

다중 에이전트 시스템 환경에서 분산된 온톨로지 저장소의 사용*

김성태

가톨릭 대학교 컴퓨터.정보공학부
(ariarax@catholic.ac.kr)

지경환

가톨릭 대학교 컴퓨터.정보공학부
(sshine106@catholic.ac.kr)

양정진

가톨릭 대학교 컴퓨터.정보공학부
(jungjin@catholic.ac.kr)

급속한 정보통신의 발달은 인간의 삶에 직접적인 영향을 끼치는 단계에 접어들게 되었다. 기존의 컴퓨팅은 사용자의 요청에 근거하여 동작하지만 유비쿼터스 컴퓨팅은 사용자의 의향과 상황을 인지해 능동적으로 목표를 설정하고 협력을 통하여 업무를 수행하게 된다. 앞으로 다가올 새로운 컴퓨팅 환경에서 주된 역할을 하게 될 개체인 에이전트와 이들이 상호 협력하는 기반구조인 다중 에이전트 시스템에 관하여 활발히 연구가 진행 중이다. 본고에선 다중 에이전트 시스템에서의 의미적 상호 운용성을 보장하고 지적활동의 중심적 자료구조가 될 온톨로지와 온톨로지를 효율적으로 관리하는 저장소에 관한 요구사항과 구조를 제시한다. 또한 분산적으로 관리 되는 온톨로지 저장소와 다중 에이전트 환경에서 일반적인 온톨로지 저장소 사용을 위한 온톨로지 에이전트를 제시 한다.

논문접수일 : 2005년 5월

게재확정일 : 2005년 12월

교신저자 : 양정진

1. 서론

급속한 정보통신의 발달은 인간의 삶에 직접적인 영향을 끼치는 단계에 접어들게 되었다. 기존의 컴퓨팅은 사용자의 요청에 근거하여 동작 하지만 유비쿼터스(Ubiquitous) 시대의 컴퓨팅은 사용자의 의향과 상황을 인지해 동적으로 목표를 설정하고 자동으로 협력해 목표를 해결하게 된다. 앞으로 곧 다가올 새로운 컴퓨팅에서 주된 역할을 하게 될 개체인 에이전트(Agent)와, 이들이 상호 협력하는 기반구조인 다중 에이전트 시스템에 관한 연구가 활발히 진행 중이다. 본고에선 다중 에이전트 시스템에서의 의미적 상호 운용성을 보장하고 지

능적인 활동의 중심적 자료구조가 될 온톨로지(Ontology)와 온톨로지를 효율적으로 관리하는 저장소에 관한 요구사항과 구조를 제시한다. 또한 분산적으로 관리 되는 온톨로지 저장소와 다중 에이전트 환경에서의 일반적인 온톨로지 저장소 사용을 위한 온톨로지 에이전트를 제시 한다. 온톨로지 에이전트는 다양한 온톨로지 저장소에 웹서비스를 통해 접근함으로써 표준적인 상호작용을 수행한다.

다음 장에서 기존의 연구 동향과 문제점을 분석하고 3장에서 다중 에이전트 시스템에서 분산된 온톨로지 저장소를 사용하기 위한 요구 사항을 설명한다. 4장에서 요구사항이 반영된 전체 적인 시

* 본 연구는 21세기 프론티어 연구개발 사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크원천 기반 기술개발사업과 2005년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

스텝 구조를 제시하고 5장에서 제시된 구조에 따른 토의를 하고 마지막으로 결론을 맺는다.

2. 관련 연구

유비쿼터스 환경에서의 어플리케이션들은 자율적이고 지능적으로 업무를 수행하고 목적에 부합되는 상호규약(Protocol)을 이용해 상호작용한다. 에이전트는 자율적이고 지능적인 특성을 가지는 어플리케이션이며[1] Foundation for Intelligent Physical Agents (FIPA)에서 에이전트의 물리적인 구성과 상호작용을 위한 표준을 제시하였다.[2]

온톨로지는 유비쿼터스 환경에서 어플리케이션 간의 의미적 상호 운용성을 보장하는 메타데이터 레지스트리역할을 수행하며 지식의 재 표현과 확장을 위한 일반적인 방법론을 제공한다.

2.1 의미 중재를 위한 온톨로지

온톨로지(Ontology)는 철학의 한 갈래로 존재의 본질을 연구하는 형이상학이다. 메타데이터 레지스트리로서 온톨로지는 개체와 각 개체간의 관계성을 기술하는 언어로서 에이전트의 인지능력을 위한 기본 구조가 될 수 있다.

온톨로지는 주어, 술어, 목적어를 기본요소로 트리플(Triple)구문 형태로 자원을 기술하는 기반구조인 Resource Description Framework (RDF)[3]를 바탕으로 서술 논리(Description Logic)를 기술하기 위해 DARPA Agent Markup Language + Ontology Inference Layer (DAML+OIL)[4]로 기술한다. 이 모든 언어는 eXtensible Markup Language (XML)[5]을 기반으로 구법을 가지게 되며 기계가 읽을 수 있는 특성을 가지게 된다.

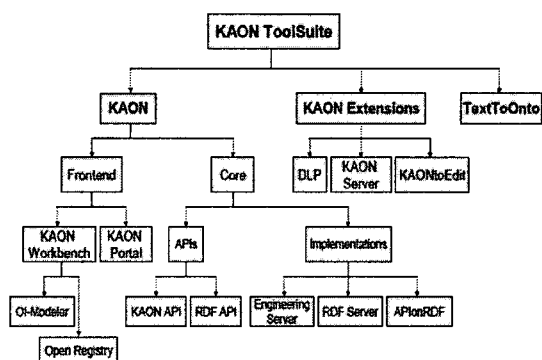
World Wide Web Consortium (W3C)는 앞서 소개한 언어들의 표준을 관리하며 DAML+OIL의 확장 버전인 Web Ontology Language (OWL)[6]을 제시하였다.

2.2 기존의 온톨로지 저장소

온톨로지는 트리플 형태로 파싱되며 서술 로직이 표현한 지식을 모두 추출하기 위하여 추론과정을 거쳐야 한다. 대용량의 온톨로지를 파싱하고 추론, 회수, 저장하기 위해서는 온톨로지 저장소가 필요하다. 대표적인 온톨로지 저장소로서 HAWK[7], Karlsruhe Ontology Management Infrastructure (KAON)[8]이 연구 되었다.

HAWK는 OWL로 표현된 지식을 위한 기반 시스템으로 관계형 데이터베이스 관리 시스템과 Java DataBase Connectivity (JDBC)를 사용하여 지식의 추론과 회수, 저장, 수정, 관리를 가능하게 하는 시스템이다. HAWK는 온톨로지의 파싱, 수정, 조작, 보존하는 구현과 저장소를 이용할 수 있는 Application Programming Interface (API)를 제공한다. HAWK는 구현위에 일반적인 인터페이스를 두는 설계 철학을 가졌다. 그렇기 때문에 높은 재사용을 가진다. 예를 들어, 온톨로지는 메모리 상주(In-memory) 저장 모델로서 직렬화(Serialization) 되어 편집 될 수 있고, 문서에 저장 될 수도 있다. 또한 지식 베이스 시스템으로서 질의 처리를 위해 사용될 수도 있다.

KAON 역시 온톨로지 기반의 지식 베이스이며 [그림 1]에서 볼 수 있듯이 HAWK에 비해 어플리케이션을 개발하기 위한 다양한 컴포넌트를 제공하는 기반구조를 가진다. KAON 엔지니어링 서버는 관계형 데이터베이스를 바탕으로 한 온톨로지 저장 메커니즘, 온톨로지 엔지니어링을 위한 저장



[그림 1] KAON Tool Overview

메커니즘, 응집성 탐지를 통한 클라이언트 사이드 캐시, 온톨로지 요소의 벌크 로딩과 분산 변화 탐지 메커니즘 등을 가지고 있고, 온톨로지를 핸들하기 위한 기반 구조가 될 수 있는 KAON API들을 기술 하고 있다. 저장 메커니즘은 RDF 모델과 관계형 데이터베이스 모델을 포함하고 KAON 질의 언어를 통하여 KAON 온톨로지에 대한 질의를 가능하게 하는 역할을 한다.

이 외에도 KAON은 어플리케이션 서버와 KAON API 기반 구조위에서 동작하는 OntoEdit를 위한 KAONtoEdit 플러그인과 같은 온톨로지 관리를 위한 모듈들을 제공한다.

앞 서 소개한 온톨로지 저장소들은 OWL로 재 표현된 지식을 저장, 관리하고 회수하기 위한 기반 구조를 제공한다. 하지만 온톨로지 저장소들이 구현된 방법과 지식회수를 위한 방법들이 다르며 지식의 질의를 위해 RDF Data Query Language (RDQL)[9], OWL Query Language (OWL-QL) [10], SPARQL Protocol And RDF Query Language (SPARQL)[11]등과 같은 다양한 질의 언어들을 지원한다. 에이전트들은 온톨로지 저장소에 접근하기 위하여 저장소가 지원하는 질의 언어정보와 상호작용을 위한 프로토콜 정보를 포함

(Ad-hoc)하여야 하고 분산된 온톨로지 저장소에 접근하기 위해서는 각 온톨로지 저장소에서 지원하는 모든 질의 언어정보와 프로토콜을 이해할 수 있어야 하며 이것은 대형 에이전트 시스템과 온톨로지 저장소 개발함에 있어 생산성을 저해 시킨다.

2.4 웹 서비스

웹 서비스[12]는 인터넷을 기반으로 한 오픈 네트워크를 통해 비즈니스 간의 컴퓨터 프로그램을 연합시키기 위한 표준화된 기술을 일컫는다. 웹 서비스는 데이터 표준에 대한 정의를 규정하여 데이터 교환의 유연성을 제공하기 위한 것으로 프로그램 언어에 독립적이고 이질적인 시스템이나 프로그램 간의 연결 역할을 수행하여 상호 운용성을 높인다. 클라이언트와 서버는 각기 다른 환경에서 구축될 수 있으며 웹 서비스를 구현하기 위해 기존 코드를 변경하지 않아도 된다. 그러므로 벤더에 독립적이고 시스템 구조의 유연성, 사용의 용이성, 통합 환경을 제공하고 자동화된 소프트웨어 통합을 추구하여 상호 운용성을 증대시킨다.

웹 서비스는 HTTP 프로토콜을 이용하여 이질적인 시스템간의 상호 운용성을 보장한다. 본고에서는 웹서비스가 제공하는 통신 메커니즘을 이용하여 에이전트 시스템과 온톨로지 저장소간 통신 프로토콜을 단일화 한다.

3. 시스템 요구사항

이번 장에서는 기존 연구에서 질의 언어와 통신 프로토콜의 다양함을 제거하기 위한 요구 사항을 제시한다. 이를 위해, 에이전트 시스템과 온톨로지 저장소간 프로토콜을 단일화 할 수 있어야 하며

에이전트 시스템내부 메시지를 변환할 수 있어야 한다. 또한 질의문 사이에 상호변환을 수행할 수 있어야 한다.

3.1 온톨로지 저장소의 요구사항

온톨로지를 기술하는 표준 언어는 OWL이며 대부분의 온톨로지 저장소는 OWL/RDF로 기술된 온톨로지를 파싱하여 관계형 데이터베이스에 저장한다. 파싱된 결과를 데이터베이스에 저장하기 위하여 각기 다른 메커니즘을 사용하고 있으며 저장된 데이터를 회수, 검색하기 위한 다양한 API를 제공한다. 하지만 에이전트 차원에서 온톨로지 저장소를 이용할 수 있는 일반적인 방법론은 없다. 온톨로지 저장소는 앞서 제시한 기능들을 목적에 적합한 통신을 제공하는 부분과 저장소의 메커니즘을 구현한 부분으로 분리할 필요가 있다.

FIPA에서는 멀티 에이전트 시스템에서 지식베이스 이용을 위한 Open Knowledge Base Connectivity (OKBC)[13] 프로토콜과 온톨로지 서비스 참조 모델을 제시하였지만 KAON, HAWK, 혹은 다른 온톨로지 저장소를 이용하는 일반적인 방법론은 아니다. 온톨로지 저장소는 데이터를 다루어 줄 수 있는 메커니즘이 될 수 있는 기반 구조로, API를 가지며 애드-혹(ad-hoc)되거나 라이브러리의 형태로 포함되는 인터페이스를 제공함으로써 정책화되는 구조로 연구가 진행될 필요가 있다. 온톨로지 저장소가 제공하는 정책 중 하나로서 웹서비스를 이용한 질의 처리를 제공한다면 에이전트 시스템을 비롯하여 다양한 어플리케이션 간 상호운용성을 보장할 수 있다.

3.2 다중 에이전트 시스템의 요구사항

멀티 에이전트 시스템에서 구현된 에이전트들

은 Agent Communication Language (ACL)[2] 메시지를 이용하여 통신을 하게 된다. 에이전트가 온톨로지 저장소를 사용하려고 할 때 사용하는 방법론이 온톨로지 저장소에 의존적이면 에이전트를 설계하기 전에 온톨로지 저장소에서 제공하는 API등을 알고 있어야 하고 구현된 에이전트는 다른 온톨로지 저장소를 사용하기 어려워질 것이다. 에이전트의 입장에서 온톨로지 저장소를 사용하기 위한 일반적인 방법론의 필요성은 자명하며 그 대안으로 온톨로지 저장소를 에이전트화 하여 온톨로지를 사용하는 방법을 에이전트간의 ACL 메시지를 통한 상호작용으로 일반화시킬 수 있다.

온톨로지 저장소를 사용하는 에이전트는 온톨로지 저장소에 질의를 할 수 있어야 하며 질의에 대한 결과를 회수할 수 있어야 한다. 또한, 에이전트가 수행한 결과를 온톨로지 저장소에 저장할 수 있어야 하며 이 경우엔 트리플 형태로 인스턴스화되어 저장되어야 한다. 추론 역할을 수행할 에이전트의 경우는 서술 논리 수준의 질의를 할 수 있어야 하며 TBox, ABox [14] 추론을 수용할 수 있어야 한다.

현재 시점에서 온톨로지 저장소의 개념들 간 관계성을 수정, 학습하는 능력을 에이전트에게 부여하기에는 기술적으로 부족한 부분이 많으며 차후에는 새로운 요구 사항이 될 수 있을 것이다.

3.2.1 온톨로지 대리인 에이전트

멀티 에이전트 환경에서 에이전트들은 필요한 정보를 얻기 위해 온톨로지 저장소에 질의를 하고, 온톨로지 저장소는 에이전트의 질의를 받아 처리한 후 결과를 에이전트에 전달하게 된다. 이때, 개별 에이전트가 분산 환경에서 이기종의 온톨로지 저장소에 질의를 하기 위해서는 필요한 온

톨로지가 존재하는 각각의 온톨로지 저장소가 받아들일 수 있는 서로 다른 질의문을 생성하여야 한다. 본고에서는 멀티 에이전트 시스템 환경에서 에이전트와 온톨로지 저장소 사이에 대리인 역할을 하는 온톨로지 에이전트(Ontology Agent)를 사용한다. 온톨로지 에이전트는 에이전트로부터 질의문을 받아 온톨로지 저장소에 질의문을 전달하고, 온톨로지 저장소로부터 결과를 받아서 질의를 요청한 에이전트에 전달하는 역할을 수행하게 된다. 온톨로지 에이전트를 위한 요구사항으로는 다음과 같다.

첫째, 온톨로지 에이전트는 각각의 에이전트들로부터의 질의요청을 ACL 메시지 형태로 전달 받는다. 이는 FIPA-Compliant 에이전트의 표준으로 사용되는 ACL 메시지를 사용함으로써 에이전트가 범용적인 에이전트들 간 통신 방법을 사용할 수 있게 한다.

둘째, 내부에 ACL 메시지를 온톨로지 질의 언어로 변환하는 파서를 가지고 있어야 한다. 본고에선 온톨로지 질의 언어의 예로 OWL-QL을 사용했기 때문에 제시할 온톨로지 에이전트는 ACL 메시지를 OWL-QL 형태로 변환하고 OWL-QL 메시지를 ACL 메시지로 변환하는 파서를 가지고 있어야 한다. 파서는 에이전트로부터 받은 ACL 메시지를 OWL-QL 형태로 변환하고, 온톨로지 에이전트는 이를 온톨로지 저장소에 전달한다. 또한 온톨로지 저장소로부터 받은 OWL-QL 형태의 결과를 ACL 메시지 형태로 변환하여 다른 에이전트에게 전달한다.

셋째, ACL 메시지를 통해 만들어진 OWL-QL 형태의 질의문을 웹 서비스에서 사용하는 Simple Object Access Protocol (SOAP)[12] 메시지 프로토콜에 담기 위해 SOAP BODY 요소에 포함시킨다. 온톨로지 에이전트는 SOAP 프로토콜을 이용

하여 이기종의 온톨로지 저장소와 메시지를 주고 받을 수 있어야 한다.

넷째, OWL-QL 질의문을 JESS language 형태로 변환시켜주는 파서가 내부에 존재해야 한다. 이는 온톨로지 저장소에서 추론을 위한 확장된 능력을 가지게 하는 기반이 된다. 본고에서는 로직 룰의 예로 JESS language를 사용한다.

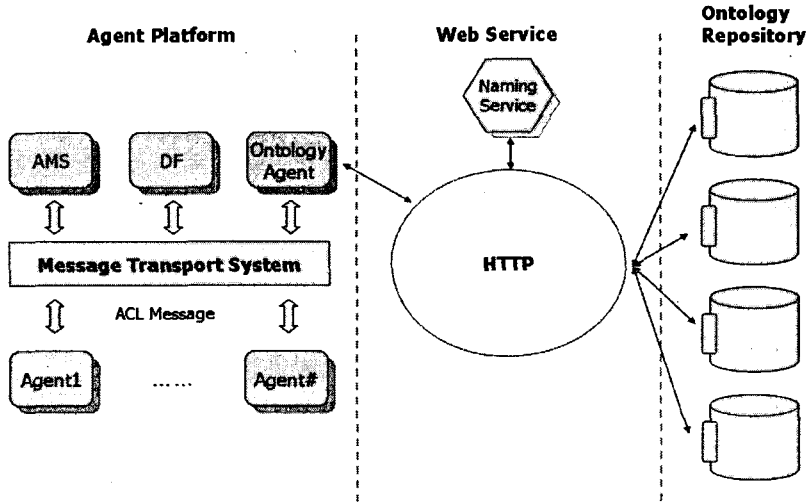
다섯째, 온톨로지 에이전트는 분산 환경에서의 온톨로지 저장소와의 통신을 위해 표준 웹 서비스를 사용함으로써 표준화의 이점을 가지며, 시스템 확장을 용이하게 하여야 한다.

3.3 질의문 변환 서비스

분산 환경에서 온톨로지 저장소의 종류에 따라 온톨로지 질의문의 종류도 바뀌게 된다. 온톨로지 에이전트는 온톨로지 저장소와의 통신을 위해 ACL 메시지를 OWL-QL 질의문으로 변환시키게 된다. 하지만 온톨로지 저장소가 요구하는 모든 종류의 질의문을 생성할 수는 없다. 그러므로 OWL-QL을 각 온톨로지 저장소에 맞는 질의문으로 변환 시켜주는 서비스를 웹 서비스를 통해 제공해야 한다. 온톨로지 에이전트는 OWL-QL을 변환 서비스를 통해 변환 후 온톨로지 저장소에 질의를 보내게 된다.

4. 시스템 구조

본고에서 제안하는 전체 시스템 구조도는 [그림 2]와 같다. 에이전트 플랫폼 부분, 웹서비스 부분, 온톨로지 저장소 부분으로 나누어진다. 에이전트 플랫폼의 각 에이전트들은 온톨로지 검색을 위해 온톨로지 에이전트에게 메시지를 보내게 된다. 온



[그림 2] 전체 시스템 구성도

톨로지 에이전트는 에이전트 플랫폼 안에 존재하며 에이전트와 온톨로지 저장소 사이에서 웹 서비스를 통해 분산되어 있는 온톨로지 저장소와 통신을 하게 된다. 온톨로지 에이전트는 다른 에이전트로부터 받은 메시지를 해석하여 온톨로지 질의 언어인 OWL-QL로 변환한다. OWL-QL로 변환되는 질의문은 일반적으로 모호성이 없는 명확한 언어이다. 변환된 OWL-QL은 웹 서비스를 통해서 온톨로지 저장소로 전달된다. 온톨로지 저장소는 전달되어진 OWL-QL을 질의 가능한 언어로 변환하는 작업을 한 후, 검색을 수행한다. 검색의 결과는 웹 서비스 호출자에게 전달되어지고, 웹 서비스 호출자인 온톨로지 에이전트는 결과값을 Agent 통신 언어인 ACL로 다시 변환하는 작업을 거쳐서 질의 요청을 한 에이전트에게 전달하게 된다.

4.1 다중 에이전트 시스템

본고에서 에이전트 플랫폼은 FIPA 표준을 따르는 에이전트 플랫폼을 기본으로 하고 있다. FIPA

표준의 에이전트 플랫폼에는 에이전트 간 통신을 위한 채널인 Agent Communication Channel (ACC)과 플랫폼 간 통신을 위한 채널인 Message Transport System (MTS)가 존재한다. 에이전트들은 이 채널로 메시지를 전달하고 각각의 에이전트들은 채널을 통해 자신의 메시지를 전달받게 된다. 또한 에이전트 플랫폼은 접속과 사용을 관리하고 에이전트 상태를 유지하고 생명주기를 관리하는 Agent Management System (AMS)과 옐로우 페이지 기능을 수행하는 Directory Facilitator (DF)를 가지고 있다.[2]

본 시스템에서 기존의 에이전트 플랫폼을 확장하여 온톨로지 저장소로의 질의를 담당하는 온톨로지 에이전트를 두어 다른 에이전트로부터 전달 받은 ACL 메시지를 변환하여 OWL-QL로 만들어 이를 온톨로지 저장소에 전달하게 한다.

4.1.1 온톨로지 에이전트

온톨로지 에이전트는 에이전트로부터 메시지를

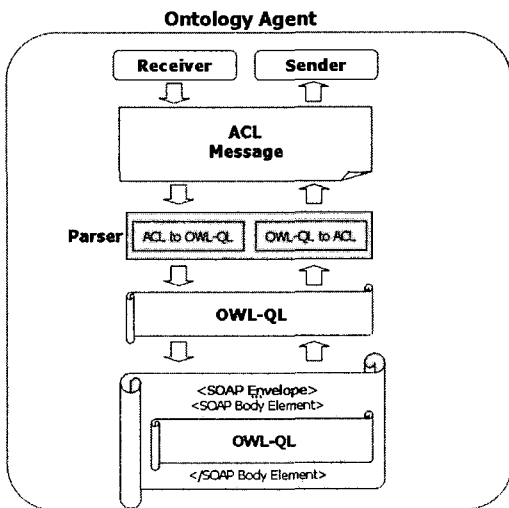
받아서 온톨로지 저장소에 질의어를 보내고, 온톨로지 저장소로부터 질의 결과를 받아서 질의를 요청한 에이전트에게 결과를 보내주는 역할을 한다. 이를 위해 온톨로지 에이전트는 앞서 서술한 요구 사항을 만족하여야 한다.

[그림 3]은 온톨로지 에이전트의 구성도이다. 온톨로지 에이전트는 ACL 메시지를 주고받는 송신자와 수신자정보를 각각 가지고 있고, 파서 부분에는 ACL 메시지를 OWL-QL로 변환시키는 ACL to OWL-QL 부분과 OWL-QL 메시지를 ACL 메시지로 변환하는 OWL-QL to ACL 부분을 포함한다. 또한 변환된 OWL-QL을 SOAP 메시지로 만드는 역할을 수행하는 부분을 포함한다. 온톨로지 에이전트는 수신자정보를 이용하여 ACC로부터 ACL 메시지를 받아온다. ACL 메시지의 내용은 파서를 통해 OWL-QL로 변환된다. OWL-QL은 SOAP 메시지의 BODY 부분에 삽입되어 SOAP 메시지가 작성된다. 작성된 SOAP 메시지는 웹 서비스 요청자를 통해 웹 서비스를 호출하게 된다, 웹 서비스 호출을 통해 SOAP 메시지는 온톨로지

저장소로 보내지게 된다. 온톨로지 저장소에서는 SOAP 메시지를 다시 파싱하여 특정 질의문으로 변환하고 질의 결과를 웹서비스를 호출했던 온톨로지 에이전트에게 반환하게 된다, 이 때, 반환되는 값 또한 SOAP 형태로 전달되어지게 된다. 온톨로지 에이전트는 전달된 SOAP에서 OWL-QL을 추출하고 이는 다시 파서에 의해 ACL 메시지로 변환되어 송신자 정보를 이용해 해당 에이전트로 전달된다.

4.1.2 변환 파서

FIPA 표준을 따르는 에이전트 환경 하에서 에이전트는 ACL 메시지를 통해 통신을 한다. ACL이 지원하는 콘텐츠 언어 중 FIPA-Semantic Language(FIPA-SL)는 ACL의 Content에 포함되며 질의를 통해 수신자 에이전트의 지식베이스에서 정보를 검색하여 추출할 수 있게 해준다. FIPA-SL은 다음과 같은 형태를 가진다.



[그림 3] 온톨로지 에이전트 구성도

A KB={P(A), Q(1, A), Q(1, B)}.

```
(query-ref
sender (agent-identifier :name B)
receiver (set (agent-identifier :name A))
content "((iota ?x (p ?x)))"
language fipa-sl
reply-with query1)
```

```
(inform
sender (agent-identifier :name A)
receiver (set (agent-identifier :name B))
content
" ((= (iota ?x (p ?x)) a) )"
language fipa-sl
in-reply-to query1)
```

FIPA-SL 사용의 예

위의 예는 에이전트 B가 에이전트 A에게 질의를 하여 A가 자신의 지식베이스에서 값을 추출하여 값을 반환한 예이다.

FIPA-SL을 확장하여 온톨로지 에이전트의 지식베이스를 온톨로지 저장소의 온톨로지로 보고 질의문을 보내게 된다. 온톨로지 에이전트는 ACL 메시지의 FIPA-SL을 읽어서 OWL-QL 형태로 변환하게 된다.

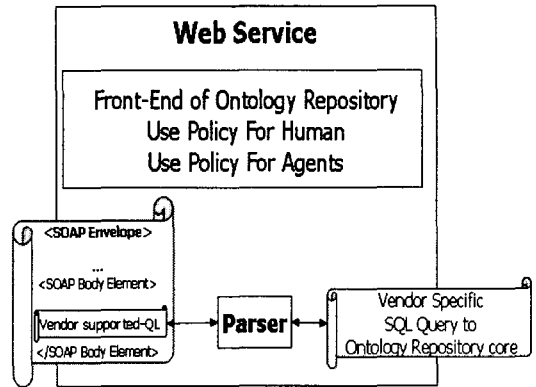
4.2 분산된 환경을 위한 온톨로지 저장소

온톨로지 에이전트는 ACL을 기반으로 목적에 적합한 범용적인 표준 질의 언어를 생성하고, 각 벤더는 표준화된 질의 언어를 자사의 구현에 적합한 질의 언어로 변환하여 표준 질의문이 지시하는 데이터를 회수, 저장할 수 있게 된다. 온톨로지 저장소는 분산되어 배치될 수 있어야 하고 구현에 독립적이며 일반적인 질의를 처리하여 주는 부분과 개발 방법론이 적용된 구현부를 분리하여 설계되어야 한다.

4.2.1 온톨로지 저장소의 전단부

[그림 4]에서 파서는 온톨로지 저장소를 개발한 벤더가 수용할 수 있는 표준 질의 언어를 해당 벤더가 제공하는 구현에 최적화 되어 있는 SQL문으로 변형 하는 역할을 한다. 각 온톨로지 저장소마다 구현부와 짝을 이루는 의존적인 파서를 가질 수 있으며 벤더가 선택한 구현의 방법론에 따른 특성을 반영하여 구현된다. 따라서 온톨로지가 사용되는 도메인에 따라 좀 더 효율적인 질의 형태가 존재 할 수 있고 일반적인 질의문을 바탕으로 가장 효율적인 질의를 할 수 있게 된다.

온톨로지 저장소는 웹 서비스의 형태로 서비스를 제공한다. 온톨로지 저장소의 URI정보를 가지

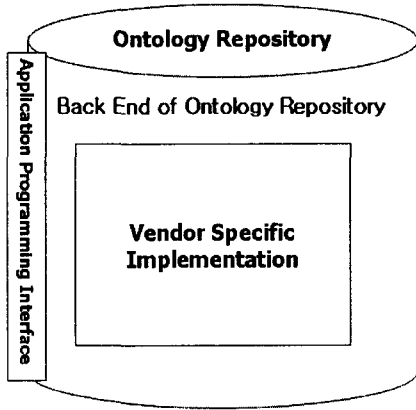


[그림 4] The Front End

는 온톨로지 에이전트는 HTTP를 통하여 분산된 온톨로지를 범용적으로 사용할 수 있게 된다. 또한 웹 서비스를 통하여 특정 영역의 전문가에게 온톨로지를 생성, 편집할 수 있는 인터페이스를 제공한다. 전문가는 어디서든지 취향에 따라 제공되는 설계 정책(예, 온톨로지 편집 틀)을 선택 하여 작업을 할 수 있다.

4.2.2 온톨로지 저장소의 후단부

제시된 대용량 온톨로지 저장소 요구사항을 모두 수용할 수 있는 강력한 온톨로지 저장소는 아직 개발 되지 않았다. 하지만 각 벤더의 구현 메커니즘에 따라 요구사항 중 일부분을 수용하는 온톨로지 저장소는 활발히 연구가 진행 중이다. 각 온톨로지 저장소는 특정 문제 영역에 적용되어 다른 온톨로지 저장소에 비하여 우월한 성능을 보이기도 하고 반대의 경우도 발생하기도 한다. 이것은 온톨로지 저장소가 서로 다른 메커니즘을 기반으로 설계, 구현되어 있기 때문에 발생하는 현상이다. 예를 들어, 데이터가 실질적으로 저장되는 RDB의 구성은 KAON, HAWK등과 같은 대표적인 온톨로지 저장소마다 서로 다른 방법을 취한다.



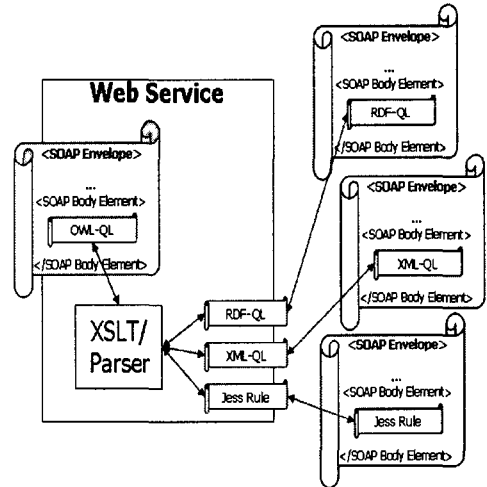
[그림 5] The Back-End

구현부분과 서비스를 제공하는 부분은 API를 통하여 연결되며 온톨로지 저장소의 데이터를 수정, 편집, 회수할 직접적인 방법도 API를 통하여 제공될 수 있다. 하지만 API는 구현부에 의존적이기 때문에 API 차원의 표준은 정해지기 어렵다.

4.3 변환 서비스

온톨로지 에이전트로부터 전송된 SOAP 메시지의 본문 요소로서 표준화된 질의 언어가 전달된다. 하지만 전달된 질의문을 변환 시켜 줄 수 있는 파서가 해당 온톨로지 저장소의 Front-End에 배치되지 않았다면 Front-End에 배치된 웹 서비스를 통하여 번역을 위한 다른 웹 서비스를 호출한다. 호출된 웹 서비스는 요청한 온톨로지 저장소가 파싱할 수 있는 질의문으로 변환을 수행한다.

[그림 6]과 같이 변환 수행을 위하여 XML문서 트리의 형태를 재구성 할 수 있는 XSL Transformations (XSLT)[15]을 고려 할 수 있으며 구체적인 파서 역시 대안이 될 수 있다. 만약 질의를 위해 웹 서비스를 경유하여 온톨로지 저장소를 사용하였다면, 질의된 결과역시 해당 웹 서비



[그림 6] Translate Service

스를 이용하여 본래의 질의 언어로 변환한 다음 전달하여야 한다. 즉, 재 변환을 위하여 변환되기 전과 변환된 후의 정보량은 같아야 한다.

5. 결론 및 토의

본고에서는 기존의 에이전트 시스템과 분산된 온톨로지 저장소의 이질적인 상호작용의 단점을 개선하기 위한 요구사항을 취합하여 다중 에이전트 시스템과 분산된 온톨로지 저장소들이 웹 서비스를 통해 일반적으로 상호작용하는 방법론을 제시 하였다. 본고에서 제시한 구조는 에이전트가 온톨로지에 질의하기 위한 절차로 내부 메시지를 변환하고 웹 서비스를 경유하기 때문에 지식을 회수하기 위한 지연시간이 길어진다. 하지만 다양한 문제에 효율적으로 사용되는 온톨로지 저장소는 구현되기 어렵기 때문에 각 문제 영역에 따라 장단점을 가지는 다양한 온톨로지 저장소로 접근하는 일반적인 방법론이 필요하다. 본고에서

는 에이전트 시스템의 의미적 상호 운용성을 광범위하게 보장할 수 있는 기반 구조를 모델링하여 다중 에이전트 시스템 개발의 생산성을 높일 수 있는 기반을 제시하였다.

참고문헌

- [1] Jacques Ferber. Multi-Agent Systems. Addison Wesley, 1999.
- [2] FIPA, FIPA specifications.
- [3] W3C, RDF Primer, W3C Working Draft 23 January 2003.
- [4] I. Horrocks, A Description Logic for the Semantic Web. IEEE Bulletin of the Technical Committee on Data Engineering, 2002.
- [5] W3C, Extensible Markup Language (XML) 1.0 (Third Edition), 04 February, 2004.
- [6] W3C, OWL Web Ontology Language Semantics and Abstract Syntax.
- [7] SWAT Projects, HAWK - OWL Repository and Toolkit, <http://swat.cse.lehigh.edu/projects/index.html#hawk>
- [8] Boris Motik, Daniel Oberle, Steffen Staab, Rudi Studer, Raphael Volz, KAON Server Architecture, Technical Report 421, University of Karlsruhe, Institute AIFB, 76128 Karlsruhe, Germany. 2002.
- [9] W3C, RDQL - A Query Language for RDF, 9 January, 2004.
- [10] Richard Fikes, Patrick Hayes, and Ian Horrocks. OWL-QL-a language for deductive query answering on the Semantic Web. J. of Web Semantics, 2(1):19-29, 2004.
- [11] W3C, SPARQL Query Language for RDF, 23 November, 2005.
- [12] W3C, Web Services Architecture, 11 February, 2004.
- [13] Jin Pan, Stephen Cranefield, Daniel Carter, A Lightweight Ontology Repository, AAMAS 2003, ACM Press, ISBN: 1-58113-683-8, New York (2003) 632-638.
- [14] F. Baader and U. Sattler. An Overview of Tableau Algorithms for Description Logics. Studia Logica, 69:5 - 40, 2001.
- [15] W3C, XSL Transformations (XSLT) Version 2.0, 3 November, 2005.

Abstract

Utilizing Distributed Ontology Repository in Multi-Agent System Environment

SungTae Kim* · KyengWhan Jee* · JungJin Yang*

The rapid growth of IT technologies enables the quality of human's daily life to be improved dramatically. Contrasting to the services in previous computing environment directed by user's request, the services in ubiquitous computing era of new IT technology are provided through recognizing user's intention and utilizing context-aware information suited to the user. According to the contextual information, agents need to set a dynamic goal to achieve and work collaboratively with other agents. Agents that take control over their behaviors with capability of communicating with other agents become a thrust in this up-coming computing environment. This work focuses on building ontologies, shared knowledge bases among agents, to improve semantical interoperability among agents. More attention is given to the construction and effective management of ontology repository along with its requirement and organization. Ontology agent suggested takes an initiative role to manage the repository in a distributed manner and to facilitate the use of ontology in a multi-agent environment.

Key words : MAS, Ontology Repository, Web Services

* School of Computer Science and Information Engineering, The Catholic University of Korea