

ebXML 기반의 에이전트 등록저장소 시스템의 설계와 구현

(Design of Agent Registry / Repository System based on ebXML)

김 일 광 [†] 이 재 영 [†] 김 일 곤 ^{††}
(Il Kwang Kim) (Jae Young Lee) (Il Kon Kim)

요 약 본 연구의 목적은 글로벌 인터넷 환경에 부합할 수 있는 새로운 형태의 에이전트 등록/저장소 시스템을 설계하는 것이다. 이를 위해, 먼저 ebXML, UDDI, FIPA DF와 같은 핵심 기술에 대한 사전연구를 진행하였고 그 결과로 ebXML을 활용한 에이전트 등록 및 저장소 시스템을 설계하였다. 제안하는 시스템은 에이전트의 등록과 저장, 검색을 위해 ebXML이 어떻게 활용될 수 있는지를 보여주며, 서로 다른 에이전트 플랫폼에서 개발된 에이전트가 어떠한 방법과 절차를 통해 등록되고 검색될 수 있는지를 설명한다. 본 시스템을 활용할 수 있는 전형적인 예를 보여주고, 본 시스템의 장점을 기술한다.

키워드 : 에이전트, 모바일에이전트, ebXML 레지스트리, 리파지토리, ebXML 등록저장소

Abstract The goal of this paper is to propose a way to register agents so that makes users or autonomouagents to find target agents more conveniently and efficiently. In order to achieve this goal, we first survey the key technologies such as UDDI, ebXML, FIPA DF. As the results, we designed a new agent registry/repository system based on ebXML technologies. We introduce our agent registry/repository system for wide area network and describe how to use our system for agent registering and searching target agents for communications. Finally, we give an example to illustrate a typical scenario in which user receives the results from our proposed agent registry/repository system and evaluate its characteristics in terms of advantage.

Key words : Agents, Mobile Agents, ebXML, UDDI, Registry, Repository

1. 서 론

웹을 기반으로 한 인터넷 환경은 네트워크를 통해 다양한 온라인 서비스를 이용할 수 있게 해 주었다. 더불어 컴퓨팅 환경도 웹을 중심으로 빠르게 변화되어 가고 있으며 네트워크와 관련된 많은 기술들이 지속적으로 연구되고 있다. 물론, 그러한 바탕에는 네트워크를 근본으로 하는 분산 컴퓨팅 환경이 존재하고 있으며 CORBA, Java RMI를 거쳐 최근에는 SOAP을 기반으로 한 웹 서비스 더 나아가 Grid Computing에까지 이

르렀다. 그리고, 이러한 분산환경의 패러다임이나 추세와 함께 많은 연구가 진행되어 온 분야 중 하나는 바로 에이전트 분야이다. 특히, 에이전트와 에이전트 플랫폼 그리고 이동 에이전트에 대한 연구와 그 유용성은 이미 많은 논의를 거쳐 확립되어 왔다. 그림 1은 에이전트

· 본 연구는 보건복지부 보건의료 기술진흥사업의 지원에 의하여 이루어진 것임 (EHR사업단 과제번호: A05-0909-A80405-05N1-00000A)

[†] 비 회 원 : 경북대학교 IHIS 연구원
dinoso@daum.net
intvis@lycos.co.kr

^{††} 정 회 원 : 경북대학교 전자전기컴퓨터학부 교수
ikkim@knu.ac.kr
논문접수 : 2006년 1월 25일
심사완료 : 2006년 3월 24일

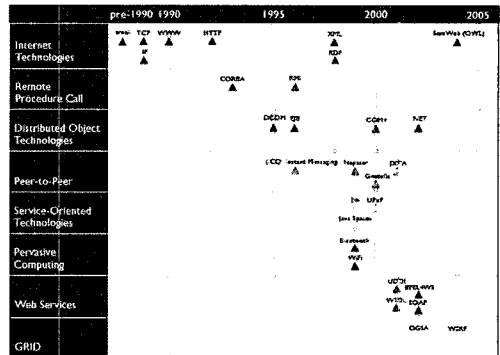


그림 1 에이전트 관련 기술 로드 맵

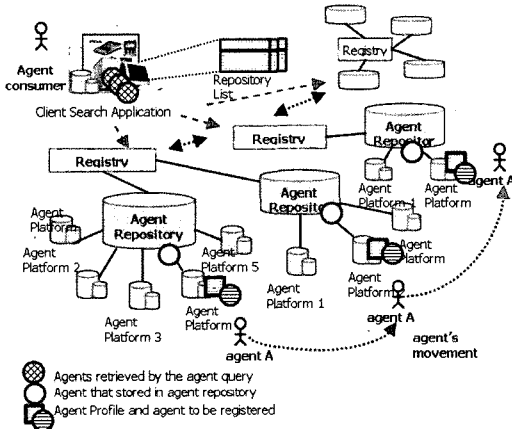


그림 2 Distributed Agent Registry/Repository Service

인프라 구축과 관련되어 있다고 할 수 있는 기술들의 로드 맵을 보인 것이다[1]. 이러한 기술들은 에이전트 인프라 구축을 위해 다양한 목적으로 사용될 수 있다. 이미, IBM Aglet, Ajanta, Voyager, GrassHopper, Tracy 등 다수의 에이전트 플랫폼들이 이러한 기술들을 기반으로 하여 연구 개발되어 왔고, 최근에는 에이전트 기반의 상황인식 서비스 개발에 대한 연구도 행해지고 있다[2-5].

이러한 움직임은 가까운 미래에 에이전트에 의해 모든 일을 처리하고, 에이전트가 제공해 주는 서비스를 이용하게 되는 에이전트 시대를 예고하고 있다. 본 논문에서는 이러한 움직임과 관련하여 다수의 플랫폼에서 생성된 에이전트를 식별, 등록, 저장하기 위한 목적으로 ebXML 기반의 새로운 에이전트 등록 및 저장소 시스템을 제안한다.

제안하는 시스템은 일종의 디렉토리 서비스와 같은 것으로 서로 다른 멀티 플랫폼 환경(OS, Machine, Agent System, etc)에서 생성된 여러 에이전트를 식별, 등록, 저장할 수 있고 필요 시 통신 대상 에이전트를 검색할 수 있는 기반환경을 제공해 준다. 그림 2는 본 연구의 최종 목표시스템을 보여준다. 다수의 registry와 repository가 분산되어 있으며 에이전트 플랫폼에서 생성된 에이전트는 repository에 그 코드가 저장되고 관련된 메타정보는 registry에 등록된다. 에이전트의 이동성을 고려하여 agent code의 전송을 지원하고, registry query를 통해 분산된 repository가 하나로 통합연동(federation)될 수 있는 구조이다.

2. 관련연구

2.1 FIPA Specification

본 절에서는 멀티에이전트 플랫폼 표준인 FIPA가 가

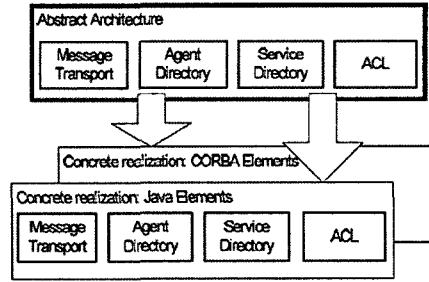


그림 3 FIPA Abstract Architecture

지는 특징을 간단히 기술하고 FIPA DF(Directory Facilitator)와 유사한 개념으로 ebXML 기반의 에이전트 등록, 저장소 시스템을 설계하게 된 배경을 설명한다.

FIPA에서는 FIPA97, FIPA98, FIPA2002를 비롯해 관련 규격을 Applications, Abstract Architecture, Agent Message Communication, Agent Management, Agent Message Transport로 나누어 표준화 작업을 진행해 오고 있는데 그림 3은 FIPA의 Abstract Architecture를 보인 것이다[6].

그림 4는 FIPA의 에이전트 관리 참조모델을 설명한다[7]. 그림을 통해 알 수 있듯이 에이전트 플랫폼은 AMS(Agent Management System), DF(Directory Facilitator), MTS(Message Transport System)으로 구성되어 있다. AMS는 에이전트 관리를 담당하고, DF는 옐로우 페이지(yellow page)와 같은 역할을 한다. MTS는 에이전트 플랫폼 내부의 에이전트뿐 아니라 다른 플랫폼에 존재하는 에이전트간의 메시지 전달을 담당한다. 이때, MTS는 기본적으로 AID(Agent ID)를 보고 대상

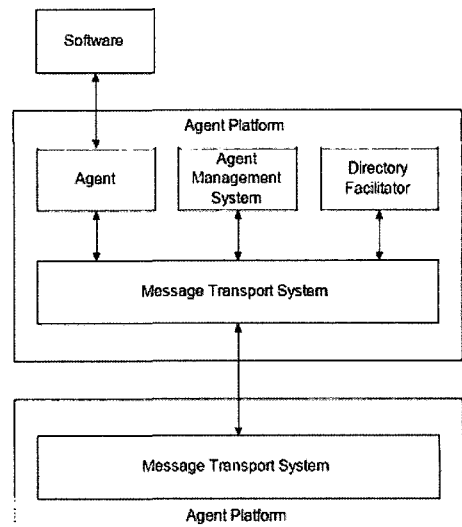


그림 4 FIPA Agent Management Reference Model

에이전트에게 메시지를 라우팅한다. 에이전트의 등록은 AMS를 통해 이루어지고 AMS를 통해 등록된 에이전트만이 에이전트 플랫폼의 서비스를 제공 받을 수 있다. 일단, 등록이 된 후에는 다른 에이전트와 메시지를 주고받거나 DF를 통해 자신을 광고할 수도 있다.

2.2 ebXML vs. UDDI

본 논문에서 제안하는 시스템은 에이전트를 등록, 광고할 수 있다는 측면에서 FIFA DS와 유사한 특성을 갖는다. 하지만, 제안하는 시스템은 사용자 입장에서 생성된 에이전트를 등록, 광고할 수 있을 뿐만 아니라 모바일 에이전트 플랫폼에 적용할 경우 에이전트 코드 서버로서의 역할까지 수행할 수 있는 것이 특징이다.

이러한 목적을 이루기 위해서는 생성된 에이전트를 안전한 방법으로 전송, 저장, 관리할 수 있는 레지스트리/리포지토리 시스템이 필요하다. 특히, 수 많은 에이전트를 잘 분류하고 활용하기 위해서는 효율적 분류スキム을 지원할 수 있어야 한다. 이를 위해 본 연구에서는 그림 5에 보인 ebXML의 RIM(Registry Information Model)을 사용한다[8-11]. ebXML은 2001년 5월 v1.0이 발표된 후 UN/CEFACT와 OASIS라는 국제표준화기구를 중심으로 표준규약 개정 작업을 진행해오고 있으며 사실상 B2B 부문 국제표준으로 인식되고 있다. ebXML은 글로벌하고 단일한 e마켓 플레이스를 만들자는 취지에서 시작됐다. 사실상 이전에도 XML 기반의 B2B 표준들이 존재하고 있었지만(예를 들어 xCBL, RosettaNet, eCo, cXML 등과 같은) 나름대로 규약을

정해 사용하다 보니 로컬 표준간의 정보교환이 힘든 것이 사실이었다. ebXML은 이러한 로컬 표준의 한계를 극복하고 전자거래를 위한 제반 기능, 업무 규약, 통신, 콘텐츠, 비즈니스 형태 등 B2B의 모든 영역에 대한 표준을 제시하였다. 이러한 ebXML 표준 중 RIM은 등록 저장소의 표준적 데이터 구조이며 RS(Registry Service)는 등록저장소에 등록, 조회, 수정, 삭제하기 위한 표준적 인터페이스를 정의한 것이다. RS에는 객체의 생명주기를 관리하는 객체관리서비스(Object Management Service)와 객체를 질의하기 위한 객체질의관리서비스(Object Query Management Service)가 있다. RIM에서 가장 중요한 객체는 RegistryObject와 RegistryEntry이다(그림 5). RegistryEntry는 레지스트리에 제출된 각각의 콘텐츠 인스턴스마다 생성된다. 레지스트리 외부에 저장되어 있는 콘텐츠의 접근경로와 식별자(URI)를 표현하기 위해 External Link가 존재하고 그 연관관계(association)를 지정할 수 있다. Classification과 Classification-Node는 객체들을 효율적이고 계층적으로 분류하기 위한 구조를 정의하기 위해 사용한다. 그 외에 객체에 대한 감사기록을 저장하기 위해 AuditableEvent가 있고 사용자와 기관정보의 저장을 위해 User와 Organization 등의 객체가 존재한다. 초기에, 웹 서비스의 UDDI[12-15] 레지스트리도 고려 하였으나 UDDI 아키텍처의 경우 UDDI 레지스트리에는 일반사항(white page), 산업분류(yellow page), 기술적사항(green page)으로 구분되어 있고 전화번호부와 같은 디렉토리 서비스를 하고

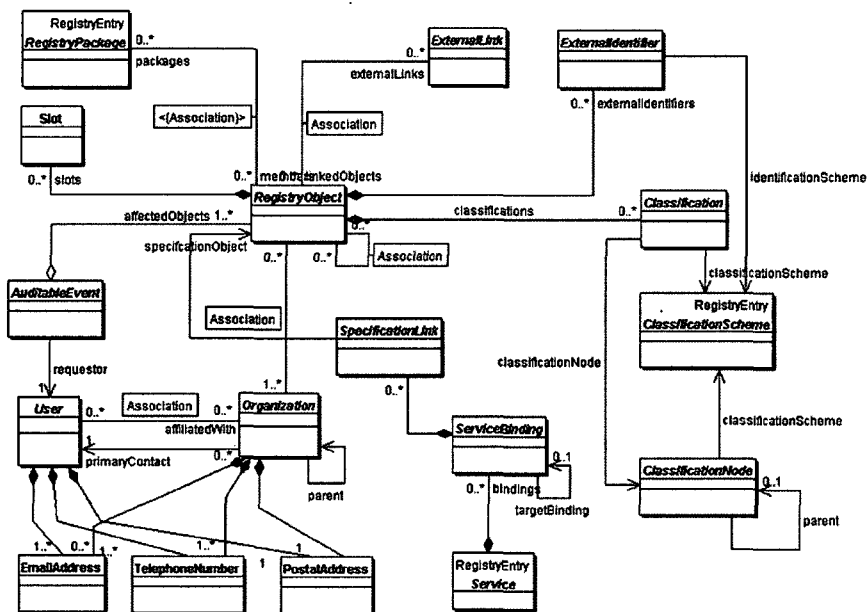


그림 5 ebXML Information Model High Public View

있으나 병원이나 기업의 정보를 보다 상세하게 저장(리파지터리)하고 관리하는 부분은 빠져 있었기 때문에 UDDI 레지스트리보다는 ebXML의 RIM을 적극 활용하게 되었다. 특히, ebXML의 경우는 비즈니스 트랜잭션을 수행하는 데 필요한 구체적인 규약을 보장하고 각종 규약의 상세한 정의를 목표로 하고 있다. 따라서 트랜잭션 처리와 안전한 거래, 시간 제어, 결함 허용(fault-tolerance) 등이 정보교환에 중요한 요소가 될 경우에는 ebXML이 UDDI 보다 다소 유리하다고 판단되었다.

다만, 상호운용성 측면에서 볼 때는 ebXML이나 UDDI와 같이 서로 다른 정보모델을 가진 Registry 간의 상호통합도 필요하다. 이러한 취지에서 ebXML V2.2에서는 다양한 ebXML 레지스트리간의 상호-연동 방법을 제안하고 있으며 SUN에서는 JAXR[16]이라는 스펙도 발표하였다.

3. 시스템 개요

본 연구에서 가정하는 에이전트는 다양한 에이전트 플랫폼에서 여러 가지 목적으로 개발되어 독립 실행이 가능한 능동적인 개체(Entity or Object)로 본다. 사용자 또는 개발자는 특정 에이전트 플랫폼에서 에이전트를 개발할 수 있으며 개발된 에이전트는 유저가 부여한 특별한 목적의 일을 수행하거나 범용적인 서비스를 제공할 수도 있다. 개발자는 공급자가 되어 자신이 개발한 에이전트를 공인된 서버에 등록해 두고 필요로 하는 사람(수요자)에게 제공해 줄 수 있다. 다시 말해, 시스템을 개발할 때 다른 사람에 의해 기 개발된 에이전트를 제공 받아서 자신의 시스템을 완성할 수 있다는 뜻이다. 물론, 이러한 경우에는 에이전트의 다운로드나 사용에 대한 과금정책이 필요할 수 있으며, 에이전트가 제공하는 서비스와 버전 관리도 대단히 중요한 일 중 하나가 된다.

실제로 e-Business 환경에서, 기업의 IT예산 중 가장 많은 부분이 기업규모가 커지면 커질수록 방대한 애플리케이션을 단순히 유지보수 하는데 쓰여지고 있다. 최근 이런 문제를 해결하기 위해 애플리케이션이 말고 있던 부분을 서비스화 혹은 컴포넌트화하여 시장상황에 따라 시시각각 변화하는 비즈니스 프로세스에 유연하게 결합하기 위한 대안으로 SOA(Service Oriented Architecture)가 주목을 받고 있다. 하지만, 이러한 SOA 기반의 환경에서도 개발자의 많은 노력은 서비스개발뿐만 아니라 비즈니스 정보와 관련 서비스의 발견이 된다. 또한, 현재의 e-Business 레지스트리들은 기본적으로 키워드 기반의 단순 검색을 제공하고 있는데 이것은 서비스 검색에 있어서 많은 한계점을 낳는다. 예를 들어, 새로운 쇼핑물을 구축하려는 개발자가 온라인 쇼핑서비스

에 대해 신용카드 결제 서비스를 제공하는 서비스를 검색하고자 할 때, 개발자는 먼저 쇼핑서비스에 대한 검색을 수행한 후 카드결제 서비스에 대한 검색과 기타 비즈니스 정보의 부합여부를 순차적으로 검색해야 할 것이다. 결국, 이러한 방법은 개발자로 하여금 서비스 검색에 많은 시간과 노력을 요구한다. 따라서, 보다 효율적인 서비스 검색환경을 제공하기 위해서는 서비스를 제공하는 에이전트도 필요하지만 그러한 서비스와 대상 에이전트를 지능적인 방법으로 탐색해 줄 수 있는 에이전트가 별도로 필요하다. 더불어, 그러한 에이전트를 등록, 저장, 관리할 수 있는 기반 시스템과 인프라의 구축이 필수적이다.

그림 6은 이러한 배경을 통해 도출한 에이전트 등록과 검색 서비스를 위한 시스템의 개념적 계층 구조이다. 이 계층 구조는 크게 4가지 관점(View)에 몇 개의 Layer가 포함되어 있는 구조이다. Agent Platform Layer는 FIPA Compatible Agent Platform을 의미하는 것으로 서로 다른 벤더에 의해 개발된 상이한 에이전트 시스템을 포함한다. Registry Repository System View에 포함된 3가지 계층은 에이전트의 등록, 저장과 관련된 서비스를 제공하는 역할을 한다. Agent Transport Layer는 XML, SOAP, ebXML 기반의 메시지 프로토콜을 사용해 에이전트 플랫폼과 레지스트리 사이의 메시지 교환 및 에이전트 전송을 담당한다. Agent Registry / Repository Service는 본 연구에서 가장 중요한 역할을 담당하고 있으며 각각 다음과 같은 책임을 갖는다.

- Agent Registry Service는 기본적으로 에이전트 소유자 정보, 기업 정보, 등록 에이전트 및 에이전트 시스템 정보 등 에이전트를 이용한 비즈니스 수행에 필

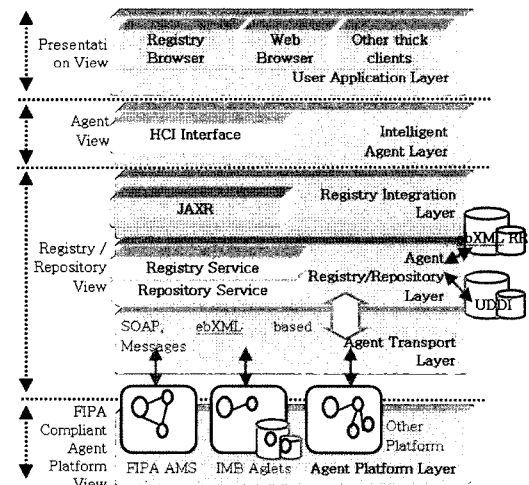


그림 6 개념적 시스템 계층구조

요한 메타 데이터를 유지/관리할 책임이 있다. 에이전트 수요자는 Registry Service를 통해 질의할 수 있고, 해당 에이전트가 어디에 위치해 있는지 알 필요는 없다.

- Repository Service는 공급자에 의해 제공된 에이전트를 영구 기억장치에 신뢰할 만한 방법으로 보관하고 관련 정보를 레지스트리에 등록할 책임이 있다. 더불어, 에이전트 검색(retrieval) 요청에 대한 응답을 한다. 이것은 일종의 코드서버 역할을 하며 모바일 에이전트 환경에서 에이전트 전송 속도를 높여 줄 수 있는 특성이 있다.

Registry와 Repository 그리고 Actor와의 transaction 관계도를 보이면 그림 7과 같다.

Agent Provider actor는 agent 생성 및 제공자를 의미하고 Agent Repository actor에게 Agent를 전송할 책임을 갖는 사용자 또는 특정 Agent System에 해당한다. Agent Consumer actor는 Agent Registry actor에게 관련 질의를 하는 사용자이거나 다른 에이전트가 될 수 있다. Agent Consumer actor는 Agent Registry actor를 통해 얻은 결과로부터 다른 에이전트 시스템에 존재하는 에이전트와 직접 연결을 시도하거나 필요하다면 하나 이상의 Repository를 통해 대상 Agent를 제공할 수 있다. 이와 같은 과정을 UML의 Sequence Diagram을 이용해 좀 더 구체적으로 표현한 것이 그림 8이다.

이 외에 Registry Integration Layer는 UDDI Registry, ebXML Registry 또는 다른 벤더에 의해 제공되는 Registry 등 서로 다른 이질적 구조를 가지는 레지스트리 정보를 JAXR 인터페이스를 이용하여 하나로 통합, 검색할 수 있는 기반을 제공해준다. Intelligent Agent Layer는 통합된 레지스트리 상에서 온톨로지를 분석하거나 등록 에이전트를 체계적으로 분류, 활용할 수 있도록 관련 정보에 대한 지식화를 담당한다. 또한, 이 계층은 HCI Interface를 이용하여 사용자 어플리케이션 계층과 연결되며 사용자의 다양한 요구를 받아들여 Registry Information Layer를 통해 질의하고 처리된 결과를 User Application Layer로 반환하는 역할을

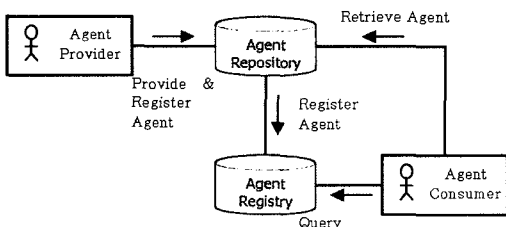


그림 7 Agent registration & retrieval diagram

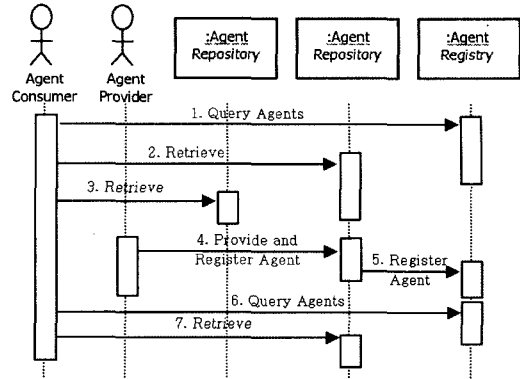


그림 8 Agent registration & retrieval sequence diagram

수행한다. User Application Layer는 Web browser와 같은 thin client 또는 Registry 전용 브라우저와 같은 thick client 형태의 사용자 어플리케이션을 의미한다.

4. 핵심 설계(Key Design)

여기서는 지금까지 소개한 내용을 바탕으로 설계된 시스템의 핵심 내용들을 기술한다. 그림 9는 제안 시스템의 핵심 클래스를 보인 것으로 ebXML의 RIM을 따른다. 그림 10은 ebXML RIM을 기반으로 본 시스템에서 확장한 class diagram이다.

그림을 통해 알 수 있듯이 AgentEntry와 AgentFolder는 각각 ebRIM의 ExtrinsicObject와 RegistryPackage를 확장하였다. AgentEntry는 AID(Agent ID), Agent Registration Time, Agent Communication을 위한 language등 Agent의 등록과 검색을 위해 가장 일반적이면서도 공통적인 속성들로 이루어져 있다. 또한, AgentFolder는 등록된 Agent를 Repository의 특정 폴더에 안전한 방법으로 저장, 보관하기 위한 목적으로 folder의 unique ID, 소유자의 ID, 마지막 수정 일자(last update time)등의 관련속성으로 이루어져 있다. 표 1은 AgentEntry의 주요 속성을 보여준다. AID는 에이전트 이름과 에이전트 플랫폼 주소의 쌍을 '@'로 연결한 형태의ID이며 Agent Unique Id는 Registry가 관리를 위한 목적으로 Agent에게 부여하는 고유 식별자이다. Agent Execution Side Type은 해당 에이전트가 Client-side 즉, Client application 내에서 수행되는지, 아니면 Server-side에서 수행되는 agent인지를 구분한다. Agent Collaboration Type은 등록 에이전트가 다른 에이전트와 협력을 통해 동작하는지(collaborative type) 아니면 혼자서 독립적으로 실행되면서 서비스를 제공하는지(stand-alone type)에 대한 여부를 저장한다.

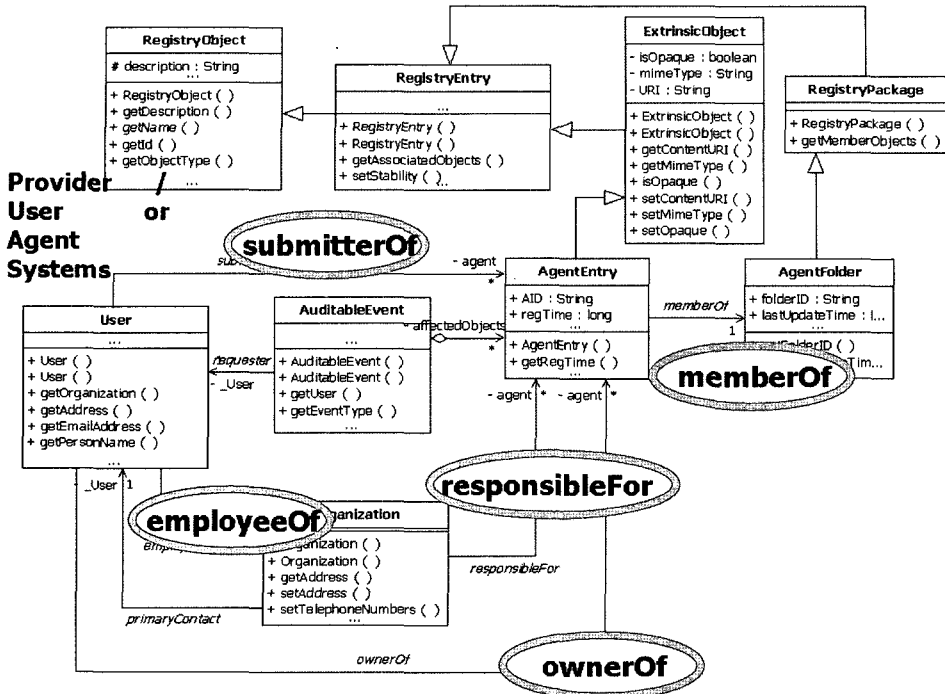


그림 11 레지스트리 객체와 Agent 사이의 상호연관관계

5. 구현 결과(Implementation)

3장과 4장에서 설계된 내용을 바탕으로 하여 구현을 위해 작성된 소프트웨어 시스템 구성도는 그림 12와 같다.

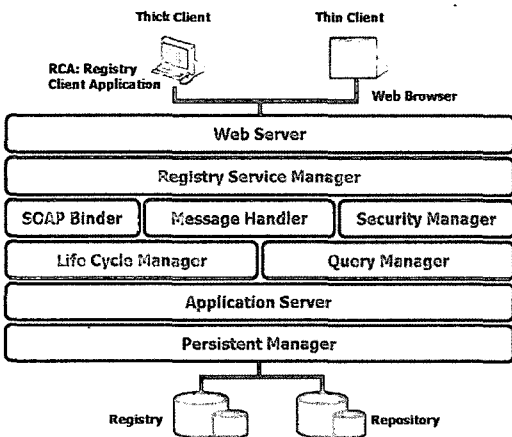


그림 12 에이전트 등록/저장소 소프트웨어 시스템 구성도

그림 12를 통해 알 수 있듯이 실용적이면서 편리하고 친숙한 사용자 접속 환경을 제공하기 위해 웹 기반의 인터페이스 환경을 기본적으로 제공한다. 또한, 웹의 비연결 지향적 성격을 일부 보완하고 보다 안정적

인 서비스 지원을 위해 RCA(Registry Client Application)를 개발하였다. RCA는 Registry Browser에 해당하는 클라이언트 어플리케이션으로 Agent의 등록과 검색, 관련 작업의 모니터링을 돕는다. 특히, 내부 메시지는 웹을 통한 HTTP 80번 포트를 사용함으로써 기업이나 병원의 방화벽 문제를 해결할 수 있고 SOAP의 사용으로 보다 유연한 상호운용성을 보장할 수 있게 했다.

시스템의 전체적 구성은 MVC패턴[17]을 적용한 3계층(3-Tier) 기반구조 설계를 J2EE 환경[18]에서 구현하였다. 그 결과로 프리젠테이션과 업무로직을 분리시키고, 트랜잭션 관리를 통한 데이터 무결성을 보장할 수 있도록 했으며 분산 컴포넌트의 배포 및 재사용성을 극대화 시킬 수 있도록 설계되었다.

Registry Service Manager는 Life Cycle Manager, Query Manager등의 다른 구성 컴포넌트와의 연결을 담당하고 유입되는 질의를 처리한다. SOAP메시지를 포함한 기타 메시지 핸들링을 위해 SOAP Binder와 Message Handler가 존재하고 PKI 기반의 XML 보안을 위해 Security Manager가 그 역할을 담당한다. 그림 13은 Agent Profile 문서 보안을 위한 암호화 및 복호화 개요를 보인 것으로 현재 지속적인 연구를 진행하고 있는 상황이다. Agent Profile 문서는 에이전트의 등

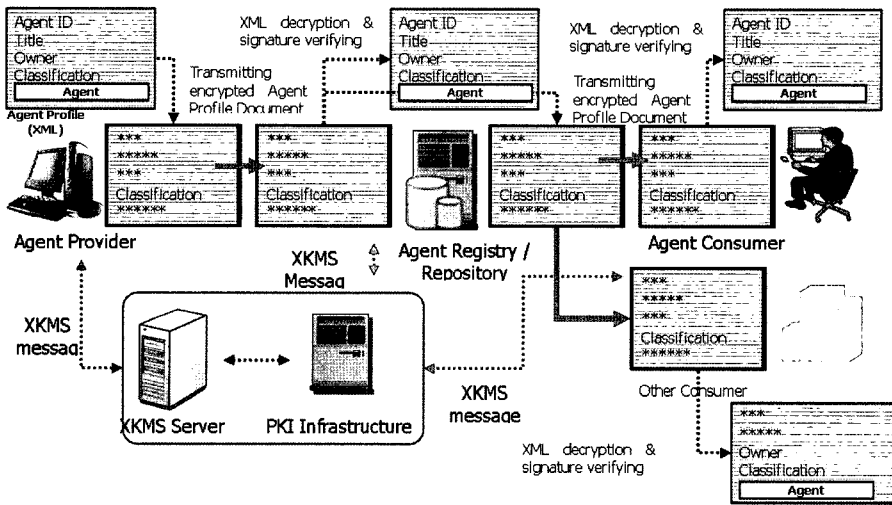


그림 13 Agent profile document에 의한 에이전트 등록 개요

록과 전송 시 에이전트 소유자의 기본 정보를 비롯하여 Agent와 관련된 다수의 메타 정보를 포함하는 XML 문서이다. 이 Agent Profile 문서 내에는 등록 agent의 바이너리 코드가 포함되어 전송되거나 Agent source의 위치정보가 URL 형태로 기록되어 전송된다. 때문에, Agent Profile 문서는 XML Encryption[19] 기술을 이용해 원문 전체 혹은 임의의 데이터(Agent code가 포함된 경우 agent code만 암호화 할 수도 있다)에 대한 암호화가 이루어지고, XML Digital Signature[20] 기술을 이용해 서명된 Agent Profile 문서가 전송된다. 이때, Security Manager는 XKMS(XML Key Management System)[21] 서버를 통해 정보를 추출하거나 인증서 검증을 요청하여 암호화된 Agent Profile 문서에 대한 복호화를 담당한다. 기타 Persistent Manager는 Repository Service와 관련되어 있으며 Agent의 안전한 저장과 영속적 서비스 제공을 위한 목적으로 존재한다. LifeCycle Manager는 Registry 내부 객체들의 생명주기(Submitted-Approved-Deprecated-Removed)를 관리하며 QueryManager는 클라이언트의 요청에 따라 대상Agent의 빠른 검색을 지원한다.

지금까지의 내용을 바탕으로 시스템을 구현한 결과 화면은 아래와 같다.

그림 14와 그림 15는 웹 기반의 Agent 등록과 검색 화면이다. 등록하고자 하는 Agent의 분류 체계를 선택하고 부가적인 정보를 입력한 다음 전송버튼을 누르는 간단한 동작으로 Agent가 시스템으로 전송되고 레지스트리에 그 정보가 기록 된다. 등록된 Agent는 간단한 키워드 검색이나 Classification Scheme을 활용한 Drill Down 검색을 통해 검색이 가능하다. 그림 16은 이렇게

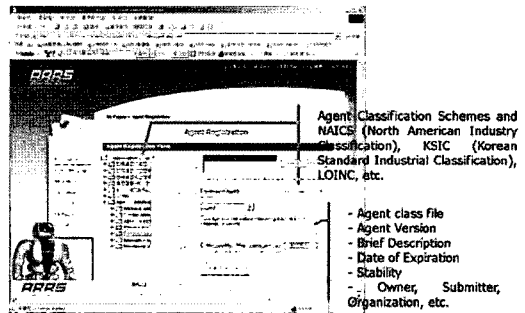


그림 14 웹 기반 에이전트 등록화면

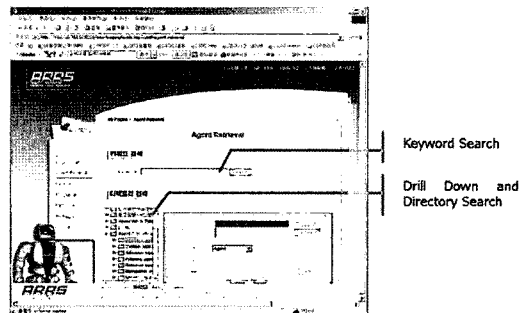


그림 15 웹 기반 에이전트 검색화면

등록된 Agent 목록을 웹을 통해 조회하는 화면이며 그림 17은 전용 클라이언트 도구로 개발된 RCA(Registry Client Application)에 의해 검색된 결과를 보여주고 있다.

이상과 같이 Agent와 Agent를 제공하는 공급자, 그리고 Agent를 이용하는 소비자 사이의 상호 연관관계를 잘 도출하고 전송된 Agent를 ebXML에 기반한

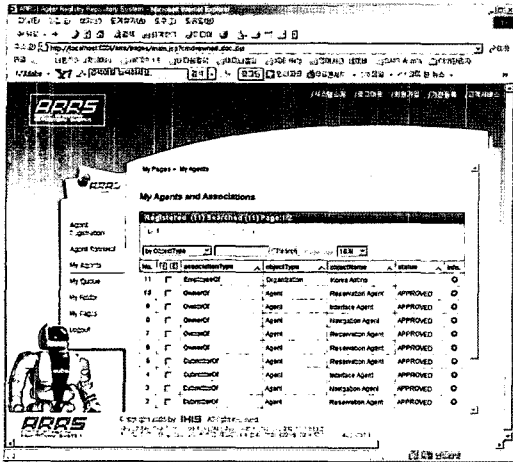


그림 16 질의 결과에 의한 에이전트 목록화면

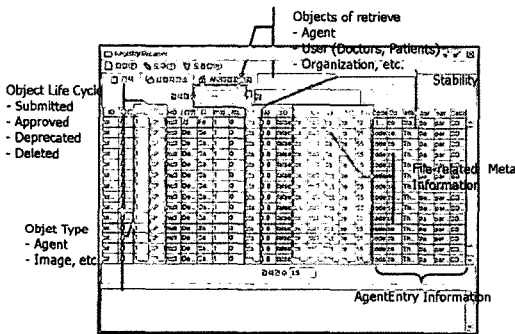


그림 17 에이전트 등록과 검색을 위한 Registry Client Application

RIM 모델과 분류체계에 따라 등록, 관리하게 함으로써 보다 글로벌 한 네트워크 환경에서의 agent 등록과 검색이 용이함을 보였다.

6. 결론 및 향후 연구방향(Conclusion and Future Works)

지금까지 Agent의 등록과 저장, 검색 서비스를 제공하기 위한 하나의 방법으로 ebXML기반의 Agent 등록 및 저장소의 설계 와 그 구현내용을 소개했다.

다음에 요약한 것은 앞으로 본 시스템에 적용할 만한 구현관련 이슈와 향후 연구과제이다.

- Autonomous Agent Registration and Retrieval: 인간의 개입 없이 Agent Profile에 의한 자동화된 에이전트 등록과 검색 방법론
- Ontology based Agent Brokering: 온톨로지 기반 Agent 중재(brokering or matchmaking)와 그에 따른 지식화된 서비스 제공

- Registry Federation: 상호 이질적 정보모델을 갖는 Registry의 간의 통합-운용 모델
- Agent Transport as a Code Server: Mobile Agent 환경에서 Agent Class의 전송과 동적 적재를 위한 코드 서버로써 활용하여 에이전트 이동에 드는 부하를 최소화
- Agent based SOA environment: Agent의 기반의 SOA(Service Oriented Architecture) 환경 구축
현재로서는 제안한 Agent Registry / Repository System에 대해 prototype 수준의 구현이 이루어져 있으며 User에 의한 에이전트 등록과 검색에 초점이 맞추어져 있다. 하지만, 본 연구를 기반으로 향후에는 상기에 나열한 과제를 중심으로 제안한 시스템의 현실적 적용에 있어서 발생할 수 있는 문제점을 분석해보고 기 개발된 다양한 에이전트 플랫폼과 연동하여 그 결과를 측정해 보고자 한다. 더불어, 현재의 단순 검색을 넘어 Filter Query를 포함하는 ad-hoc 질의 서비스가 가능하도록 보다 최적화된 Agent Query Model을 별도로 설계, 개발할 예정이며 본 시스템을 mobile agent 환경에 적용할 경우 agent 코드 서버로서의 성능 테스트와 agent 이동에 따른 agent tracking 서비스를 지원할 수 있는 방안도 연구할 계획이다.

참고 문헌

- [1] Agent Technology Roadmap: Overview and Consultation Report, Agent based computing. AgentLink, 2004.
- [2] Danny B. Lange and Mitsuru Oshima, Programming and Deploying Java™ Mobile Agents with Aglets. Addison-Wesley Professional, 1998.
- [3] Milen Tilev, "A Decisive Agent Based Exchange Platform for Tracy Mobile Agent Systems," Tracy Technical Report No 7, January 2005.
- [4] Arkady Zaslavsky, Mobile Agents-Can They Assit with Context Awareness? Proceeding of IEEE international conference on mobile Data management, 2004.
- [5] Paolo Bellavista, Dario Bottazzi, Antonio Corradi, Rebecca Montanari and Silvia Vecchi, Mobile Agent Middleware for Context-aware Applications. Handbook of Mobile Computing, 2004.
- [6] FIPA Abstract Architecture Specification, FIPA TC Agent Management, <http://www.fipa.org>. 2002.
- [7] FIPA Agent Management Specification, FIPA TC Architecture, <http://www.fipa.org>. 2002.
- [8] ebXML Specification, Available at: <http://www.ebxml.org/specs/index.htm>. Accessed April 2, 2005.
- [9] ebXML RIM V2.0, Available at: http://www.oasis-open.org/committees/regrep/documents/2.0/specs/eb_rim.pdf. Accessed April 2, 2005.

[10] ebXML RS V2.0, Available at: <http://www.oasis-open.org/committees/repreg/documents/2.0/specs/ebrs.pdf>. Accessed April 2, 2005.

[11] OASIS Standards and Other Approved Work, Available at: <http://www.oasis-open.org>. Accessed March, 2004.

[12] UDDI Specification, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec. Accessed April 2, 2005.

[13] Web Services Architecture, W3C Working Group Note 11 February 2004, Available at: <http://www.w3.org/TR/ws-arch>. Accessed March, 2004.

[14] W3C Web Services Activity, Available at: <http://www.w3.org/2002/ws>. Accessed March, 2004.

[15] Service Oriented Architecture and Web Service, IBM, Available at: <http://www-306.ibm.com/software/solutions/webservices/documentation.html>. Accessed March, 2004.

[16] SUN, Java API for XML Registries, <http://java.sun.com/webservices/jaxr/index.jsp>

[17] Model-View-Controller, Available at: <http://java.sun.com/blueprints/patterns/MVC-detailed.html>. Accessed March. 2004.

[18] J2EE V1.4 Documentation, Available at: <http://java.sun.com/j2ee/1.4/docs/index.html>. April 2, 2005.

[19] XML Encryption Syntax and Processing W3C Recommendation 10 December 2002, Available at: <http://www.w3.org/TR/xmlenc-core>. Accessed April 2, 2005.

[20] XML-Signature Syntax and Processing W3C Recommendation 12 February 2002, Available at: <http://www.w3.org/TR/xmlsig-core>. Accessed April 2, 2005.

[21] XML Key Management Specification (XKMS) Version 2.0, W3C Editor's Draft 30th March 2005. Available at: <http://www.w3.org/2001/XKMS/>. Accessed April 5, 2005.

프트 개발PM. 관심분야는 패턴인식, 기계학습, 데이터마이닝, 의료정보학



김 일 곤

1980년 서울대학교 수학교육과 졸업
 1988년 서울대학교 전산과학과 석사학위 취득.
 1991년 서울대학교 전산과학과 박사학위 취득.
 1992년 3월~현재 경북대학교 전기전자컴퓨터학부 교수, 의료정보학과 교수.
 2003년 6월~현재 지능형진료지원 및 정보공유시스템개발연구소장.
 2005년 1월~현재 한국 ISO/TC 215 WG2 대표자.
 2005년 9월~현재 ISO/TC 215 WG2 Document Registry Framework Project Leader.
 2005년 12월~현재 EHR 공동핵심기술 연구사업단 제3세부 과제책임자.
 관심분야는 의료정보학, 에이전트, 서비스 그리드



김 일 광

1997년 계명대학교 컴퓨터공학과 졸업
 1996년~2001년 이지콤 정보기술 대표
 2000년 계명대학교 컴퓨터공학과 석사학위 취득.
 2005년 경북대학교 컴퓨터공학과 박사과정수료.
 2005년~현재 지능형진료지원 및 정보공유시스템개발연구소 근무
 관심분야는 이통에이전트시스템, 분산시스템, 그리드 컴퓨팅, 의료정보학

이 재 영

1998년 경북대학교 컴퓨터공학과 졸업
 2000년 경북대학교 컴퓨터공학과 석사학위 취득.
 2005년 경북대학교 컴퓨터공학과 박사과정수료.
 2005년~현재 (주)유케어스