

# 서비스 예약 기능을 지원하는 에이전트 기반의 서비스 저장소의 설계

\*나성준 윤여진 김동욱 신동렬

성균관대학교 정보통신공학부

\*nsj3802@ece.skku.ac.kr

## Design of Agent-based Service Repository supporting Service Reservation

\*Sung-Jun Na, Dong-Uk Kim, Yoe-Jin Yoon, Dong-Ryeol Shin

School of Information and Communication Engineering, Sungkyunkwan University

### 요약

멀티 에이전트 시스템 환경에서 에이전트들은 자신이 제공하는 서비스에 대해 일정한 형태로 멀티 에이전트 시스템의 저장소에 서비스를 등록, 삭제, 검색할 수 있는 기능을 갖는다. 효율적인 서비스의 저장소인 DF (Directory Facilitator)는 에이전트들이 제공하는 서비스에 대한 옐로우 페이지 서비스를 제공한다. 에이전트 간의 협업을 위한 에이전트 플랫폼에서 서비스 디스커버리 기능을 지원하기 위한 DF 개발은 각각의 서비스를 지원하는 에이전트와 사용자 에이전트간의 서비스를 유지하여야 한다.

본 논문에서 제안하는 DF는 서비스 예약 기능을 제공함으로써 사용자 에이전트에게 좀 더 편리한 서비스 검색 메커니즘을 제공하는 것에 목적을 두고 있다. DF로 사용자 에이전트가 서비스를 등록 했을 경우에 Reservation Repository를 통하여 그 등록된 서비스가 다른 사용자 에이전트가 제공받고자하는 서비스인 경우를 인지했을 때 Reservation Repository에 등록된 에이전트주소와 서비스이름을 토대로 하여 예약한 사용자 에이전트와 서비스를 제공하게 한다.

### 1. 서론

유비쿼터스 환경이 발전함에 따라 다양한 기반 기술들에 대한 연구가 활발히 진행되고 있다. 특히, 유비쿼터스 환경에서 필수적인 기술로 많은 연구가 이루어지고 있는 에이전트 기술은 다양한 객체들이 기능을 가지고 행동할 수 있도록 하는 토대를 마련하고 있다. 이러한 에이전트 기술의 발달과 더불어 에이전트의 국제 표준을 제정하는 단체인 Foundation for Intelligent Physical Agents (FIPA)[1]에서는 이러한 다양한 에이전트들 간의 통신을 위한 통신 규약 및 에이전트 아키텍처 등을 제시하고 있다. 이러한 활발한 활동을 통해, 유비쿼터스 환경에서는 다양한 장치에 에이전트 기술을 도입하여 사용자에게 지능적이고, 위치 독립적인 서비스를 제공하기 위한 연구가 진행되어 왔으며 현재 이러한 FIPA 표준에 부합하는 다양한 멀티 에이전트 시스템이 존재한다. 그 중에서 Java Agent DEvelopment Framework (JADE)[2]는 학계나 연구소에서 가장 주목을 받고 있는 에이전트 시스템이다. 현재 JADE와 여러 종류의 다양한 FIPA표준에 부합하는 멀티 에이전트 시스템의 서비스 예약을 위한 연구가 활발히 진행 중에 있다.

현재의 멀티 에이전트 시스템에서 서비스를 제공하는 방법은 단순한 Publish and Subscribe 방식을 통하여 서비스를 제공하지만 이러한 방식으로는 서비스를 제공하는 데에 한계점이 있다. 그 한계점은 Publish가 되지 않은 상태의 서비스를 제공받으려 하는 경우에 일어날 수 있는 문제를 말한다. 즉, 서비스 저장소(CALM-based DF)에 Publish된 서비스만을 제공받을 수 있는 문제를 야기하는 것이다. 하지만 서비스 예약 기능을 사용하는 경우 이러한 문제를 예약이라는 기능을 사용하여 현재는 Publish되지 아니한, 즉 서비스 저장소에 서비

스가 존재하지 않는 경우에도 추후에 서비스를 Publish할 경우에 사용자가 원하는 서비스를 제공받을 수 있다. 따라서 현재의 멀티 에이전트 시스템에서는 서비스와 서비스 저장소에 대하여 신뢰성과 효율성을 위한 서비스 예약 메커니즘이 필요하다.

본 논문은 다음과 같이 구성된다. 2장에서는 본 논문에서 관련연구의 주제인 JADE 에이전트 시스템 및 Directory Facilitator에 대하여 구체적으로 소개한다. 3장에서는 본 논문에서 제안한 서비스 예약 메커니즘과 서비스 예약 메커니즘을 반영한 CALM-based DF의 아키텍처에 대하여 소개한다. 마지막으로 4장에서는 결론 및 향후 연구방향에 대하여 기술한다.

### 2. 관련연구

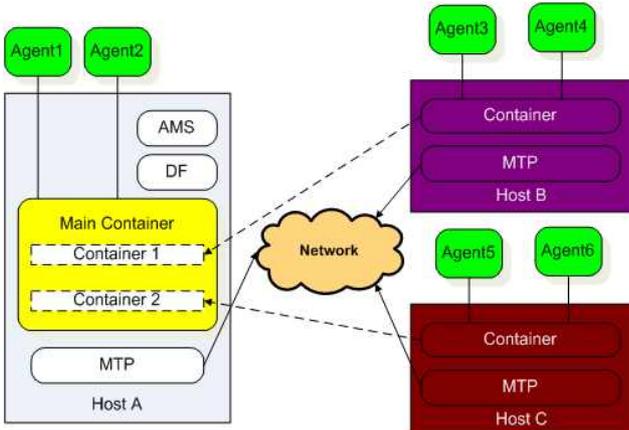
#### 가. JADE

JADE는 P2P 통신 아키텍처를 기반으로 하는 JAVA 기반의 분산 멀티 에이전트 어플리케이션의 개발을 위해 TILAB에서 개발된 에이전트 기반 미들웨어이다. 이 JADE 미들웨어를 사용하면 고정된 네트워크안의 컴퓨터나 모바일 터미널 상에서 정보, 자원, 제어 등이 고정 네트워크안의 컴퓨터들뿐만 아니라 이동 단말기에도 분산되어 수행될 수 있다. 이러한 JADE 기반 에이전트는 어플리케이션, 사용자 또는 환경의 요구나 필요에 따라 생성, 소멸되기도 한다.

JADE 플랫폼을 통한 peer들 사이의 통신은 유무선 네트워크 환경에 관계없이 에이전트간의 FIPA-ACL 메시지를 통한 메시지 과실 방식으로 통신이 이루어진다. FIPA 표준에 부합되는 JADE 미들웨어의 구성은 그림 1에서 보듯이 컨테이너, 플랫폼, AMS(Agent Management System), DF(Directory Facilitator) 등이 있다. 다양한

에이전트들을 포함하는 것을 컨테이너라 하며, 활동적인 컨테이너들의 집합을 플랫폼이라 한다[3].

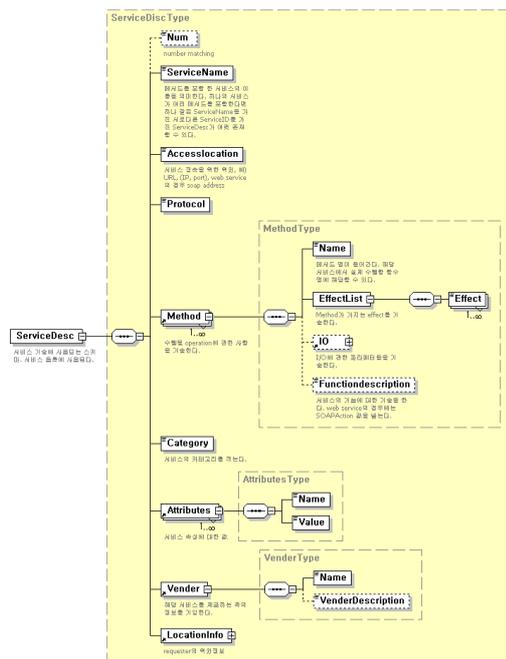
그림 1은 FIPA 표준에 부합하는 에이전트 시스템의 구조를 보여 준다. 그림 1에서 보는 것과 같이 에이전트 시스템은 다양한 컨테이너로 구성되며, 그 중 Main Container는 AMS 에이전트와 DF 에이전트를 가지고 있다. AMS는 플랫폼 내의 서로 다른 에이전트들이 각각의 독립적인 이름을 가질 수 있도록 하는 네이밍 서비스를 제공하며, 에이전트의 등록, 구동, 종료 등의 전반적인 관리를 제공한다. 또한 DF는 각 에이전트들이 제공하는 서비스에 대하여 yellow pages 서비스를 제공한다. 즉, 에이전트가 제공하는 서비스에 대한 서비스 저장소의 역할을 수행한다.



(그림 1) JADE Platform

#### 나. Directory Facilitator

DF는 다양한 유저 에이전트들의 서비스를 등록하고, 해제하는 기능을 수행하는 서비스 저장소이다. 본 논문에서 제안하는 메신저 서비스를 위해, JADE DF는 다양한 유저 에이전트의 lookup 기능을 수행하기 위한 저장소로서 사용된다. 에이전트와 DF 사이의 통신 및 이기종 에이전트 시스템간의 통합과 웹 서비스를 연동할 수 있도록 하는 XML 메시지를 사용하며, 그림 2는 DF에 사용자 에이전트가 서비스를 등록하기 위한 XML 메시지의 스키마에 대한 예를 보여준다.



(그림 2) 서비스 등록을 위한 XML 스키마

그림 2에서 나타난 바와 같이 서비스 등록을 위한 스키마에 대응하는 XML을 표현한 그림 3은 실질적으로 Description Repository에 저장되는 서비스를 나타낸 것이다. 이러한 XML은 XML Parser를 통하여 XML 스키마의 유효성 검증 단계를 거쳐 DF Function의 기능을 수행하게 된다. 서비스 등록 메시지인 경우는 Register의 기능, 서비스 삭제 메시지인 경우는 Deregister, 서비스 검색 메시지인 경우는 Search의 기능을 수행한다.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!--Sample XML file generated by XMLSpy v2006 rel. 3 sp1 (http://www.altova.com)-->
3 <ServiceDesc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 xsi:noNamespaceSchemaLocation="**Service**servdesc.xsd">
5 <ServiceName>SCCTV</ServiceName>
6 <Accesslocation>http://202.30.22.170:8080/SCCTV/</Accesslocation>
7 <Protocol>WebService</Protocol>
8 <Method>
9 <Name>viewCamera</Name>
10 <EffectList>
11 <Effect>E-DisplayVideo</Effect>
12 </EffectList>
13 <IO>
14 <inputs>
15 <parameter>
16 <name>ID</name>
17 <type>String</type>
18 <description>Parking-Lot CCTV ID</description>
19 </parameter>
20 </inputs>
21 <outputs>
22 <parameter>
23 <name>Result</name>
24 <type>String</type>
25 <description>Invocation Result</description>
26 </parameter>
27 </outputs>
28 </IO>
29 </Method>
30 <Category>ControlCenter</Category>
31 <Attributes>
32 <Name>String</Name>
33 <Value>String</Value>
34 </Attributes>
35 <Vender>
36 <Name>CUS</Name>
37 <VenderDescription>Ubiquitous System</VenderDescription>
38 </Vender>
39 <LocationInfo>
40 <Fixed>
41 <L_Position>ControlCenter</L_Position>
42 </Fixed>
43 </LocationInfo>
44 </ServiceDesc>

```

(그림 3) 서비스 등록을 위한 XML

### 3. CALM-based DF

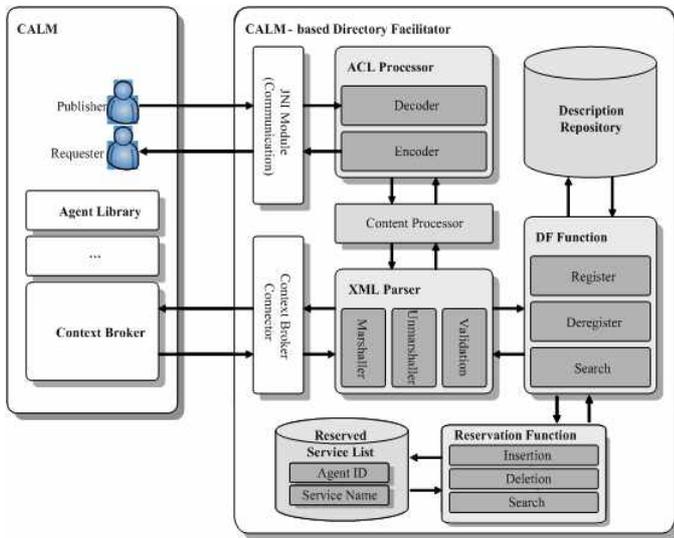
#### 가. Architecture

CALM-based DF는 FIPA 표준에 부합하는 ACL 메시지 내부의 Content 메시지에 XML(eXtensible Markup Language)를 사용하여 서비스 등록, 검색, 삭제할 수 있는 세 가지 커뮤니케이션 기능들로 이루어진다. 서비스 검색을 위한 다양한 검색 방식을 지원하는 DF를 개발하기 위해서는 XML 메시지를 이용하여 서비스 등록 및 검색 시 각각의 해당하는 element를 참조한다. 그림 4는 CALM-based DF의 아키텍처를 보여준다. CALM-based DF 내부에는 각각의 기능을 수행하는 모듈들이 존재한다. 이러한 내부 모듈에 대해서는 아래에서 살펴보기로 한다.

#### 나. ACL Processor

CALM-based DF는 FIPA 에이전트 플랫폼에 있는 다른 에이전트들과 마찬가지로 FIPA ACL 메시지 구조를 표준 커뮤니케이션 방법으로 사용한다. ACL Processor의 역할은 에이전트가 보낸 메시지를 ACL 메시지만을 추출하는 기능을 제공하는 Decoder의 역할과 DF내 부처리에 의해 생성될 ACL 메시지를 조합하는 Encoder의 역할로 나뉘어지게 된다.

본 논문에서 제안한 DF의 커뮤니케이션 또한 ACL 메시지로 통신을 하게 되고, 에이전트들은 세 가지 커뮤니케이션 기능들로 DF와 상호작용할 수 있다. 이 기능들에 대해서는 DF Function절에서 언급한다.



(그림 4) Architecture of CALM-based DF

### 다. XML Parser

XML Parser는 binding compiler로써 XML schema를 자바 코드 형식으로 변환할 수 있게 한다. XML Parser 모듈의 수행하는 기능으로는 세 부분으로 나누고 각각의 기능들은 Marshal, Unmarshal, Validation으로 정의된다. 각각의 기능을 살펴보면 다음과 같다.

Marshal의 기능을 살펴보면 Context objects로부터 XML 문서로 변환하는 과정을 marshal이라고 하며 ObjectFactory class, Set method등을 이용하여 출력될 문서의 형태를 정하고, 비순차적으로 필요한 데이터에 접근할 수 있다. Unmarshal의 기능은 Unmarshal 과정을 거쳐 생성된 context objects는 문서의 내용을 체계적으로 나타내어주므로 빠르고 쉽게 해당 데이터에 접근할 수 있도록 해 준다. javax.xml.bind, javax.xml.util, javax.xml.helper 와 같은 JAXB API 사용하여 JAXBContext, Unmarshaller와 같은 클래스를 생성한 뒤 get, set method 이용하여 data를 쉽게 조작, 접근할 수 있다. XML Parser의 마지막 기능인 Validation은 Marshal, Unmarshal 중에 입력 데이터와 출력데이터가 올바르게 작성되어있는지 확인할 수 있다. 사용자의 필요에 따라 수행할 수 있으며 유효성의 여부가 중요하지 않을 경우 유효하지 않은 점이 발견되었다 하더라도 프로세스는 계속 진행 가능하게 하는 유연성이 있다.

### 라. DF Function

등록(Register)기능을 수행하면 새로운 서비스 항목을 DF의 Knowledge Base에 등록시킨다. DF는 에이전트가 등록하고 있는 정보의 정확성은 보장하지 않는다. 등록해제(Deregister)의 기능을 수행하면 에이전트는 DF의 Knowledge Base에 있는 항목을 제거할 것을 요청할 수 있다. 이 기능은 에이전트와 관련된 정보를 중개하는 DF의 의무를 없앤다. 검색(Search)의 기능을 수행하면 에이전트는 특정 서비스를 만족시킬 수 있는 한 개 이상의 에이전트들을 찾도록 DF에 요청할 수 있다. DF는 이 검색 기능에 반응하여 제공되는 정보의 정확성에 대해서는 보장할 수 없다. 즉, 악의적인 에이전트 또는 신뢰할 수 없는 에이전트가 제공하는 서비스에 대한 책임에 대해서는 보장할 수 없다.

### 마. Reservation Function

Reservation Function은 현재 서비스 저장소에 존재하지 않는 서비스를 예약하여 추후에 그 존재하지 않았던 서비스가 등록요청이 있을 경우 종전에 서비스를 요청한 에이전트에게 예약된 서비스를 제공하는 방식을 사용하고 있다. 이러한 서비스 예약 기능을 제공하는 이유는 어떠한 서비스를 요청한 에이전트는 자신이 원하는 서비스가 없어서 이용을 하지 못하게 된다. 또한 그 에이전트는 원하는 서비스를 얻기 위하여 지속적으로 DF에 질의를 하게 되는 불필요한 과정을 거치게 된다. 하지만 예약 기능을 이용하면 지속적으로 질의하는 불필요한 과정을 거치지 않게 되므로 에이전트가 목적이 다른 작업을 좀 더 편하게 진행할 수 있다.

Reservation Function 모듈의 주된 기능은 세 가지의 기능으로 나눈다. 첫 번째의 기능은 Insertion으로 DF Function 모듈에서 Search 기능을 수행한 결과로 에이전트가 원하는 서비스가 DF 내부의 KnowledgeBase에 존재하지 않을 경우에 에이전트가 원하는 서비스를 예약하는 기능을 수행한다. 서비스를 예약하기 위해서는 예약하려는 서비스 이름과 서비스를 필요로 하는 에이전트의 이름(Agent ID)를 반드시 Reserved Service List에 저장하여야 한다. Deletion의 기능은 예약된 서비스를 에이전트에게 제공한 후 더 이상 필요성이 없어지게 되므로 해당하는 예약된 서비스는 삭제한다. 마지막으로 Search 작업은 에이전트가 원하는 서비스가 DF 내부의 KnowledgeBase에 등록이 완료가 된 후에 예약된 서비스 리스트에서도 예약을 했던 서비스가 존재하는지 검색하여 판단한다. 예약된 서비스 리스트에서 해당하는 서비스 이름이 검색되면 해당하는 에이전트에게 바라던 서비스를 제공한다. 하지만 예약된 서비스 리스트를 검색을 했을 때 서비스가 검색이 되지 않았을 때는 해당 에이전트가 예약한 서비스가 없는 경우이므로 아무런 기능을 수행하지 않게 된다. 이러한 세 가지 기능들은 DF Function 모듈과 Reservation Function 모듈, Reserved Service List와의 협동에 의하여 수행된다.

### 4. 결론

본 논문에서 제안하는 DF는 Reservation Function을 통하여 추후에 사용될 서비스를 예약하는 기능을 제공한다. 에이전트는 원하는 서비스를 얻기 위하여 지속적으로 DF에 질의를 하게 되는 불필요한 과정을 생략하게 되므로 에이전트가 다른 목적을 가진 작업을 좀 더 편하게 진행할 수 있다. 이는 서비스의 예약 기능을 통하여 유비쿼터스 환경에서 사용자에게 보다 편리하고 정확한 서비스를 제공한다. 또한 서비스 예약과정에서 에이전트 ID와 서비스 이름의 예약을 지원함으로써 인하여 검색 결과에 대한 신뢰성을 보장함으로써 서비스 저장소의 표준화가 가능해질 전망이다.

### Acknowledgement

"본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅및네트워크원천기술개발사업의 지원에 의한 것임"

### 참고문헌

- [1] Foundation for Intelligent Physical Agents, <http://www.fipa.org>
- [2] Fabio Bellifemine, Federico Bergenti, Giovanni Caire and Agostino Poggi: JADE-A Java Agent Development Framework,

Multi-agent Programming, LNCS, Vol. 15, Springer-Verlag, Berlin Heidelberg US, 125-147

[3] F. Bellifemine, G. Caire, A. Poggi and G. Rimassa: JADE A White Paper, <http://jade.tilab.com>

[4] Giovanni Caire JADE TUTORIAL JADE PROGRAMMING FOR BEGINNERS 2003

[5] Giovanni Caire, "LEAP USER GUIDE", March 2005.