URI 서버 내 DB 스키마 자동 생성 연구[1]

정한민[0] 이미경 강인수 성원경
한국과학기술정보연구원 정보서비스연구팀
jhm@kisti.re.kr

Study on the Automatic Generation of DB Schema in URI Server

Hanmin Jung[0] MiKyung Lee In-Su Kang Won-Kyung Sung
Information Service Research Lab., KISTI

요 약

본 연구는 시맨틱 데이터 정합성 검사와 RDF 트리플 생성 기능을 가진 URI 서버 내 DB 스키마의 자동 생성 방안을 기술한다. URI 서버는 시맨틱 웹 기술 기반 프레임워크 구성에 필요한 핵심 엔진으로서 인스턴스 생성 관리기, 검색 엔진, 추론 엔진 등과의 상호 작용을 통해 시맨틱 데이터를 서비스한다. 도메인이 바뀌거나 서비스가 변경되는 경우 DB 스키마를 전문가에 의해 수작업으로 생성함으로써 서비스 환경 변화에 즉각 대응하지 못하는 문제점을 가진 기존 URI 서버를, 적재되는 온톨로지에 따라 DB 스키마를 동적으로 자동 생성하는 방식으로 변경함으로써 다양한 응용 분야와 도메인에의 높은 이식성(Portability)을 보장할 수 있도록 한다. 기반정보 온톨로지, 응용 온톨로지, 개인화 온톨로지 등 3개 온톨로지 스키마와 11만 건 이상의 Citeseer Open Access Metadata로부터 추출된 인스턴스를 대상으로 한 실험을 통해 URI 서버 내 DB 스키마 자동 생성 가능성을 실증하였다.

1. 서론

시맨틱 웹에서의 온톨로지는, 온톨로지 간 통합 및 상호운용성(Interoperability)을 보장하기 위해 국제 표준에 따라 기술될 필요가 있다. 또한, 구축된 온톨로지의 정합성(integrity) 보장을 위한 장치가 시맨틱 웹 프레임워크 구성을 위해 필수적으로 요구된다. 현재 온톨로지 기술을 위한 국제 표준으로는 데이터 모델 관점에서 W3C의 RDF(Resource Description Framework)와 ISO의 토픽 맵(Topic Maps)이 있으며, 제약언어 관점에서 W3C의 RDFS, OWL과 ISO의 TMCL(Topic Map Constraint Language)이 존재한다[8]. 온톨로지 정합성 보장을 위해서는, 온톨로지 개발 방법론 측면에서 모델링 대상이 되는 도메인에 대해 요구 사항 분석, 개념·속성 도출, 온톨로지 스키마 설계, 응용 지향적 스키마 검증 등의 단계를 반복 수행하여 온톨로지 스키마의 실세계 정합성을 검사하게 된다. 특히, 기술 논리(Description Logics)에 기반을 둔 OWL 온톨로지의 경우 별도의 검증 도구를 사용하여 온톨로지 스키마의 논리적 일관성 검사를 추가로 수행한다.

그러나, 온톨로지 정합성 확보 측면에서 현재의

[1] 관련 특허: 11/576457(미국출원), 11/576442(미국출원), PCT/KR2006/005535(PCT출원), PCT/KR2006/004096(PCT출원), 10-2006-0114957(국내출원, 10-2006-0081788(국내출원)

온톨로지 기술 표준, 개발 도구, 개발 방법론 및 검증 도구가 중요하게 고려되지 못하는 부분이 있는데, 온톨로지 정합성이 모델링 대상이 되는 실세계 영역과 온톨로지 간의 투명성을 의미한다고 볼 때, 현재의 온톨로지 기술은 실세계 개체를 온톨로지로 사상시키는 프로세스에 대한 통제 부족에 기인한다. 다시 말하자면, 현재 온톨로지 기술에서는 온톨로지 내 특정 클래스의 인스턴스가 등록될 때 기존에 등록된 인스턴스들과의 중복 검사나, 인스턴스 속성의 응용 의존적 범위 검사에 대한 지원 장치가 부재하거나 빈약하다.

연구 개발 전주기 지원 서비스를 위한 OntoFrame은 시맨틱 웹 기술 기반 프레임워크로서 URI 서버를 이용하여 상기 문제에 대처하고 있다[2][3]. 개념적으로 URI 서버는 모델링 대상이 되는 실세계와 온톨로지 사이에 위치하여, 온톨로지에 등록되는 인스턴스의 정합성을 검사하고, 온톨로지 데이터의 저장소 역할을 수행하며, 특정 인스턴스 등록과 동시에 RDF 트리플을 생성한다. URI 서버는 온톨로지 개체의 정합성 검사를 위해 관계형 데이터베이스 기술을 활용한다. 즉, 온톨로지 스키마 상의 클래스, 속성(Property), 클래스 간 관계를 E-R모델에서의 개체(Entity), 속성(Attribute), 관계(Relationship)로 대응시키고, 온톨로지에서 결여된 클래스 내 인스턴스의 고유성 보장을 위해 Key Attribute를 지정한다.

기존 URI 서버는 이러한 차별화된 기능에도 불구하고 도메인이 바뀌거나 서비스가 변경되는 경우 DB 스키마를 전문가에 의해 수작업으로 생성함으로써 서비스 환경 변화에 즉각 대응하지 못하는 문제점을 가지고 있었다. 이에 대한 하나의 해결 방안으로서 온톨로지와 동적으로 연계되는 URI 서버 내 DB 스키마 자동 생성이 있으며, 본 연구는 여기에 초점을 맞추고자 한다.

본 연구의 목적은 결국 다양한 도메인과 응용 분야에 시맨틱 웹 기술 기반 프레임워크를 쉽게 이식하고자 그 중심이 되는 URI 서버를 온톨로지에 따라 자동 생성하는 방안을 제시하는 데 있다. 2장에서는 관련 연구를, 3장에서 URI 서버 내 DB 스키마 자동 생성 기법을 설명한다. 4장에서는 실제 적용을 통해 URI 서버 내 DB 스키마 자동 생성 가능성을 실증하고자 한다.

2. 관련 연구

URI 서버에 대응하는 개념으로 온톨로지 서버, 온톨로지 관리기 등이 있다. 온톨로지 관리에 대한 선구적인 연구는 [6]에 의해 시작되었는데, 여기에서는 온톨로지와 중재자(Mediator)를 통한 다양한 자원들로부터 시맨틱 데이터로의 통합 모델을 제시하였다. 비록 이론적인 연구지만, 이미 오래 전에 온톨로지를 통한 자원 관리의 중요성을 인지했다는 데 그 중요성이 있다.

[4][9]는 데이터베이스 중심적(Database-centric) 구조를 이용하여 온톨로지 데이터를 저장하고 관리하는 기법을 제시하였다. 또한, 추론(Reasoning)의 역할을 동시에 부여하기 위해 통합 인스턴스 테이블(Fact Table) 등을 구축하였다. 그렇지만, 이들은 인스턴스 테이블을 단순화시킴으로써 시맨틱 데이터의 정합성 체크, DB 스키마의 유연성, 사용자 정의 추론 규칙 처리의 어려움 등 또 다른 부작용을 가져오는 문제점을 가지고 있다. 대용량 지식을 저장하고 관리하기 위해 DBMS를 사용하고, 대용량 처리의 실현 가능성을 보여준 또 다른 연구로 [5][7]이 있는데, 특히, [7]에서는 'Knowledge-relational Mapping'이라는 개념을 통해 RDBMS와 온톨로지 간 변환 방식을 제안하였으며, 온톨로지 스키마와 인스턴스를 분리하여 적재함으로써 더 높은 성능 향상을 꾀하고자 하였다. 그렇지만, 초기 온톨로지로부터 RDBMS를 생성하는 방안은 제시하지 못하고 있어 본 연구와는 초점을 달리한다.

온톨로지 관리의 다른 목적으로는 다중 온톨로지 간 병합(Merge)이 있지만[12], 본 연구에서 사용하는 다중 온톨로지는 병합보다는 독립적 관리 대상이 되므로 본 논문에서는 더 이상 언급하지 않는다. URI 서버의 한 기능인 RDF 트리플 생성 방법을 제안한 연구도 있는데[13], 스키마 비인지(Schema-oblivious) 형태의 트리플 구조 생성을 통해 추론을 시도하고 있다는 점에서 본 연구의 일부와 유사하다.

3. URI 서버 내 DB 스키마 자동 생성

3.1절에서는 URI 서버의 개념과 관련 시스템과의 상호작용을 설명하고, 3.2절에서는 본 연구의 초점인 클래스 관리기가 담당하는 온톨로지에 따른 DB 스키마 동적 생성을 설명한다.

3.1 URI 서버

데이터의 정합성 검사와 기 등록 인스턴스와의 중복 검사를 위해 도입한 URI 서버는 기본적으로 온톨로지 인스턴스를 저장·관리하는 기능을 담당한다. URI 서버는 온톨로지 스키마를 등록하고 관련 DB 스키마를 생성하는 클래스 관리기(3.2절 참조), 사용자 정보를 등록하고 관리하는 사용자 관리기, DB 내의 인스턴스를 관리하는 인스턴스 관리기, DB 내용을 RDF 트리플로 변환하는 트리플 생성기, sameAs 처리를 담당하는 sameAs 관리기[1] 등으로 구성된다. 트리플 생성기는 인스턴스를 RDF 트리플 형식으로 변환하여 추론 시스템이 직접 사용할 수 있는 형태로 제공하는 역할을 한다.

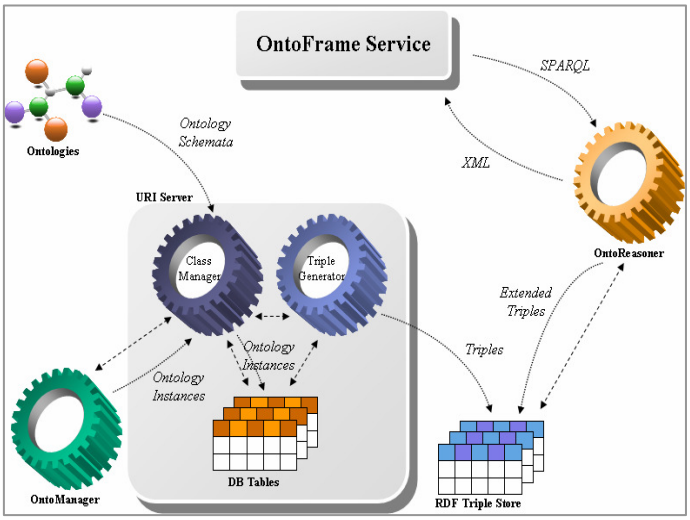

그림1. URI 서버 연계도(신규 구축 환경)

연구 개발 전주기 지원 서비스를 위한 시맨틱 웹 기술 기반 프레임워크인 OntoFrame은 서비스 실현을 위해 추론 엔진인 OntoReasoner, 검색 엔진, Open API 관리기 등과 상호작용을 한다. 추론 엔진에게는 SPARQL 형식의 질의를 사용하여 사용자 요청을 전달하고, XML 형식의 결과를 웹서비스를 통해 제공받는다(그림1, 그림2 참조). OntoFrame이 적용되는 환경은 크게 두 가지로 나눌 수 있는데, 첫째는 기존 환경의 영향을 받지 않고 신규로 프레임워크를 구축하는 환경이며, 둘째는 기존 시스템(Legacy System) 및 DB 등 외부와 연동되는 환경이다.

전자의 경우에는 그림1과 같이 클래스 관리기를 통해 온톨로지를 적재하고 DB 스키마를 자동 생성한 후 인스턴스 생성 관리기인 OntoManager와의 상호작용을 통해 인스턴스를 등록한다. 등록된 인스턴스는 트리플 생성기인 Triple Generator를 통해 RDF 트리플로 변환되어 추론 엔진에게 제공된다. 추론 엔진은 전방 추론(Forward Chaining) 방식으로 트리플 형태의 지식을 확장하고 서비스 요구에 따라 추론 결과를 제공한다.

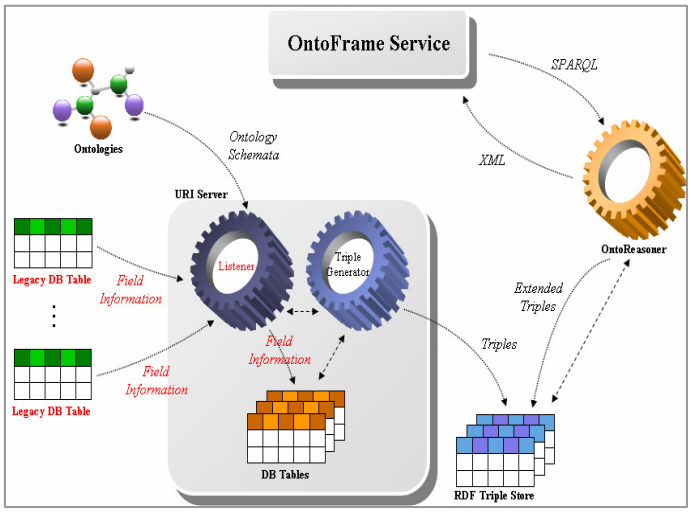

그림2. URI 서버 연계도(외부 연동 환경)

후자의 경우에는 그림2와 같이 수집기이자 중재자인 Listener가 클래스 관리기를 대체하는 방식으로 구성된다. 수집기는 클래스 관리기와 같이 온톨로지를 적재하고 DB 스키마를 자동 생성하는 기능 이외에 기존 DB로부터 특정 필드 정보를 수집하여 이를 URI 서버 내 DB에 적재하는 방식으로 인스턴스를 생성한다. 이러한 방식은 기존 시스템과 환경을 훼손하지 않으면서 OntoFrame을 적용시킬 수 있다는 장점을 가진다. 3.2절에서는 본 연구의 초점인 클래스 관리기와

Listener가 공통적으로 가지고 있는 기능인 온톨로지 적재와 DB 스키마 자동 생성을 다루고자 한다.

3.2 클래스 관리를 통한 DB 스키마 자동 생성

그림3은 기반정보 온톨로지 'ResearchRefOntology' 내의 'Person' 클래스에 대한 클래스 관리 화면을 보여준다.

그림3. URI 서버 클래스 관리기 예 ('Person' 클래스)

온톨로지 스키마를 적재하면 스키마에 정의된 모든 클래스를 관리할 수 있는 환경이 생성된다. 그림3과 같이 'Person' 클래스를 선택하면 URI 서버 내 DB 스키마 자동 생성을 위해 설정해야 할 정보가 보이는데 이를 설정하면 해당 클래스와 매핑되는 DB 스키마가 URI 서버 내에 자동 생성된다. 주 설정 대상인 속성(Property)은 다음과 같은 상세 정보를 가진다.

- T: RDF 트리플 생성 여부. 체크되는 경우 해당 속성에 대해 트리플을 생성한다.
- D: 인스턴스 생성 관리기에서의 출력 여부. 체크되는 경우 온톨로지 인스턴스를 편집하는 인스턴스 생성 관리기에서의 인스턴스 간략보기가 가능해진다.
- R: 필수 필드 여부. 체크되는 경우 인스턴스 생성 관리기에서 해당 필드에 정보가 반드시 입력되어야 한다.
- S: 검색 대상 여부. 체크되는 경우 검색 엔진에 의해 해당 필드 내용이 수집·색인되어 검색 가능해진다.
- Property: 객체 관계 속성(Object Property)이나 데이터타입 속성(Datatype Property) 이름
- Range: RDF 트리플에서 객체(Object)에 해당. 이 필드에 클래스 이름(예. 'CategoryAreaOfPerson', 'Department' 등)이 오는 경우 객체 관계 속성을 의미하며, 'string', 'date', 'number' 등 데이터타입이 오는 경우 데이터타입 속성을 의미한다.
- Label: 속성 설명
- DataType: 생성될 DB 내 필드의 데이터타입
- Length: 생성될 DB 내 필드의 크기
- Operation: 인스턴스 생성 관리기 내 편집 인터페이스의 필드 유형. 'Text'는 문자열 입력, 'File'은 파일 선택, 'Object'는 인스턴스 검색 인터페이스가 제공되어야 함을 의미한다.
- Format: 인스턴스 생성 관리기 내 편집 인터페이스의 필드 형식. 필드에 입력되는 문자열을 검증하기 위한 용도이다.
- Cardi: 온톨로지 스키마에 정의된 계량수(Cardinality)

표1. URI 서버가 자동 생성한 'Person' 클래스에 매핑되는 DB 스키마 예

```
create table Person_hasCategoryAreaOfPerson(
        uri varchar(255) ,
        related_uri varchar(255),
        primary key (uri, related_uri));
```

```
create table Person_hasDepartmentOfPerson(
        uri varchar(255) ,
        related_uri varchar(255),
        primary key (uri, related_uri));
```

```
create table Person_hasInstitutionOfPerson(
```

```
        uri varchar(255) ,
        related_uri varchar(255),
        primary key (uri, related_uri));
create table Person_hasTopicAreaOfPerson(
        uri varchar(255) ,
        related_uri varchar(255),
        primary key (uri, related_uri));
create table Person_standForSameAsGroupOf(
        uri varchar(255) ,
        related_uri varchar(255),
        primary key (uri, related_uri));
create table Person(
        URI varchar(255) primary key,
        CREATION_DATE DATETIME ,
        CREATOR varchar(255) ,
        LASTUPDATE_DATE DATETIME ,
        LASTUPDATER varchar(255) ,
        emailAddressOfPerson VARCHAR(255),
        engNameOfPerson VARCHAR(255),
        korNameOfPerson VARCHAR(255),
            urlOfPerson VARCHAR(255));
```

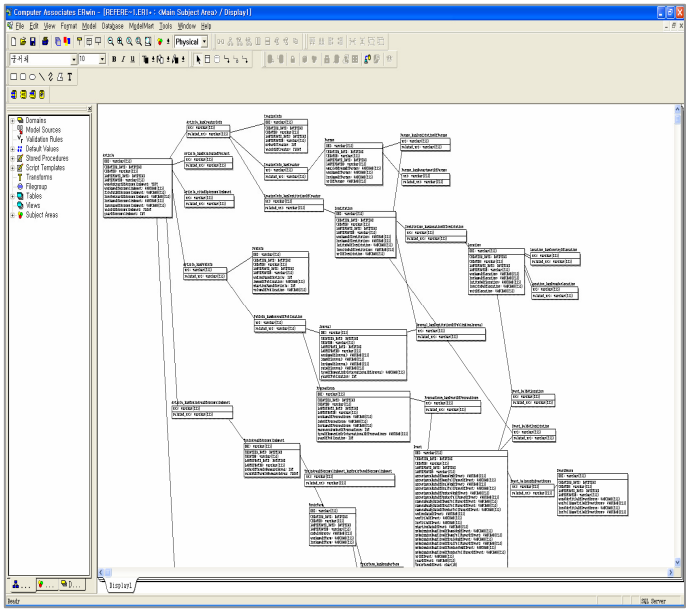

그림4. 'ResearchRefOntology' 온톨로지에 대한 ERD

특정 클래스에 대해, 상기 클래스 관리기 설정 후 자동 생성되는 DB 스키마에서의 테이블 개수는 (객체 관계 속성 개수 + 1)이다. 1에 해당하는 것은 클래스 자체 테이블이며, 여기에는 클래스 인스턴스 URI와 데이터타입 속성이 저장된다. 객체 관계 속성 테이블에는 클래스 인스턴스 URI와 객체(Object, Range) URI가 저장된다(표1의 마지막 DB 스키마). 표1은 그림3에서 정의된 'Person' 클래스 정보로부터 자동 생성된 DB 스키마를 보여주며, 그림4는 온톨로지 내 전체 클래스로부터 만들어진 DB 스키마의 ERD(Entity-Relationship Diagram)를 보여준다.

4. 실험 결과

본 연구의 실험을 위해 기반정보 온톨로지, 응용 온톨로지, 개인화 온톨로지 등 3개 온톨로지의 스키마를 동시에 적재하고 그 인스턴스를 등록하였다(표2 참조).

표2. 다중 온톨로지 스키마 정보

기반정보 온톨로지('ResearchRefOntology')	
클래스	27
데이터타입 속성	83
객체 관계 속성	28
깊이	2
응용 온톨로지('ResearchPortalAppOntology')	
클래스	9
데이터타입 속성	13
객체 관계 속성	15
깊이	1
개인화 온톨로지('ResearchPortalPersonalizationOntology')	
클래스	1
데이터타입 속성	0
객체 관계 속성	8
깊이	0

온톨로지 간 연계 구조는 기반정보 온톨로지를 중심으로 하여 응용 온톨로지와 개인화 온톨로지가 기반정보 온톨로지에 정의된 클래스를 객체(Object)로서 가리키는 형태이다. 예를 들어, 응용 온톨로지 'ResearchPortalAppOntology'의 'Picture' 클래스는 'hasRegistererOfContents' 속성으로 기반정보

온톨로지에 정의된 'Person' 클래스를 가리키며, 개인화 온톨로지 'ResearchPortalPersonalizationOntology'의 'User' 클래스는 'hasInterestedInstitution' 속성으로 기반정보 온톨로지에 정의된 'Institution' 클래스를 가리킨다.

표3. 주요 클래스, 인스턴스 개수, 적재 소요 시간

클래스 이름	인스턴스 개수	적재 소요 시간(분)
Article	114,423	102
Person	136,022	20
Location	451,762	97
TopicAreaOfAccomplishment[2]	510,478	63
Term	160,567	35

인스턴스는 Citeseer Open Access Metadata로부터 수집한 2000년 이후 논문으로부터 추출되었다. 현재 관리되고 있는 주요 클래스와 인스턴스 수는 표3과 같으며, 표4는 트리플 생성기에 의해 생성된 트리플 정보를 보여준다.

표4. 트리플 정보

클래스 이름	RDF 트리플 개수
Accomplishment	3,707,795
Cited	107,597
Institution	98,565
Location	3,162,089
Person	272,042
Term	521,543
Total	7,869,631

DB 스키마는 클래스 관리기 설정 완료와 동시에 자동으로 생성되므로 별도의 시간을 필요로 하지 않는다. 자동 생성된 DB 스키마에 두 가지 방법으로 인스턴스가 등록되는데, 첫 번째 방법은 표3과 같이 일괄 적재하는 것이며, 두 번째 방법은 인스턴스 생성 관리기를 통해 상호작용적(Interactive) 방식으로 등록하는 것이다. 후자의 경우, 3.1절에서 설명한 바와 같이 클래스 관리기에서 설정하는 상세 정보에 의해 제약된 인스턴스 생성 관리기와 URI 서버가 AJAX 방식으로 통신하며 입력되는 각 필드 값을 실시간에 검사함으로써 데이터 정합성을 보장한다.

[2] 논문에 자동 부착된 주제로 논문이 어떤 주제를 연구하는 가를 표현하는 방법이다.

5. 결론

본 연구는 시맨틱 데이터 정합성 검사와 RDF 트리플 생성 기능을 가진 URI 서버의 자동 생성을 통해 다양한 도메인과 응용 서비스에 쉽게 이식될 수 있는 방법과 그 결과를 제시하였다. URI 서버의 다양한 장점을 살리면서 높은 이식성을 보장한다면 결국 시맨틱 웹 프레임워크의 확산이 더욱 용이해질 수 있다는 점에서 본 연구의 의미를 찾을 수 있다. 향후 연구는 대용량 인스턴스 적재 효율성을 강화하고, 실시간 인스턴스 적용 모델링을 통해 즉각적 서비스를 가능하게 하는 데 초점을 맞추고자 한다. 또한, 현재 온톨로지 관리의 또 다른 이슈인 온톨로지 진화(Ontology Evolution) 문제를 다루기 위해 정합성을 보장하는 범위에서 DB 스키마 및 필드 내용 변경을 포함한 동적 대처에 대한 연구를 진행할 예정이다[7][8].

참고 문헌

[1] 강인수, 정한민, 이승우, 김평, 이미경, 성원경, 시맨틱 웹 온톨로지에서의 OWL sameAs 적용, 정보과학회논문지: 소프트웨어 및 응용 34(4), 2007.

[2] 구희관, 정한민, 강인수, 성원경, 이승준, 심빈구, 국가 과학기술 R&D 기반정보 온톨로지 구축을 위한 URI 관리 및 서비스 시스템 구현, 한국컴퓨터종합학술대회(KCC), 2006.

[3] 정한민, 강인수, 구희관, 이승우, 성원경, URI 서버에 기반한 국가 R&D 기반정보 온톨로지 설계 및 구현, 정보관리연구 37(2), 2006.

[4] S. Bechhofer, I. Horrocks, and D. Turi, The OWL Instance Store: System Description, In Proceedings of the

20th International Conference on Automated Deduction, 2005.
[5] W. Ceusters and P. Martens, LinkFactory: an Advanced Formal Ontology Management System, www.isi.edu/~blythe/kcap-interaction/papers/LinKFactoryXWhiteXPaperXfinal.doc
[6] A. Cui, V. Tamma, and F. Bellfemine, Ontology Management in Enterprises, In Journal of BT Technology Journal 17(4), 1999.
[7] A. Das, W. Wu, and D. McGuinness, Industrial Strength Ontology Management, In Proceedings of the International Semantic Web Working Symposium, 2001.
[8] L. Garshol, Living with Topic Maps and RDF: Topic Maps, RDF, DAML, OIL, OWL, TMCL, In Proceedings of XML Europe, 2003.
[9] J. Lee and R. Goodwin, Ontology Management for Large-Scale Enterprise Systems, IBM Research Report RC23730(W0509-109), 2005.
[10] A. Liang, Enabling Active Ontology Change Management within Semantic Web-based Applications, A mini-thesis submitted for transfer from MPhil to PhD, 2006.
[11] A. Maedche, B. Motik, L. Stojanovic, R. Studer, and R. Volz, Managing Multiple Ontologies and Ontology Evolution in Ontologging, In Proceedings of the Conference on Intelligent Information Processing, 2002.
[12] N. Noy and M. Musen, Ontology Versioning in an Ontology Management Framework, In journal of IEEE Intelligent Systems 19(4), 2004.
[13] C. Yeh and R. Lin, Design and Implementation of an RDF Triple Store, In Proceedings of the 1st Workshop of Digital Archive Technology, 2002.