produce the interface for the users. . This module would be appropriate for only the display devices, like Personal computer, TV etc.

Context Reasoner : Context reasoner provides the reasoning mechanism on the ontologies provided by the stationary or dynamic context provider. By stationary we mean the devices which do not sense information like microwave oven, digital cameras, personal video cameras. By dynamic we mean the context gained from the sensors, RFIDs or some external web server to get the weather information.

Context Converter: Context converter provides the mechanism for dynamic context providers to convert the sensed data to a context representation language such as OWL or RDF.

Repository: is a collection of the data present in the system. It keeps the data within the system e.g. Collection of service descriptions, ontologies and sensed data. Repository is of two kinds. First, one is a minimize repository. , which is kept by every device to keep the service descriptions and sensed data if any. Second one is a large repository which keeps the historical or un-used sensed data for a time period specified by administrator. This type of repository is kept in a device, which enough or large disk space available for it. From time to time summarization technique can be applied to remove the already used data or deleted with the concern of administrator.

IV. Conclusion

Today, the researchers are trying to make the human intervention as less as possible in the computing. Our research is also focusing on this and this paper has propsed a service oriented modular distributed middleware, based on our proposed URI model for devices.work we employed OSGi and currently differen

Currently, we are working on the implementation stage of our middleware and to employ the URI mechanism in it. We envisage a modular middleware with URI approach would resemble object oriented paradigm with service oriented functionality. Just as in object oriented computing, researchers and engineers can focus only on their respective service , while leveraging the use of unified middleware.

References

1 Andrew S Tanenbaum, Computer Networks,4th Edititon,2003

2. R. Fielding,UC Irvine, J. Gettys, J. Mogul, H. Frystyk, L. Masinter,P. Leach , T. Berners-Lee, HTTP 1.1, RFC 2616, June 1999

3. T. Berners-Lee ,R. Fielding, U.C. Irvine, L. Masinter , Uniform Resource Identifiers (URI): Generic Syntax, August 1998

4. Sean B. Palmer, The Semantic Web: An Introduction, http://infomesh.net /2001/swintro/, September 2001

5. Tim BL Primer, Getting into RDF & Semantic Web using N3, http://www.w3.org/2000/10/swap/Primer

6. Dipanjan Chakraborty, Filip Perich, Sasikanth Avancha, Anupam Joshi, DReggie: Semantic Service Discovery for M-Commerce Applications,2001

7.Tom Broens, Context-aware, Ontology based, Semantic Service Discovery, a master' s thesis, Enschede, The Netherlands ,July, 2004

8.Understanding Code Mobility, Alfonso Fugegetta, Gian Pietro Picco , Giobvanni Vigna , IEEE transaction on Software Engineering, vol. 24, No. 5,May 1998

9.The Open service gateway initiative ,An introductory overview, Dave Marples, Peter kierens, IEEE communication Magazine ,December 2001

10 Towards an OSGi-based infrastructure for Context-Aware Aplications, Tao Gu, Hung Keng Pung,Da Qing Zhang, Pervasive computing , IEEE, 2004

11 Device and Service Discovery in Home Networks with OSGi , Pavlin Dobrev, AGDavid Famolari, Christian Kurzke, Brent A. Miller, IEEE Communications Magazine, August 2002