

OWL 온톨로지 사용을 위한 SPARQL 쿼리 툴

조대웅*, 최지웅*, 김명호**

SPARQL Query Tool for Using OWL Ontology

Dae-Woong Jo *, Ji-Woong Choi *, Myung Ho Kim **

요 약

시맨틱 웹에서는 RDF, RDFS, OWL과 같은 온톨로지 구축 언어를 이용하여 웹 리소스의 메타데이터를 정의한다. 이러한 시맨틱 웹 기술을 이용한 연구는 에이전트를 기반으로 한 온톨로지의 개념에 대한 관계 및 트리플을 추출하기 위한 목적으로 진행되고 있다. 하지만 에이전트 기반의 온톨로지의 개념에 대한 관계 및 트리플의 추출은 에이전트의 특성상 한정된 쿼리 문을 작성 하게 된다. 이는 온톨로지 안의 다른 개념에 대한 관계 및 트리플을 추출하고자 할 때는 유연성이 떨어진다. 온톨로지 정보 자체에 대한 유연성 있는 정보 검색을 하기 위해서는 표준 온톨로지 언어에 접근 및 표준 쿼리 언어의 이용이 가능한 쿼리 툴이 필요 하다. 본 논문에서는 OWL 온톨로지에 HTTP 프로토콜을 이용하여 접근이 가능하고, 쿼리를 할 수 있는 SPARQL 쿼리 툴을 제안한다. 툴에 의해 생성되는 쿼리 결과는 SOAP 메시지 형태로 변환 출력함으로써 웹 서비스를 이용한 접근도 가능하다.

Abstract

Semantic web uses ontology languages such as RDF, RDFS, and OWL to define the metadata on the web. There have been many researching efforts in the semantic web technologies based on an agent for extracting triple and relation about concept of ontology. But the extraction of relation and triple about the concept of ontology based on an agent ends up writing a limited query statement as characteristics of an agent. As for this, there is the less of flexibility when extracting triple and relation about the other concept of ontology. We are need a query tool for flexible information retrieval of ontology that is can access the standard ontology and can be used standard query language. In this paper, we propose a SPARQL query tool that is can access the OWL ontology via HTTP protocol and it can be used to make a query. Query result can be output to the soap message. These operations can be support the web service.

▶ Keyword : 쿼리 툴(query tool), SPARQL, OWL, 온톨로지(ontology), SOAP, 웹 서비스(web service), BI (business intelligence)

• 제1저자 : 조대웅 교신저자 : 김명호

• 투고일 : 2009. 08. 07, 심사일 : 2009. 11. 02, 게재확정일 : 2009. 11. 03.

* 숭실대학교 IT대학 컴퓨터학과 학생 ** 숭실대학교 IT대학 컴퓨터학부 교수

※ 이 논문은 2009년 한국정보과학회 HPC 연구회 동계학술대회에 발표한 "SPARQL 쿼리 툴 HTTP 프로토콜을 이용한 OWL 온톨로지 접근 및 SPARQL 쿼리 결과 SOAP 메시지 출력"을 확장한 것이다.

※ 이 연구는 2009년도 숭실대학교 교내연구비에 의하여 수행 되었음.

1. 서론

현재 W3C(World Wide Web Consortium)에서는 공동 이용이 가능한 웹 기술들을 이끌며 발전시키고 있다[1]. 이러한 웹 기술들 중, 시맨틱 웹(Semantic Web)이라고 불리는 개념은 W3C의 팀 버너스 리에 의해 2001년에 제안된 것으로, 웹의 확장된 개념이자 기술자체를 말한다[2].

시맨틱 웹 방식을 이용한 시스템 개발은 시스템 자체를 지능적으로 만들기 보다 시스템에서 사용될 데이터를 지능적으로 만드는 방식을 택한다. 그리하여 기계가 처리 가능한 웹 데이터를 생성하고 기존 정보시스템의 한계를 극복할 수 있다. 시맨틱 웹 엔지니어들은 데이터를 지능적으로 만들기 위해 온톨로지를 구축하고 있으며 현재 W3C에서는 온톨로지를 구축하기 위한 언어로 OWL(Web Ontology Language)[3]을 표준화 하고 있다. 그림 1은 W3C의 시맨틱 웹 스택으로 현재 시맨틱 웹 기술은 OWL과 같은 온톨로지 구축언어 레이어 까지 발전된 상태이다[4]. 현재는 OWL을 이용하여, 컴퓨터와 사람 간에 공동으로 할 수 있는 작업을 효율적이고, 지능적으로 하기 위한 온톨로지 구축 중심의 연구가 진행 중이다[5]. 그래서 보다 쉽게 온톨로지를 구축하기 위한 구축 툴이 개발, 발전되고 있으며 그 중 Protege와 같은 툴이 개발되면서 이를 이용하여 효과적으로 온톨로지를 구축하고 있다[6].

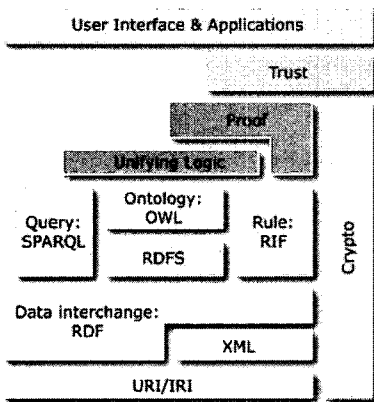


그림 1. 시맨틱 웹 스택
Fig. 1. Semantic web stack

구축된 온톨로지를 사용하면 의미 검색 또는 추론, BI 툴에서 마이닝 결과와 온톨로지 정보를 바탕으로 한 웹 데이터 분석과 같은 일을 할 수 있다. 현재는 에이전트를 기반으로 이와 같은 일을 수행하고 있다[7]. 에이전트는 온톨로지의

개념에 대한 관계 및 트리플을 자동으로 추출하여 시스템이 원하는 결과를 반환한다. 에이전트는 시스템과 OWL 저장소 사이에서 시스템이 원하는 OWL 온톨로지의 결과를 반환하기 위해 쿼리를 자동으로 수행한다. 따라서 에이전트 기반에서 온톨로지에 대한 개념의 추출은 정해진 규칙에 의한 한정된 쿼리문 작성과 정적인 형태의 쿼리 결과를 가지게 된다. 이러한 점은 OWL 온톨로지 정보에 대하여 유연성 있는 정보 검색을 힘들게 한다. OWL 온톨로지에 있는 다양한 개념의 대한 관계 및 트리플 추출을 통해 유연성 있는 정보검색을 하기 위해선 사람이 직접 쿼리를 작성할 수 있는 쿼리 툴이 필요하다. 쿼리 툴은 다음과 같은 점을 만족해야한다. 첫째, OWL 온톨로지의 접근이 로컬, 웹에 상관없이 자유로워야한다. 둘째, 온톨로지의 개념의 대한 관계 및 트리플을 추출할 수 있는 표준 온톨로지 쿼리 언어인 SPARQL(SPARQL Protocol and RDF Query Language)[9]을 이용한 쿼리가 가능하여야 한다. 셋째, OWL에 대한 디스크립션 정보를 쿼리 툴 안에서 확인할 수 있어야 한다. 넷째, 쿼리 결과에 대해 웹 서비스 형태로 이용할 수 있도록 결과 포맷을 변경할 수 있어야 한다.

알려진 쿼리 툴로는 Protege와 Twinkle[17]등이 존재한다. 하지만 이들은 HTTP 프로토콜을 이용한 URL로의 접근이 제한적이며 툴에 종속적인 형태의 OWL 온톨로지 접근만이 허용된다. 그리고 쿼리를 하기 위해 필요한 OWL 정보의 디스크립션이 되지 않아 쿼리 작성이 불편하다. 또 한, 쿼리 결과가 분산된 에이전트에게 웹 서비스 형태로 전달할 수 있도록 결과 포맷의 변경에 대해서도 지원하지 않고 있다. 따라서 규칙 기반 시스템이나 지능형 에이전트에서 필요한 개념에 대한 추출결과를 웹 서비스 형태로 이용할 수 없다.

이에 본 논문에서는 앞에서 언급한 네 가지 점을 만족하는 SPARQL 쿼리 툴을 제안한다. 본 논문에서 제안하는 SPAR-QL 쿼리 툴은 OWL 온톨로지에 대한 접근 및 디스크립션이 가능하며, 지능형 에이전트에게 개념에 대한 추출결과를 웹 서비스로 이용할 수 있도록 결과에 대한 포맷을 변경 할 수 있다. 쿼리 결과가 SOAP(Simple Object Access Protocol)[10] 메시지 형태로 변경 됨으로서 웹서비스로의 접근이 용이해지며, 이는 분산 환경에서의 OWL 온톨로지 정보의 필요한 결과를 플랫폼에 상관없이 지원이 가능하게 된다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문을 위해 사용된 개념인 OWL과 SPARQL, SOAP에 대한 간단한 설명과 다른 쿼리 툴 및 저장 툴에 대한 부분을 비교 설명한다. 3장에서는 HTTP 프로토콜을 이용한 OWL 온톨로지 접근의 필요성 및 SPARQL 쿼리 결과에 대해 SOAP 메시지로 변환했을때의 필요성에 대해 기술한다. 4장에서는 SPARQL 쿼리

리 툴에 대한 요구사항 및 설계에 대해 기술하고, 5장에서는 구현 사항 및 성능평가에 대한 부분을 기술한다. 6장에서는 결론 및 향후 발전사항에 대한 부분을 기술한다.

II. 관련 연구

본 장에서는 온톨로지 구축언어인 W3C의 OWL의 대한 부분과 OWL 온톨로지에 쿼리를 보내기 위한 SPARQL 쿼리 언어, 웹서비스 표준 프로토콜인 SOAP에 대해 알아본다. 그리고 다른 온톨로지 저작 툴 및 쿼리 툴에 대해 살펴본다.

1. OWL

OWL은 웹에서 온톨로지를 표현하기 위한 언어로 W3C에서 2002년부터 표준화 작업을 하여 현재 OWL 1.1 까지 표준화가 되었으며, OWL 2.0이 표준화 작업 중에 있다.

OWL은 기존의 RDF와 RDF Schema를 바탕으로 온톨로지를 표현하기 위한 더 많은 문법들이 추가 되었다. OWL 구문은 클래스(class), 속성(property)과 원소(individual)의 3가지 기본 요소를 가진 것으로 요약될 수 있다.

OWL은 크게 3가지의 형태가 있으며 각각은 OWL-Lite, OWL-DL, OWL-Full로 나누어진다. OWL-Full은 OWL-DL을 포함하고, OWL-DL은 OWL-Lite를 포함하는 구조로 OWL-Lite에서 OWL-Full로 갈수록 온톨로지를 표현 할 수 있는 표현력이 늘어난다[3]. 현재는 OWL을 추론(Reasoning) 할 수 있는 추론 엔진이 OWL-DL 수준까지여서 대부분의 OWL 파일은 OWL-DL의 문법에 따라 쓰여 지고 있는 추세이다.

2. SPARQL

시맨틱 웹 환경에서 웹 온톨로지 정보를 검색하기 위한 RQL[11], RDQL[12], SPARQL과 같은 다양한 질의 언어가 개발되었으며, 이 중에서 SPARQL은 W3C에 의해 권고안으로 선정되었다.

RDQL은 관계형 데이터베이스를 쿼리하기 위한 SQL과 비슷한 구조로 설계되어 SQL에 익숙한 사용자들이 사용하기 용이한 특성을 가지고 있다. SPARQL은 RDQL과 비교하여 구문(syntax)의 큰 틀은 유지하였고, 더욱 정교해진 데이터 조작과 다중 그래프 패턴 지원, 강화된 표현력 등을 추가적으로 제공한다[14].

SPARQL 온톨로지 쿼리 언어는 트리플 패턴 매칭을 기본적으로 사용하여 쿼리를 한다. 기본적인 쿼리플 패턴 방식과 크게 다르지 않음 패턴 SELECT, FROM, WHERE절을 이용하

여 쿼리를 하고, 자세한 그래프 패턴 매칭을 위해 FILTER, OPTIONAL과 같은 연산자를 사용하여 결과를 볼 수도 있다.

3. SOAP

SOAP은 웹서비스의 스펙중의 하나이며, 웹서비스의 구성 요소는 SOAP, WSDL, UDDI의 요소가 있다. SOAP은 원격 응용 프로그램간의 정보를 교환하는데 필요한 구조를 W3C에서 표준화한 것이며, WSDL은 웹서비스를 기술하는 언어이고, UDDI는 웹 서비스에 대한 검색 및 탐색을 할 수 있는 저장소 역할을 한다.

그림2는 SOAP 메시지의 구조를 나타낸 그림이다. 엔벨롭(Envelope)원소는 헤더(Header)와 바디(Body)라는 자식 원소를 갖는다. 헤더부분은 생략 가능하나, 바디부분은 반드시 기술 하여야 한다. SOAP에서 사용되는 데이터 타입은 스칼라 타입(scalar type)과 합성타입(compound type)으로 구분 할 수 있다. 스칼라 타입은 한 개의 값만 갖는 경우이고, 합성 타입은 구조체나 배열과 같이 여러 개의 값을 갖는 경우이다.

본 논문에서는 SOAP envelope안의 SOAP body의 Message body에 쿼리 결과를 지정함으로써 SOAP 메시지 구조에 맞게 쿼리 결과가 SOAP 형태의 포맷으로 변환된다.

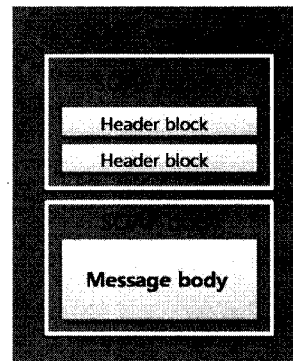


그림 2. SOAP 메시지 구조
Fig. 2. SOAP message architecture

4. 온톨로지 저작 툴 및 쿼리 툴

OWL 및 RDF 파일을 편집하고, 나타내는 데는 대표적으로 Stanford University에서 만든 Protégé[6]가 있다. Protégé는 기본적으로 OWL을 구축하기 위한 구축 툴이자, 편집 툴이다. 본 논문에서 제안하는 쿼리 툴과 같이 OWL 온톨로지에 쿼리를 하기 위해 만들어진 툴이 아니다. 물론 쿼리를 하기 위한 플러그인을 설치 함으로서 쿼리를 하고, 결과의

확인 가능하다. 하지만, 웹에 있는 OWL 온톨로지에 접근이 용이하지 않다. 즉, 웹 상의 OWL 파일의 접근은 툴의 종속적인 형태로의 접근이 요구된다. Protégé에서 쓰이는 pprj 형태인 프로젝트 파일을 서버에 올려놓아야 접근이 될 뿐, OWL 파일의 확장자인 .OWL에 대한 접근이 웹을 통해 바로 접근이 안된다. 이러한 점은 Protégé에서 만든 온톨로지 파일이 아니면 안되므로 웹 접근을 통한 온톨로지 활용에 있어서는 불편한 점이 있다. 또한, 쿼리를 하고 난 후, XML이나 SOAP형식과 같이 변환을 하여 다른 쪽으로의 확장도 지원 하지 않고 있다.

그리고 Twinkle와 같은 SPARQL 쿼리 툴은 웹 URL을 통한 OWL과 RDF로의 접근 및 SPARQL 쿼리를 작성할 수 있다. 이는 본 논문에서 제안하는 SPARQL 쿼리 툴과 다르지 않다. 하지만 OWL 온톨로지에 대한 디스크립션 정보가 있지 않아 유저 입장에서는 Twinkle만 가지고 쿼리를 하기에는 OWL 정보자체가 부족하므로 쿼리 문을 작성하여, 쿼리를 하기가 불편하다. 또한, 쿼리 결과 포맷에 있어서 XML 형태 및 SOAP 메시지에 대한 변환은 아직은 안되고 있는 부분이 본 논문의 SPARQL 쿼리 툴과는 다른 점이다.

그 외, ontoprise의 OntoStudio, Eclipse IDE 기반의 OWL 에디터인 SWeDE(Semantic Web Development Environment), MIT 에서 만든 Swoop과 같은 온톨로지 편집 툴들이 있다. 이러한 툴들은 OWL 온톨로지를 만들기 위한 저작 툴, 혹은 OWL 온톨로지에 대한 편집에 초점을 맞춘 툴이라 기본적으로 SPARQL 쿼리 자체가 불가능하다. 즉, 로컬 상태의 OWL 파일을 가져오거나 온톨로지를 만드는 데 초점이 있는 반면, 웹에 있는 OWL 파일을 접근하거나 로딩해서 SPARQL을 이용하여 쿼리 작성을 위한 기능을 제공하지 않고 있다. 물론 쿼리가 안되므로, 쿼리 결과에 대한 XML, SOAP 같은 포맷 변화 또한 제공 하지 않고 있다. 이와 같은 점을 표 1에서 각 툴의 기능과 본 논문에서 제안하는 툴과의 기능상의 차이점을 설명하고 있다.

표 1. 쿼리 툴 성능 비교
Table 1. Query tool performance comparison

툴/기능	기능	SPARQL 쿼리	웹 접근	디스크립션	SOAP/XML 변환
SPARQL 쿼리 툴 (제안)	쿼리	O	O	O	O
Protégé	저작, 쿼리	O	O (중속적)	O	X
Twinkle	쿼리	O	O	X	X
Ontoprise	저작, 편집	X	X	O	X
SWeDE	저작, 편집	X	X	O	X
Swoop	저작, 편집	X	X	O	X

본 논문에서 제안하는 SPARQL 쿼리 툴은 기존의 툴 종속적인 형태의 OWL 온톨로지 접근 및 쿼리 툴에서 OWL 정보에 대한 디스크립션이 잘 되지않아 SPARQL 쿼리 작성에 어려웠던 점을 보완 하였으며, 쿼리 결과에 대해서도 웹 서비스의 접근이 가능하도록 하였다. 즉, HTTP 프로토콜을 이용하여 웹에 있는 OWL 온톨로지에 접근을 하고, SPARQL 쿼리를 할 수 있다. 쿼리 결과에 대해서는 XML, SOAP 메시지 형태로 변환 출력 할 수 있다. 변환 출력된 SOAP메시지는 웹서비스를 위한 SOAP 프로토콜로써 사용을 할 수 있으며, 이는 지능형 에이전트에게 웹 서비스를 통한 특정 쿼리 정보를 교환 할 수 있는 방법이 될 수 있다.

III. 필요성 및 활용방안

본 장에서는 HTTP 프로토콜을 이용한 OWL 온톨로지 접근의 필요성에 대한 부분과 SOAP 메시지 변환의 필요성 그리고 본 논문에서 제안하는 쿼리 툴의 활용 방안에 대해 살펴본다.

1. HTTP 프로토콜을 이용한 OWL 온톨로지 접근의 필요성

시맨틱 웹이 되기 위해선 기존의 HTML로 이루어진 웹 페이지가 아닌 온톨로지 정보를 바탕으로 모든 웹이 하나로 연결된 상태여야 한다. 그랬을 때, 기계가 알아서 웹에 있는 온톨로지 정보를 바탕으로 의미를 처리해서 결과를 보여주는 시맨틱 웹을 이룰 수 있게 된다. 이러한 조건이 되기 위해선 우선 웹에 이미 온톨로지 파일들이 있다는 가정이 이루어져야 한다. 때문에 표준 프로토콜인 HTTP를 이용하여 웹에 접근을 통한 온톨로지 파일의 파싱은 구축된 온톨로지 정보를 활용하기 위해선 필요한 부분이다.

앞에서도 언급한 Protege와 같은 온톨로지 저작 툴에서는 HTTP 프로토콜을 이용한 OWL 온톨로지에 접근이 제한적이다. 즉, 다른 웹 상에 있는 OWL 온톨로지 url로의 접근은 허용이 안되며, 웹을 이용한 접근을 하려면 자체 서버를 만들어서 그 안에 OWL 파일을 놔두고 접근을 해야하는 불편한 점이 있다. 이러한 방법으로는 웹에 존재하는 OWL 온톨로지 파일을 이용한 활용에 제한적일 수 밖에 없다.

따라서 본 논문에서는 구축된 온톨로지가 웹에 있고, 그것을 활용하기 위한 방법으로 먼저 선행되어야 할 것이 표준 프로토콜인 HTTP를 이용한 OWL 온톨로지 접근이다. 그래서 HTTP를 이용하여 OWL 온톨로지를 가져와서 온톨로지 정보를 볼 수 있도록 본 논문의 SPARQL 쿼리 툴에서는 제공한다.

2. SOAP메시지 변환 필요성

SOAP은 분산 환경에서 구조적이고 타입이 있는 정보를 XML을 이용해서 교환하기 위한 표준 프로토콜로서 자리 잡아 가고 있다. 기존에도 IIOP와 ORPC와 같은 원격함수 호출 프로토콜이 있으나 바이너리 형태로 이루어져서 플랫폼이 다양한 인터넷 환경에서는 적당하지 않다. SOAP은 XML을 사용하는 텍스트 기반의 프로토콜이기 때문에 웹과 같은 환경에서 메시지를 전달하기가 용이하다. 또 한, SOAP은 HTTP 프로토콜 위에서 실행된다. HTTP 프로토콜은 모든 플랫폼의 웹 서버와 웹 브라우저에서 지원이 되므로 호환성 및 접근성의 측면이 높다. 이러한 점 때문에 본 논문에서는 쿼리 결과를 SOAP 메시지로 변환 출력 한다.

그림3에서 설명하는 것은 SOAP을 이용하여 원격 메소드를 호출하기 위한 구문이다. 그림에서 표시된 쿼리 결과라고 표시된 곳에 SPARQL 쿼리 툴에서 나온 결과를 스트링 형식으로 변환하여 메시지 바디에 넣는다. 이러한 방법으로 변환된 SOAP 메시지는 원격 함수 호출 및 메시지 전달에 웹서비스 형태로 전달될 수 있다.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <ns1:echoString xmlns:ns1="http://soapinterop.org/">
      <msg xsi:type="xsd:string">Message</msg>
    </ns1:echoString>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

쿼리 결과

그림 3. SOAP 요청
Fig. 3. SOAP request

3. 활용방안

그림 4는 본 논문에서 제안한 SPARQL 쿼리 툴을 활용한 그림이다. 웹에 OWL, RDF와 같은 온톨로지 파일들이 있다는 가정 하에 SPARQL 쿼리 툴에서는 HTTP 프로토콜을 이용하여 특정 온톨로지 파일에 접근한다. 접근된 파일은 올바른 쿼리를 보내기 위해 온톨로지를 트리 형태로 시각화 시킨다. 사용자는 시각화된 온톨로지 정보를 바탕으로 올바른 쿼리를 보낼 수 있다. 쿼리 엔진에서 쿼리에 대한 결과를 XML 형태로 저장 할 수 있으며, 이와 같이 저장된 XML 쿼리 결과는 다음번에 다시 쿼리를 보낼 때 참고 할 수 있다. 즉, 저장된 XML 문서를 불러와서 계속적인 작업을 요청 할 수 있게 된다.

쿼리 결과를 SOAP 메시지로도 변환이 가능한데 이것은 SOAP 변환 엔진에서 담당한다. 변환된 SOAP 메시지는 분산 환경에서 웹 서비스로의 전환 시 필요한 사항이 된다. 변환된 SOAP 메시지는 SOAP 프로토콜을 이용해서 웹서비스로의 접근이 가능하다.

그림4의 메인 에이전트는 웹 서비형태로 서비스 되고 있으며, SPARQL 쿼리 툴에서는 쿼리 결과를 SOAP 프로토콜을 이용해서 메인 에이전트에게 보낸다. 메인에이전트는 받은 쿼리결과를 SOAP 프로토콜을 이용해서 분산된 다른 에이전트들에게 웹서비스 형태로 메시지를 전달한다. 각각의 에이전트들은 받은 SOAP 메시지를 바탕으로 쿼리 결과가 필요한 에이전트끼리 SOAP을 이용해서 다시 통신을 한다. 각각의 에이전트는 마이닝 엔진에서 추출한 정보를 가지고 있으며, 마이닝 정보와 온톨로지 쿼리 결과를 토대로 서버에 위치한 레포팅 툴에 전송한다. 전송된 정보를 바탕으로 레포팅 툴에서는 다변적으로 분석을 하여 올바른 레포팅이 될 수 있도록 작성 할 수 있다.

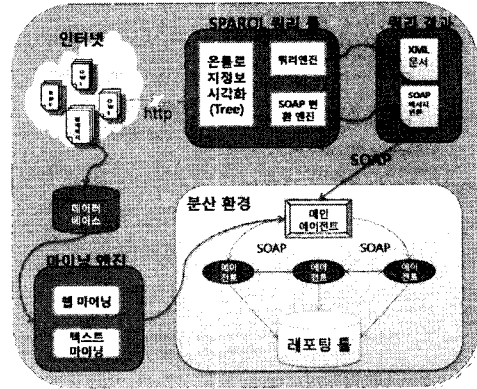


그림 4. SPARQL 쿼리 툴 활용방안
Fig. 4. SPARQL query tool practical use

이와 같은 활용방안은 BI적인 측면에 근거 한 활용 방안이었으며, 그 외에도 분산시스템에서 SOAP 메시지를 이용한 메시지의 전달은 웹 서비스를 이용한 환경에서 표준으로 자리 잡아 가고 있다. 따라서 이와 같이 전달하고자하는 메시지를 SOAP으로 변환하는 것에 대한 요구사항은 더 많아 지게 된다.

IV. 시스템 요구사항 및 설계

본 장에서는 본 논문에서 제안하는 SPARQL 쿼리 툴 시스템의 요구사항과 쿼리 툴의 시스템 설계에 대해 살펴본다.

1. 시스템 요구사항

본 논문에서 소개하는 SPARQL 쿼리 툴을 구현하기 위한 시스템 요구사항은 다음과 같다.

1) HTTP 프로토콜을 사용한 OWL 온톨로지 접근이 이루어져야 한다. 2) OWL파일의 클래스구조(ClassHierarchy) 및 제한(Restriction)을 포함한 클래스 묘사(Description) 정보들을 볼 수 있도록 UI로 나타내 주어야 한다. 3) SPARQL 문법을 모르는 일반 사용자를 고려하여 SPARQL 쿼리를 직접 쿼리 문 작성이 아닌 UI를 통해 작성을 할 수 있어야 한다. 4) SPARQL 쿼리 결과를 나타낼 수 있는 UI가 필요하다. 5) SPARQL 쿼리 결과를 XML 형식으로 저장하고, 다시 그 파일을 불러 왔을 때 결과가 로딩 되어서 계속적인 분석이 이루어질 수 있도록 한다. 6) SPARQL 쿼리 결과를 다른 인텔리전트 에이전트와의 정보 교환을 위해 웹 서비스 프로토콜인 SOAP 형태로 변환이 요구된다.

2. 쿼리 툴 시스템 설계

앞 장에서 살펴본 시스템 요구사항을 바탕으로 시스템에서 필요한 기능의 스택은 그림 5와 같다.

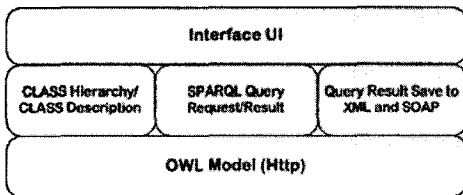


그림 5. SPARQL 쿼리 툴 시스템 스택
Fig. 5. SPARQL query tool system stack

그림 5의 SPARQL 쿼리 툴 시스템 구조에서 가장 아래의 OWL Model(Http)은 시스템 요구사항 1)번에 해당하는 부분으로 HTTP 프로토콜을 이용해서 OWL 온톨로지에 접근하여 온톨로지 정보를 SPARQL 쿼리 툴 안에서 내부 모델화 시킨다.

그 위의 클래스 구조/ 클래스 묘사(CLASS Hierarchy/CLASS Description)에서는 모델화 된 OWL 온톨로지의 클래스를 트리 형태로 나타내 주기 위한 부분과 클래스의 다른 제약사항이나

관계를 정의한 프로퍼티 부분, OWL 온톨로지의 인디비주얼 리스트(Individual List)정보를 보여주기 위한 로직이 들어가 있다. 이것은 요구사항 2)번을 충족하기 위한 것이다.

요구사항 3), 4)번에 해당하는 부분이 그림 5의 SPARQL Query Request/Result 부분이며, 여기서 모델화된 OWL 온톨로지에 쿼리를 요청하고 결과를 받아 올 수 있는 로직이 담겨 있다. 그리고 쿼리 결과에 따라 XML 파일 형식 또는 SOAP 메시지 형태로 출력, 저장 될 수 있도록 하는 부분이 그림 5에서 두 번째 레이어 가장 끝에 있는 Query Result to Save X-ML and SOAP 부분이다. 이것은 요구사항 5), 6)번에 해당하는 부분이다. 마지막으로 모든 요구사항에 대한 부분은 사용자 입장에서 쉽게 쿼리를 작성하고 정보를 보기 위해 자바의 Swing을 이용한 Interface UI로 구현되어져 있다.

그림 6은 SPARQL 쿼리 툴의 클래스 다이어그램으로 Main 클래스에서 SView클래스의 객체를 참조하여 뷰 화면을 로딩 한다. SView에는 SPAQRQL 쿼리 툴 시스템 스택에서 언급한 HTTP 프로토콜을 이용하여 접근을 할 수 있는 주소창, 클래스 정보 트리출력, SPARQL 쿼리를 입력할 수 있는 입력 창 및 결과 화면 표시, 쿼리 결과를 변환 할 수 있는 사용자 관점의 인터페이스를 구성 할 수 있는 클래스들을 참조한다.

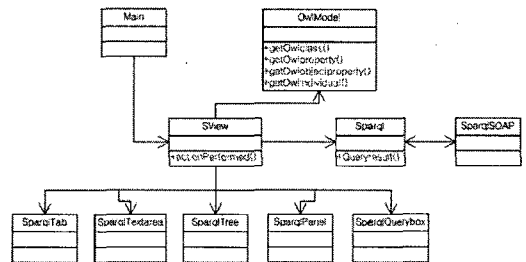


그림 6. SPARQL 쿼리 툴의 클래스 다이어그램
Fig. 6. Class diagram of SPARQL query tool

HTTP 프로토콜을 위한 접근과 클래스, 프로퍼티, 인디비주얼 정보들은 OwlModel 클래스의 객체로부터 참조하여 가져온다. 그래서 SView에서 리턴된 정보를 화면에 보여준다. 또 한, 사용자가 입력창에서 SPARQL 쿼리를 보내게 되면 내부에서 Sparql 클래스의 객체를 생성, 참조한다. 그리고 쿼리 결과에 대해 SOAP 형태로 변환 시 Sparql 클래스 안에 Queryresult() 메소드를 통해 SparqlSOAP 객체를 참조한다. 그래서 SOAP 형태로 SPARQL 쿼리 결과를 변환 출력 하게 된다.

V. 구현 및 성능평가

본 장에서는 본 논문에서 제안하는 SPARQL 쿼리 툴의 구현을 위해 사용된 API와 실제로 구현된 툴의 인터페이스에 대해 살펴본다. 그리고 SPARQL 쿼리 툴의 성능에 대해 평가한다. 성능평가는 HTTP 프로토콜을 이용한 접근이 제대로 되고 있는지에 관한 것과 SPARQL 쿼리 작성 여부 및 SOAP 메시지 변환에 대해 살펴본다.

1. API

다음은 본 논문에서 구현한 SPARQL 쿼리 툴에서 사용된 API에 대한 설명이다. 기본 베이스는 Java 6을 기반으로 UI는 Swing을 사용하였다. OWL 파일의 모델화 및 SPARQL 쿼리 툴을 위한 API로는 Jena2, Protégé, ARQ가 사용되었으며, 기본 설명은 다음과 같다.

Jena2 - Jena는 시맨틱 웹 응용프로그램을 만들기 위한 자바 프레임워크(framework)다. Jena는 RDF, RDFS와 OWL, SPARQL을 프로그램 하기 위한 API를 제공한다[14].

Protégé-owl - Protégé-owl API는 OWL과 RDF를 위한 Java 기반의 오픈소스이며, 좀 더 풍부한 UI의 API를 위해 사용되었다[6].

ARQ - ARQ는 SPARQL 및 RDQL과 같은 RDF, OWL 쿼리 언어를 지원하기 위한 Jena 기반의 쿼리 엔진이다[15].

2. 인터페이스

SPARQL 쿼리 툴은 웹 및 로컬에 존재하는 OWL 온톨로지에 접근이 가능하다. 접근된 OWL 온톨로지는 트리 형태로 사용자에게 보여줄 수 있다. SPARQL 쿼리 툴 내부에서 정리된 OWL 파일의 디스크립션 정보는 사용자가 SPARQL 쿼리 문을 작성 하는데 도움을 준다. SPARQL 쿼리 문법에 맞는 쿼리 문의 작성을 통해 원하는 OWL 트리플 정보 및 결과에 대해 SOAP 메시지로 변환 출력할 수 있다.

그림 7에서는 실제 웹 상에 존재하는 OWL 파일을 주소창을 통해 접근하여 트리형태로 OWL 정보를 보여주고 있는 전체적인 툴의 모습이다. Individual 탭을 통해 OWL 파일의 인디비주얼 리스트도 볼 수 있다. 사용자는 인디비주얼 리스트와 클래스 구조를 SPARQL 쿼리를 작성하기 위해 참고한다. 그림 7에서 우측 가장 위 부분에는 이 OWL파일의 Prefix 부분에 대한 정보를 나타내주고 있으며, 탭을 통한 컨디션에서 관련된 클래스의 리스트리션 정보가 나타난다. 그

림 7에서 가장 밑에는 SPARQL 쿼리를 보낼 수 있는 쿼리 창이며, Prefix창 하단에는 사용자가 쿼리 창에서 작성한 쿼리에 대한 결과가 나타나는 부분이다.

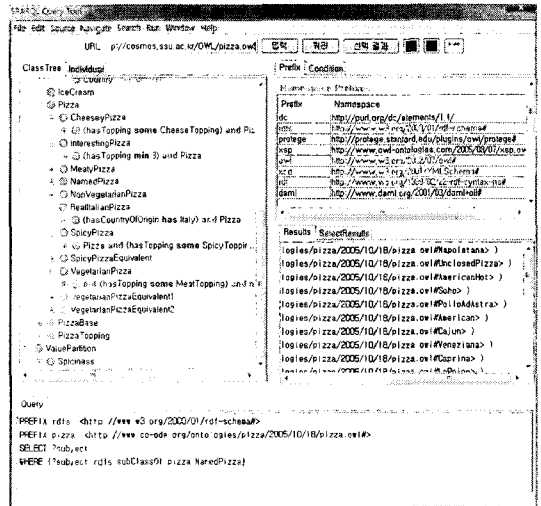


그림 7. SPARQL 쿼리 툴의 인터페이스
Fig. 7. Interface of SPARQL query tool

3. 성능평가

본 절에서는 본 논문에서 제안하는 SPARQL 쿼리 툴의 성능에 대해 평가한다. 성능평가는 HTTP 프로토콜 접근에 대한 부분과 SPARQL 쿼리가 제대로 되고 있는지에 대한 부분 및 SOAP 메시지 변환에 대해 평가를 한다.

3.1 HTTP 프로토콜 접근

그림 7에서는 로컬에 있는 OWL 온톨로지 파일을 로드해서 보여주고 있는 것이 아닌, 다른 곳에 존재하는 OWL 온톨로지에 HTTP 표준 웹 프로토콜을 이용해서 접근하고 있는 모습이다. 사용된 웹 url은 http://cosmos.ssu.ac.kr/OWL/에 위치한 pizza.owl 파일에 접근해서 로드 한 것이며, 그에 관한 정보가 클래스, 서브클래스, 제약조건, 접두어로 나누어서 사용자에게 일목요연하게 OWL 디스크립션이 이루어 지고 있다. 현재는 클래스 정보 및 인디비주얼 리스트 정보를 가지고 쿼리 작성이 가능하다.

3.2 SPARQL 쿼리

그림 7의 Query창에 작성된 SPARQL 쿼리 문은 "NamedPizza클래스의 하위 클래스는 무엇인가?"라는 쿼리를 수행하고 있는 모습이다. NamedPizza는 SPARQL 쿼리 툴에서 접근한 pizza.owl의 Pizza클래스의 하위 클래스로서

그에 대한 하위 Pizza클래스의 결과가 Results창에 출력되고 있으며, 올바른 쿼리 결과를 확인할 수 있었다. 이외, 다른 쿼리 문을 더 작성해서 테스트 해보았다. OWL 그래프는 그림 8과 같다.

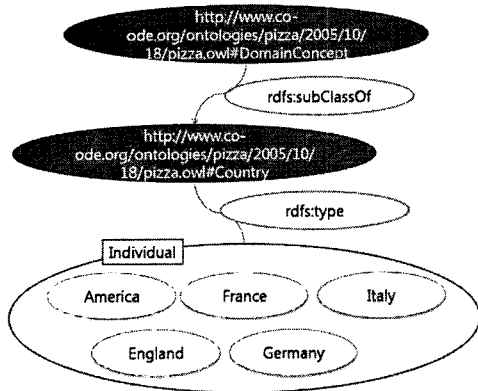


그림 8. OWL 그래프의 예
Fig 8. Example of owl graph

그림에서 DomainConcept클래스가 최상위 클래스이며, 하위 클래스로 Country클래스가 있다. Country클래스는 5가지의 Individual을 가지고 있었으며, 각각은 America, France, Italy, England, Germany이다.

이러한 OWL 그래프를 가지고 있는 정보를 도출하기 위해 간단한 쿼리를 작성하였으며, 그에 대한 결과가 다음 표2와 같다. pizza.owl에서 표기된 uri(<http://www.co-ode.org/ontologies/pizza/2005/10/18/pizza.owl#>)을 따르고 있으며, prefix를 pizza로 명명하여 쿼리를 작성하였다.

표 2. SPARQL 쿼리 및 결과 값
Table 2. SPARQL query and result value

쿼리	결과 값
PREFIX rdf: < http://www.w3.org/1999/02/22-rdf-syntax-ns# > PREFIX rdfs: < http://www.w3.org/2000/01/rdf-schema# > PREFIX pizza: < http://www.co-ode.org/ontologies/pizza/2005/10/18/pizza.owl# > SELECT ?individual WHERE (?individual rdfs:type pizza:Country)	Italy Germany France England America

Country는 제약조건이 "∩ {America England France Germany Italy} and DomainConcept" 이었으며 그에 대한 조건을 틀 안에서 확인을 하였다. 그리고 Country가 가지고 있는 인디비주얼 리스트를 쿼리 함으로써 앞선 제약 조건에 맞는 리스트가 반환이 되었고, 제약조건에 맞는 Country 클래스의 특성을 확인할 수 있었다.

3.3 SOAP 변환

제안하는 틀에서는 쿼리 결과를 가지고 SOAP 메시지로 변환하여 웹서비스를 호출한다. 서비스는 쿼리 결과를 받아서 다시 리턴해주는 간단한 서비스를 서버에 올려놓고 테스트를 해보았다. 서버환경은 Tomcat 6.0을 사용하였으며, 웹서비스는 Axis 1.1을 가지고 서비스를 올렸다. 쿼리결과가 SOAP으로 변환되고, 그에 대한 확인을 위해 TCPMonitor를 이용하였다. TCPMonitor는 클라이언트와 서버 사이에 전달되는 SOAP 메시지를 확인할 수 있는 툴이다.

그림 9의 TCPMonitor는 두 개의 창이 존재하며 왼쪽 창은 클라이언트에서 보내는 SOAP 메시지이며, 오른쪽 창은 서버에서 받는 SOAP 메시지이다. 메시지를 통해 쿼리 결과가 SOAP 프로토콜로 변환되어 서버의 서비스를 호출했음을 확인할 수 있다. 클라이언트에서 보낸 메시지는 표2에서 작성한 "Country 클래스에 속해있는 Individual 리스트를 보여 달라"의 쿼리 결과 중 하나의 Individual이 보내지고 있는

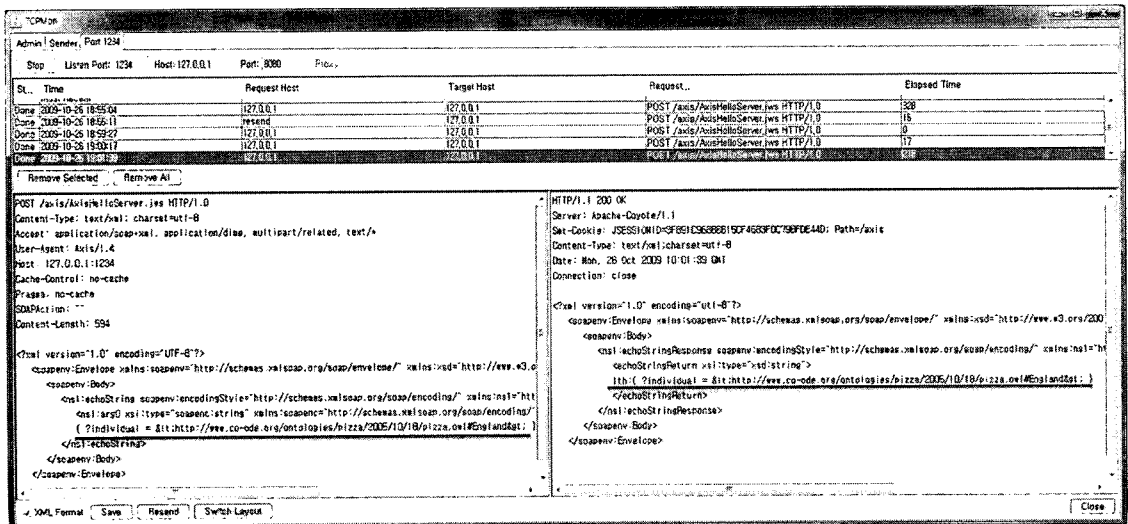


그림 9. TCPMonitor를 이용한 SOAP 메시지의 확인

모습이다. 그 부분은 밑줄 친 부분을 통해 확인을 할 수 있다. 간단한 서비스와 쿼리 문을 통해 틀에서 제안한 기능 등은 무리 없이 되고 있음을 확인 할 수 있었다.

VI. 결론 및 향후 과제

시맨틱 웹 연구에서 핵심 요소는 데이터들을 온톨로지화 하여 메타데이터를 풍부하게 하는 것이다. 온톨로지를 이용하여 현재는 에이전트를 기반으로 한 온톨로지의 개념간의 관계 및 트리플 추출로 연구가 진행 되어 왔다. 하지만 이러한 에이전트 기반의 온톨로지의 개념간의 관계 및 트리플의 추출은 한정된 쿼리 및 정형적인 결과를 가져왔다. 이는 온톨로지 안의 다른 개념간의 관계 및 트리플의 추출에 있어선 유연성이 떨어지는 측면이 있다. 따라서 온톨로지의 유연성 있는 검색을 하기 위해선 쿼리 틀이 필요하다. 이에 본 논문에서는 표준 온톨로지 언어에 접근이 자유롭고, 표준 쿼리 언어를 이용한 쿼리가 가능한 쿼리틀을 제안하였다. 제안한 SPARQL 쿼리 틀은 웹에 있는 OWL 온톨로지에 HTTP 프로토콜을 통해 접근을 하고, SPARQL 쿼리 및 결과를 SOAP 메시지 형태로 변환 출력할 수 있었다.

SPARQL 쿼리 틀은 기존에 틀에 종속적인 모습의 OWL 온톨로지 접근이 아닌 표준 온톨로지 파일(OWL, RDF)로 바로 접근을 한다. 그리하여 OWL 파일 안의 클래스, 프로퍼티, 인디비추얼과 같은 온톨로지 정보들을 가져와서 확인하고, S-PARQL 쿼리를 통해 원하는 트리플을 검색이 가능하였다.

또 한, SPARQL 쿼리 결과를 SOAP 메시지 형태로 변환하여 웹서비스로의 접근이 용이하도록 결과 포맷을 변환 할 수 있었다. 그렇게 함으로써 SOAP 프로토콜을 이용한 웹서비스로의 접근이 용이 하기 때문에 분산 시스템에서 이러한 요구사항은 많을 것으로 기대 되며, BI로의 접근을 위한 레포팅 툴로의 활용방안에 대해서도 앞서 살펴보았다.

향 후 보완사항에서는 자동으로 SPARQL 쿼리를 작성하고자 한다. 즉, 사용자는 원하는 오브젝트 또는 트리플 결과를 얻기 위해 쿼리 문을 입력할 필요 없이 조건 및 제어를 할 수 있는 콤보 박스를 마련해서 그곳에서 원하는 조건들을 클릭 하면 쿼리가 자동으로 매칭이 되는 형식으로 보완하고자 한다. 이렇게 했을 시 장점은 정확한 쿼리 문법을 몰라도 OWL 파일에서 원하는 결과 값을 이끌어 낼 수 있다는 장점이 있다.

둘째, 본 논문에서 제안한 활용방안에 대한 부분이다. 레포팅 툴[16]에 SPARQL 쿼리 툴 을 결합하여 웹 상의 OWL 파일을 읽어 들여 원하는 개념에 대한 관계 및 트리플을 추출 할 수 있다. 그렇게 함으로써 레포팅 틀에서 온톨로지 정보를 바탕으로 한 레포팅이 이루어 질수가 있다. 이러한 BI로의 접근을 통해 SPARQL 쿼리 툴을 활용하겠다는 것이 향 후 과제로써 남아 있다.

참고문헌

- [1] W3C, <http://www.w3.org/>
- [2] T. Berners-Lee, J. Hendler, and, O. Lassila, "The Semantic Web," ScientificAmerican, May 2001.
- [3] OWL, <http://www.w3.org/TR/owl-guide/>
- [4] Semantic Web, <http://www.w3.org/2001/sw/>
- [5] J. Seidenberg, A. Rector, "Web Ontology Segmentation: Analysis, Classification and Use," WWW 2006, pp.13-22, Edinburgh, Scotland, UK, May 2006.
- [6] Protégé, <http://protege.stanford.edu>
- [7] W. Ketter, A. Batchu, G. Berosik, D. McCreary, "A Semantic Web Architecture for Advocate Agents to Determine Preferences and Facilitate Decision Making," 10th ICEC'08, pp.1-10, Innsbruck, Austria, Aug. 2008.
- [8] D. Sell et al, "SBI: A Semantic Framework to Support Business Intelligence," ACM International Conference Proceeding Series: Vol. 308, Article No. 11, OBI'08, Karlsruhe, Germany, Oct. 2008.
- [9] SPARQL, <http://www.w3.org/TR/rdf-sparql-query/>
- [10] SOAP, <http://www.w3.org/TR/soap/>
- [11] RQL, <http://139.91.183.30:9090/RDF/RQL>
- [12] RDQL, <http://www.w3.org/Submission/RDQL/>
- [13] 김대현, 권준희, "SPARQL: 시맨틱 웹에서의 온톨로지 쿼리언어," 데이터베이스연구, 제 23권, 제 1호, 53-63쪽, 2007년 4월.
- [14] Jena2, <http://jena.sourceforge.net>
- [15] ARQ, <http://jena.sourceforge.net/ARQ/>
- [16] 최지용, 김명호, "웹 기반의 Ad Hoc 리포팅을 위한 Fat-client를 갖는 리포팅 툴," 정보과학회논문지, 제12권, 제 4호, 264-274쪽, 2006년 8월.
- [17] Twinkle, <http://www.ldodds.com/projects/twinkle/>
- [18] 선복근, 위다현, 한광록, "OWL 온톨로지를 기반으로 하는논문 검색 시스템에 관한 연구," 한국컴퓨터정보학회논문지, 제14권, 제2호, 169-180쪽, 2009년 2월.
- [19] 이순미, "시맨틱 웹 문서를 위한 관계형 저장 스키마 설계 및 질의 처리기법," 한국컴퓨터정보학회논문지, 제 14 권, 제 1호, 35-45쪽, 2009년 1월.

저자 소개



조 대 응

2008: 한림대학교 컴퓨터공학과 학사
2008 - 현재: 송실대학교 컴퓨터학과 석사과정

관심분야: 분산/병렬 컴퓨팅, 시맨틱 웹, BI, 웹 서비스, 보안



최 지 응

2001: 송실대학교 컴퓨터학부 학사
2003: 송실대학교 컴퓨터학과 공학석사
2007 - 2008: 고등기술연구원 연구원
2003 - 현재: 송실대학교 컴퓨터학과 박사과정

관심분야: 분산/병렬 컴퓨팅, 시맨틱 웹, BI, 보안, 유비쿼터스



김 명 호

1989: 송실대학교 컴퓨터학부 학사
1991: 포항공과대학교 전자계산학과 공학석사

1995: 포항공과대학교 전자계산학과 공학박사

1995: 한국전자통신연구소 선임연구원
1998, 2006: 미국 테네시주립대 교환교수

1995 - 현재: 송실대학교 컴퓨터학부 교수

관심분야: 분산/병렬 컴퓨팅, 그리드, 웹서비스, BI, 보안