Ontology based Service Discovery Middleware for Heterogeneous Agent Platforms

Sejung Oh, Juryon Paik, Heeyong Youn, and Ungmo Kim

***Abstract*—In ubiquitous computing environments to use distributed services and devices efficiently the basic information should be provided about types or locations of services in advance. A process explores such information that meets a user's needs. We call it as service discovery which is the most primitive system in distributed mobile computing environment such as ubiquitous. It is a trend that use of a combination of developed agent technology and service discovery to manage efficiently and flexibly distributed services, devices, and users. A number of agent platforms have been developed by many academic institutes; however there exist problems about interoperability among agent platforms. FIPA suggest a discovery middleware module that provides to interoperate between agent platforms. In this paper, we developed the prototype of ontology based discovery middleware based on FIPA specification to facilitate sharing services between FIPA-OS, JADE, and SLP platforms.**

***Index Terms*—Service Discovery, Discovery Middleware, Context Aware Service, Agent Platform**

I. Introduction

UBIQUITOUS computing is the method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user [3]. Distributed computing is more pertinent to this environment than centralized computing. Because network connection or service information is changed dynamically if a centralized system controls those changes it'll produce a lot of network traffics. In distributed environments, the information about types or locations of services should be provided to interoperate services and devices. We call it as service discovery which provides such information [4]. Service discovery technology and management systems, which manage individual services, were needed and agent platforms are developed to support those systems [2]. An agent is software which works behalf of a user. Agent platform provides every resource and run time environments that agents need to meet the user's needs. Furthermore, it provides discovery mechanism for searching service environment and acquiring of information.

This work was supported in part by Ubiquitous computing Technology Research Institute (UTRI) funded by the Korean Ministry of Information and Communication, Republic of Korea.

The authors are with the Department of Computer Engineering, Sungkyunkwan University, Gyeonggi-do, 300 Chunchun-dong Jangan-gu Suwon, Republic of Korea. (phone: 82-31-290-7218; fax: 82-31-290-7211; e-mail: ohsejung@skku.edu).

The famous agent platforms are JxTA [9], UPnP [11], JADE [13], and FIPA-OS [7] also there is SLP [1] which is not a platform but a protocol that facilitate service discovery [12].

The domain reiteration of agent platforms can be occurred in ubiquitous environments. Therefore, a standard mediation is necessary to support compatibility. FIPA suggest a discovery middleware module that provides to resolve such problems [6]. The current version of FIPA-OS agent platform based on FIPA specification does not provide the implementation of discovery middleware.

In this paper, we present the prototype of ontology based discovery middleware to facilitate sharing services between heterogeneous agent platforms. The use of suggested discovery middleware make it possible to share a service of agent platforms and also to perform as following operations 'registration', 'deletion', and 'modification' between heterogeneous agent platforms.

The rest of this paper is divided into three sections. In section 2, we introduce several agent platforms. In section 3, we describe discovery middleware and the design of our system architecture. Finally we discuss conclusion and future works.

II. Related Works

A. SLP

SLP is a protocol for automatic resource discovery on IP based networks. It is possible to operate registration, deletion, and modification in SLP domain [10]. The main purpose of using SLP is a discovery of location of services in a domain by network level. Therefore it is not treated application level of providing a real service. SA represents providers every device and application, and UA represents consumers that request using a specific service. DA mediates between UA and SA, and also stores information of services. Figure 1 illustrates the basic functionality of the protocol. If there is more than one DA, SA and UA search every DA in their domain using multicasting. SA register a service to DA, and then UA acquires information of SA through DA.

As mentioned above, the purpose of SLP is acquisition of location of services. Because cannot provide implementation of application. Therefore SLP needs some expansion of application level for interoperation of heterogeneous agent platforms.

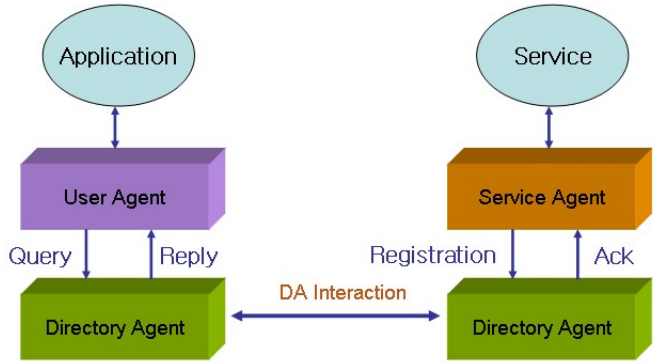

Fig. 1. SLP architecture

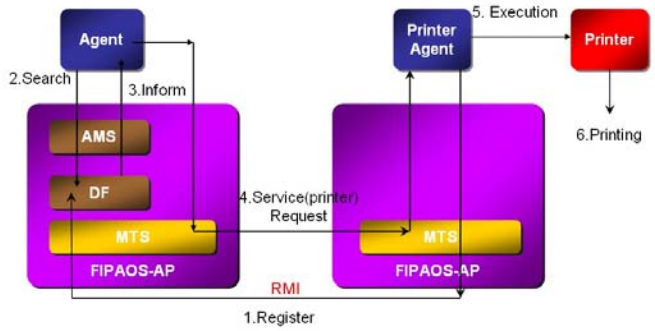

Fig. 2. FIPA-OS architecture

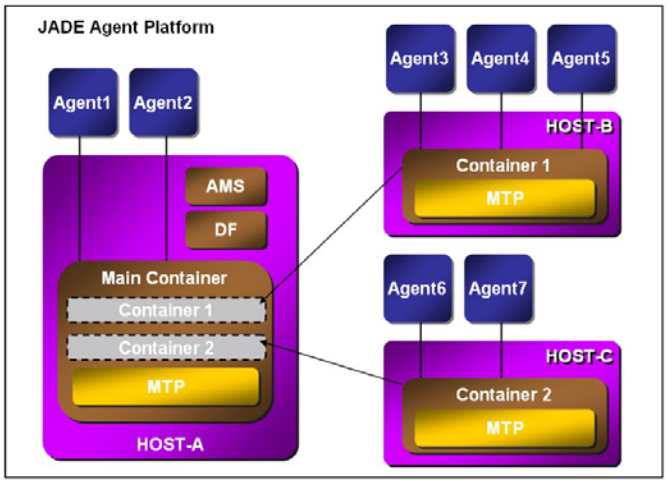

Fig. 3. JADE architecture

B. *FIPAOS*

The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent based applications [8]. FIPA-OS is an agent platform developed by the international standard based on FIPA specification. The architecture of FIPA-OS consists of Directory Facilitator (DF), Agent Management Service (AMS), and Message Transport Service (MTS) [5]. And the mechanism of service discovery is similar to SLP [7]. First, a provider agent of specific service registers its service on the DF. After a consumer agent, want to use the service, acquires the information of the service by searching the DF. And then, the consumer agent communicates with the provider agent through the MTS. The protocols of communication are HyperText Transfer Protocol (HTTP), Internet Inter-ORB Protocol (IIOP), Remote Method Invocation (RMI), and etc. Figure 2 illustrates the system architecture of FIPA-OS and working procedures.

The current version of FIPA-OS agent platform does not provide the implementation of discovery middleware. In this paper, we implement the prototype system of discovery middleware on the FIPA-OS agent platform and provide the interoperability to heterogeneous agent platforms.

C. *JADE*

JAVA Agent DEvelopment Framework (JADE) is a software framework fully implemented in Java. It allows reducing the time-to-market for developing distributed multi-agent applications by providing a set of ready and easy-to-use functionalities that comply with the standard FIPA specifications and a set of tools that supports the debugging and monitoring phases [13]. The architecture of JADE consists of AMS, DF, and Containers likely FIPA-OS. Each running instance of the JADE runtime environment is called a Container as it can contain several agents. And a Container contains functions of communication such as MTS and ACC. The set of active containers is called a Platform. A single special Main container must always be active in a platform and all other containers register with it as soon as they start. It follows that the first container to start in a platform must be a main container while all other containers must be non-main containers and must be told where to find their main container. The AMS and DF must be located in Main container and treat a service to several operations as follow 'registration', 'deletion', 'modification', and 'searching' through the DF. Figure 3 illustrates the system architecture of JADE.

The current version of JADE agent platform does not provide the implementation of discovery middleware such as FIPA-OS. Therefore it needs additional expansions.

III. Discovery Middleware

Figure 4 illustrates the service discovery mechanism suggested by FIPA [6]. Discovery middleware is used to use services registered in heterogeneous agent platforms. DM module makes it possible to use services without any consideration of interoperability between agent platforms. The Agent Discovery Service (ADS) provides discovery functionality in ad hoc networks, in which network nodes join or leave more frequently or less frequently. It provides a high-level DF-like interface for agents, while taking advantage of various discovery middleware, depending on the underlying ad hoc technology. An AP optionally hosts a DF. If an ADS is present on the AP, the DF should only be used for handling df-agent-descriptions related to the local AP. The ADS should only be used by agents of the local AP for provision of their df-agent-descriptions to the ad hoc network as well as for

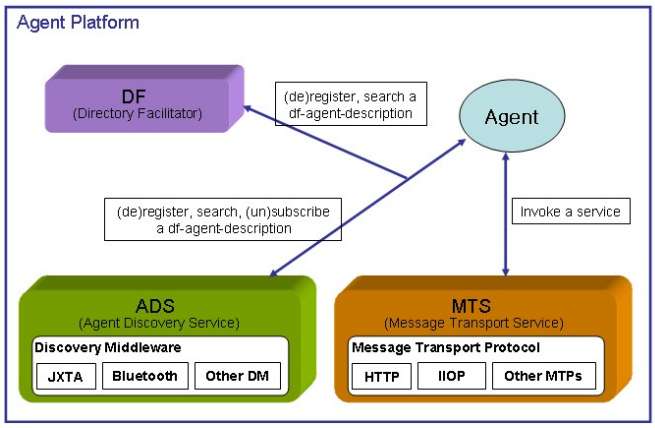

Fig. 4. A service discovery mechanism using discovery middleware

discovery of agents on remote devices in the ad hoc network, i.e. the ADS allows local agents to be discovered by agents on remote devices and vice versa. This means that the DF should provide a yellow pages service restricted to the scope of the local AP, and the ADS provides a yellow pages service restricted to the scope of the ad hoc network. The ad hoc network accessible via the ADS is a compound of all ad hoc networks supported by the maintained DM technologies, for instance JXTA or Bluetooth [9].

However the implementation of discovery middleware is not easy. There are several considerations. For example, it needs communication protocols between heterogeneous agent platforms, and also needs data consistency mechanisms to manage inconsistency caused by service registration, deletion, and modification. Therefore, it is a trend of development that considers a specific subject of agent platform. In this paper, our approach concentrates on providing a real service between heterogeneous agent platforms.

A. *System Design*

In following paragraphs will represent the considerations of design a discovery middleware.

First, it should be satisfied data consistency between heterogeneous agent platforms. The problem of data inconsistency is inevitably occurred between agent platforms when use operations as follow registration, deletion, and modification. This problem is ongoing study, and we consider several solutions such as using a time stamp, real-time synchronization of database, and etc. Using the time stamp method, it attaches a time stamp to each service when it created. It is not available to use, if a service pass the validate time. The real-time synchronization method is that changes in one database are reflected in the other database.

Second, it needs a standard definition of resources such as service, device, agent, and etc. The difference of resource description between JADE and SLP should be mapped a standard description, because JADE and SLP have there's own description. Therefore we define ontology for resource description as shown in the table 1. In our system, we represent

TABLE I
ONTOLOGY FOR RESOURCES

Subject	Predicate	Object	Graph ID
PrintAgent1	Title	Print agent	11
PrintAgent1	Type	Agent	11
PrintAgent1	Serve	Color printing service	11
PrintAgent1	Device Location	Room 27310	11
PrintAgent1	Availability	Yes	11
PrintAgent2	Title	Print agent	11
PrintAgent2	Type	Agent	11
PrintAgent2	Serve	Laser printing service	11
PrintAgent2	Device Location.	Room 27309	11
PrintAgent2	Availability	Yes	11

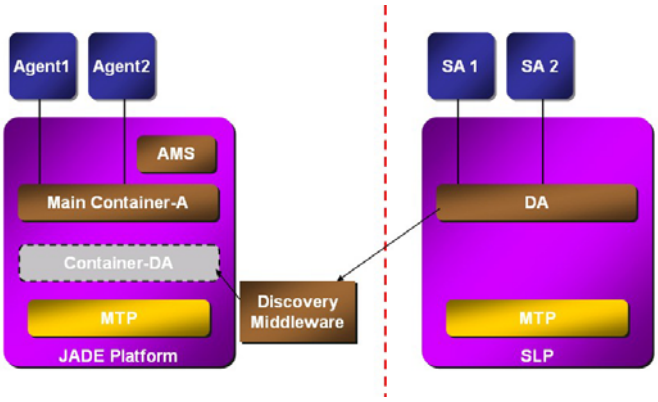

Fig. 5. The architecture of DM using SLP

resources in first-order predicate calculus. The basic model consists of Subject, Predicate, Object, and GraphID, in which

- Subject : the set of subject names (for example, service, agent, device, person, or etc)
- Predicate : the set of predicate names (for example, is located in or has status)
- Object : the set of all values of Subjects (for example, Room 27, Yes, Agent, or empty)
- GraphID : the number linked with an OWL file

To illustrates, (PrintAgent1, Type, Agent, 11) means that the type of PrintAgent1 is an agent, and linked with an OWL file #11.

Finally, it needs a defined method of service invocation between JADE and SLP. It needs some additional works, because agent platforms have there's own service invocation method each other. This part is ongoing study. We consider several methods about web application or socket invocation or etc.

B. *Implementation of DM for interoperable between SLP and JADE*

Figure 5 illustrates our architecture of discovery middleware using JADE and SLP. According to the FIPA specification, a service discovery of heterogeneous agent platforms passes

through many communication procedures, because an agent platforms have service information its own database. If an agent an agent query a specific service to ADS, ADS requests the same query to DM. And then DM requests the service to heterogeneous agent platform. After, reply procedures inverse of request procedures. Precisely, Agent ↔ ADS ↔ DM ↔ AP.

Our architecture of DM reduces replying time of query, because it has simpler discovery procedures than current procedures. Figure 5 illustrates our mechanism that stores services of SLP on JADE agent platform in advance such as container. It simplifies communication procedures each time of requests. When an agent needs to discover a certain service, it first looks at its local container to check whether that service is available. On failure, by looking at its own container for heterogeneous agent platforms, it checks the service of SLP to discovery the service. The service information of providing SLP is retrieved to JADE agent platform when JADE agent platform or SLP starts its running. After the service information of SLP is periodically synchronized with container of JADE agent platform.

IV. Conclusions

In ubiquitous computing environments to use distributed services and devices the functions of service discovery should be provided about types or locations of services. In this paper, we developed the prototype of ontology based discovery middleware to facilitate sharing services between heterogeneous agent platforms. Using this prototype make it possible to construct multi agent platform systems. And our architecture of DM reduces replying time of query, because it more simplifies discovery procedures than suggested by FIPA. However several problems should be proved for providing the complete service such as data consistency, communication protocol, standard service description, and SLP applications.

In future, we concentrate on development of a real service application and proving data inconsistency. And we'll extend our discovery middleware to conform FIPA-OS agent platform. Additionally, we are also interested in context-aware service discovery.

References

[1] James Kempf, Pete St. Pierre “Service Location Protocol for Enterprise Networks” Wiley Computer Publishing.

[2] N. Davies, A. Friday, S. P. Wade and G. S. Blair, L2imbo : A distributed systems platform for mobile computing, Mobile Networks and Applications, pp. 143-156, August 1998.

[3] M. Weiser, “Some Computer science issues in ubiquitous computing” Communications of the ACM, pp. 75-84, July 1993.

[4] N. Gibbins, W. Hall, “Scalability Issues for Query Routing Service Discovery”, Proceedings of the Second Workshop on infrastructure for Agents, MAS and Scalable MAS, 209-217, May 2001.

[5] FIPA Agent Management Specification. Foundation for Intelligent Physical Agents, 2004. http://www.fipa.org/specs/fipa00023

[6] FIPA Agent Discovery Service Specification. Foundation for Intelligent Physical Agents, 2003. http://www.fipa.org/specs/fipa00095

[7] FIPA-OS V2.1.0 Distribution Notes. http://fipa-os.sourceforge.net

[8] Foundations for Intelligent Physical Agent. http://www.fipa.org

[9] FIPA JXTA Discovery Middleware Specification. Foundation for Intelligent Physical Agents, 2003. http://www.fipa.org/specs/fipa00096

[10] E. Guttman, “Service location protocol: Automatic discovery of IP network services”, IEEE Internet Computing, pp. 71-80, July 1999.

[11] “Univeral plug and play forum.” 2001. http://www.upnp.rog

[12] C. Bettstetter and C. Renner, “A comparison of service discovery protocols and implementation of service location protocol.” In Proceedings EUNICE 2000, (Twente, Netherlands), Sept, 2000.

[13] JADE Tutorial Giovanni Caire. http://jade.tilab.com

[14] Project JXTA. http://www.jxta.org

Sejung Oh received the B.E degree in Engineering of Advanced Materials from Sungkyunkwan University, Korea, in 2003.

He is currently a M.E. student in the Department of Computer Engineering in Sungkyunkwan University. His main areas of research include service discovery, agent platform, context-aware system and ontology building. For service discovery, he is interested in ontology based service discovery, discovery middleware for sharing services between heterogeneous agent platforms.

Juryon Paik received the B.E. degree in Information Engineering and the M.E. degree in Computer Engineering from Sungkyunkwan University, Korea, in 1997 and 2005, respectively.

She is currently a Ph.D. student in the Department of Computer Engineering in Sungkyunkwan University. Her main areas of research include xml mining, xml data replication and ontology building. For xml mining, she is interested in mining frequent tree patterns, mining association rules from xml documents and similarity computation between xml documents.

Heeyong Youn received the BS and MS degrees in electrical engineering from Seoul National University, Seoul, Korea, in 1977 and 1979, respectively, and the PhD degree in computer engineering from the University of Massachusetts at Amherst, in 1988. From 1979 to 1984, he was on the research staff of Gold Star Precision Central Research Laboratories, Korea.

He is presently an endowed professor in the School of Information and Communication Engineering, Sungkyunkan University, Suwon, Korea. His research interests include storage systems, distributed computing and networking, Internet and mobile computing, and fault-tolerant computing. He has published more than 120 papers in international journals and conference proceedings, and received an Outstanding Paper Award from the 1988 International Conference on Distributed Computing Systems, 1992 Supercomputing, and the 2001 Korean Society Internet Information Spring Symposium, respectively. He also served as a lecturer of the ACM Lectureship Series from 1993 to 1997.

Dr. Youn is a senior member of the IEEE. Since 2002, Prof Youn works for ***21C Frontier Ubiquitous Computing and Networking project*** (supporting by ministry of information and communication republic of Korea) and ***Korea Research Foundation project*** (KRF – 2003 – 041 –D20421) and ***Brain Korea 21 Project***.

Ungmo Kim received the B.Sc. degree in the Department of Mathematics in Sungkyunkwan University, Korea, in 1981, and the M.Sc degree in Computer Science from Old Dominion University, USA, in 1986, and the Ph.D. degree in Computer Science from Northwestern University, USA, in 1990. He has worked several years as consultant and analyst in universities and private companies, and since 2001, he has been working for the Silver-Net Campaign Society as a Regional Chief-in-Committee Member.

He is a Professor in the School of Information and Communication Engineering in Sungkyunkwan University, Korea. His current research interests include XML/Web mining, Data Mining, Access Control, and Domain-Specified ontology modeling.

Dr. Kim is the Inquiry and Consulting Member of Korea Information Processing Society.