

# ISO/IEC 11179 기반의 온톨로지 확장 모델 (An Ontology Population Model based on ISO/IEC 11179)

정혜진<sup>†</sup>      백두권<sup>\*\*</sup>      정동원<sup>\*\*\*</sup>  
(Hyejin Jeong)      (Doo-Kwon Baik)      (Dongwon Jeong)

**요약** 이 논문에서는 ISO/IEC 11179 기반의 온톨로지 확장 모델을 제안한다. 시맨틱 웹과 웹 2.0의 결합을 위한 많은 연구가 진행되고 있으며, 이를 웹 3.0으로 정의하고 있다. 웹 3.0을 구현하기 위한 가장 중요한 이슈는 온톨로지 스키마를 정의하고 온톨로지를 위한 인스턴스를 확장하는 것이다. 이를 위해서는 먼저 온톨로지 스키마를 보다 정확하게 정의하여야 하며 웹 자원으로부터 온톨로지를 보다 풍부하게 확장할 수 있는 방법이 개발되어야 한다. 이 논문에서는 표준화 된 공통 개념을 이용하고 관리하기 위해 개발된 국제 표준인 ISO/IEC 11179 - 메타데이터 레지스트리(Metadata Registry, MDR) 기반의 웹 온톨로지 확장 모델을 제안한다.

**키워드** : 시맨틱 웹, ISO/IEC 11179, 메타데이터 레지스트리, 온톨로지, 온톨로지 확장

**Abstract** This paper proposes an ontology population model based on ISO/IEC 11179. Much research has recently been done on harmonizing Web 2.0 and the Semantic Web, and the harmonization is defined as Web 3.0. The most important issues for realizing Web 3.0 include defining ontology schemas and populating instances for ontologies. To resolve the issue, Web ontology schemas should be precisely defined, and a method for populating Web ontology from Web resources should be developed. This paper proposes a Web ontology population model based on ISO/IEC 11179 - Metadata Registry (MDR), which is the international standard, developed to manage and use common standard concepts.

**Key words** : Semantic Web, ISO/IEC 11179, Metadata Registry, Ontology, Ontology Population

## 1. 서론

시맨틱 웹과 웹 2.0의 결합으로 정의되는 웹 3.0이 대두되면서 이를 구체화하기 위한 많은 연구가 진행되고 있다. 웹 3.0의 핵심 기술 중 하나는 시맨틱 웹이다[1]. 시맨틱 웹은 기존의 웹을 확장한 웹으로 1998년 팀 버

너스 리에 의해 제안되었다. 이는 웹 정보에 잘 정의된 의미를 부여하여 사람뿐만 아니라 컴퓨터도 쉽게 문서의 의미를 해석할 수 있도록 하여 컴퓨터를 이용한 정보의 검색·해석·통합 등의 업무를 자동화하기 위한 것이다[2,3].

이러한 시맨틱 웹을 구현하기 위해서는 온톨로지의 구축이 필수적으로 요구되며 현재 온톨로지의 관리 및 활용을 위한 다양한 편집기, 온톨로지 기술 언어, 저장소 및 추론 엔진 등이 개발되었다. 이러한 애플리케이션으로 인하여 온톨로지의 구축이 용이하기 때문에 이제 는 구축된 온톨로지를 응용 시스템에 활용하기 위하여 보다 풍부한 정보를 제공하기 위한 온톨로지의 확장에 대한 연구가 요구되고 있다. 온톨로지 확장은 크게 온톨로지 스키마의 확장과 온톨로지 클래스의 인스턴스 확장으로 구분할 수 있다. 두 가지의 방법 중 이 논문은 보다 풍부한 정보를 제공하는 데에 직접적으로 영향을 줄 수 있는 온톨로지 인스턴스의 확장에 초점을 둔다. 온톨로지 클래스의 인스턴스를 확장하기 위해서는 먼저 온톨로지의 보다 정확한 정의가 요구된다. 온톨로지에

· 2008년 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구(KRF-2008-314-D00485)

† 학생회원 : 군산대학교 정보통계학과  
xhyejin86x@gmail.com  
\*\* 종신회원 : 고려대학교 컴퓨터학과 교수  
baikdk@korea.ac.kr  
\*\*\* 종신회원 : 군산대학교 정보통계학과 교수  
djeong@kunsan.ac.kr  
논문접수 : 2009년 7월 6일  
심사완료 : 2009년 9월 1일

Copyright©2009 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

대한 보다 정확한 정의 및 확장을 위해서 요구되는 사항은 다음과 같다[4-8].

- 보다 정확한 온톨로지 정의를 위하여 일관성 있고 보편적인 개념 즉, 표준화된 공통 개념을 이용한 온톨로지의 스키마 생성과 이를 기반으로 한 온톨로지 간 상호운용성의 향상
- 보다 풍부한 정보 제공을 위한 온톨로지 확장 방법의 개발

이 논문에서는 앞서 언급한 온톨로지의 확장을 위해 요구되는 사항들을 충족시키기 위하여 ISO/IEC 11179 - 메타데이터 레지스트리(Metadata Registry, MDR)를 기반으로 한 온톨로지 확장 모델을 제안한다. ISO/IEC 11179는 ISO/IEC에서 개발한 국제 표준으로서 이를 기반으로 한 많은 MDR이 개발되었다[10]. MDR은 표준 개념, 즉 일관성 있는 의미 정보를 관리하기 때문에 이를 기반으로 생성된 데이터베이스 간 의미 공유 및 교환이 용이하며 보다 정확한 의미 교환이 가능한 온톨로지 생성 및 확장이 가능하다. 다시 말해, MDR의 표준 데이터 요소를 이용하여 정의한 실제 데이터베이스의 데이터(인스턴스)를 기존에 생성한 온톨로지에 추가함으로써 보다 풍부한 정보 제공 및 서비스 개발이 가능하다. 따라서 이 논문에서는 이러한 MDR의 특징을 이용하여 온톨로지를 확장할 수 있는 새로운 모델을 제안한다.

이 논문의 구성은 다음과 같다. 제2장에서는 MDR의 기본 구조 및 특징, 제3장에서는 제안 모델의 전체적인 개념과 프로세스에 대하여 기술한다. 제4장에서는 구현 및 평가에 대하여 기술하고 마지막으로 제5장에서는 결론 및 향후 연구에 대하여 기술한다.

## 2. 관련 연구

이 장에서는 ISO/IEC 11179, 즉 MDR 모델에 대한 개념 및 특징에 대하여 소개한다.

### 2.1 ISO/IEC 11179 - 메타데이터 레지스트리(Metadata Registry, MDR)의 개념

MDR은 ISO/IEC JTC 1/SC 32에 의해 개발된 국제 표준으로, 데이터 의미의 표현, 등록, 관리, 교환, 공유를 주목적으로 한다. MDR 표준문서는 식별, 명명, 정의, 분류 등에 중점을 두어 교환 및 공유 될 데이터의 의미, 표현방식 등에 대하여 이용자의 이해를 돕기 위한 메타데이터의 등록과 유지를 위한 지침을 제공하여 레지스트리에 등록된 정보의 상호운용성 확보에 중요한 기반을 제공한다[9].

MDR은 구현 내용의 재사용과 공유를 용이하게 하기 위하여 메타 모델의 형태로 기술되어 있으며 가장 중요한 개념적 모델의 요소는 다음과 같다.

- 데이터 요소 개념(Data Element Concept) : 특정 데이터에 관한 개념(또는 의미)을 나타내며, 데이터 모델링 관점에서 보면 객체 클래스(Object Class)와 프로퍼티(Property)로 구성되며, 표현하고자 하는 데이터 요소의 상황적 의미를 표현한다.
  - 개념 영역(Conceptual Domain) : 값 의미(Value Meaning)의 집합으로, 모든 값 영역의 의미는 개념 영역을 통해 나타난다.
  - 값 영역(Value Domain) : 특정 데이터 요소에 허용된 값(Permissible Value)의 집합이다.
  - 데이터 요소(Data Element) : 생산, 관리, 공유의 가장 기본이 되는 데이터 단위로 데이터 요소 개념과 표현(Representation)과 연계되었을 때 생성된다.
- 그림 1은 MDR의 핵심 구성 요소인 개념 영역, 값의 영역, 데이터 요소 개념, 데이터 요소 간의 관계를 보여준다[10].

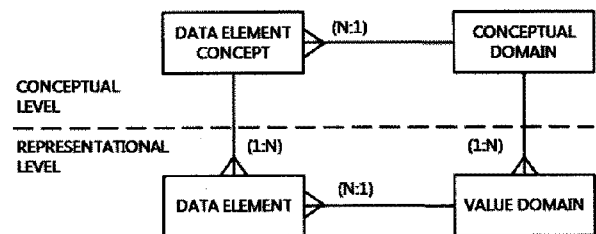


그림 1 ISO/IEC 11179 - Metadata Registry 모델

하나의 개념 영역은 여러 개의 데이터 요소 개념을 가질 수 있으며 하나의 데이터 요소 개념은 여러 개의 데이터 요소를 가질 수 있다. 또한 하나의 값 영역은 여러 개의 데이터 요소를 가질 수 있으며, 하나의 개념 영역은 여러 개의 값 영역을 가질 수 있다.

MDR에 의해 관리되는 데이터의 기본 단위는 위의 요소들 중에서 데이터 요소이며, 그림 2에서 볼 수 있듯이 다음과 같은 3가지 구성 요소로 이루어진다[10].

- 객체 클래스(Object Class) : 현실 세계의 아이디어, 추상 개념, 사물의 집합으로 동일한 규칙을 따르는 행동, 의미, 명시적 범위들로 식별할 수 있다.
- 프로퍼티(Property) : 객체 클래스의 객체들이 갖는 공통적인 특성이다.
- 표현(Representation) : 데이터의 형식 즉, 값 영역, 데이터 타입, 표현 클래스, 그리고 필요한 경우에는 차수를 포함할 수 있다.

### 2.2 MDR 구현 시스템

현재 MDR 모델을 기반으로 시스템을 구현한 나라는 호주, 캐나다, 영국, 미국 등이 있다. 구현된 MDR 구현 시스템에 대한 요약 정보는 표 1과 같다.

표 1에서 알 수 있듯이, 현재 구현된 MDR 구현 시

표 1 MDR 구현 시스템

Organization	Registry
Australian Institute of Health and Welfare	Metadata Online Registry (METeOR)
Canadian Institute for Health Information	CIHI Data Dictionary
UK Cancer Grid	Cancer Grid Metadata Registry
US National Cancer Institute	Cancer Data Standards Repository (caDSR)
US Environmental Protection Agency	Environmental Data Registry
US Health organizations (multiple)	US Health Information Knowledgebase (USHIK)
U.S. Department of Homeland Security (DHS) and U.S. Department of Justice (DOJ)	US National Information Exchange Model (NIEM)
US Department of Justice	Global Justice XML Data Model (GJXDM)

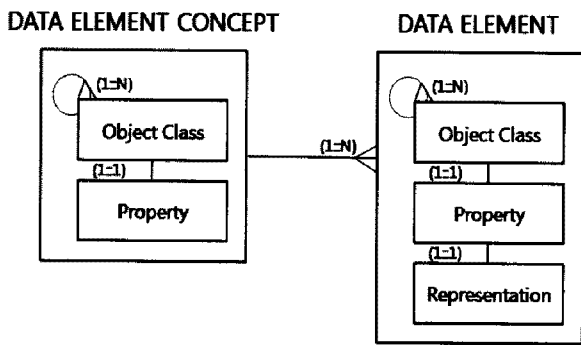


그림 2 데이터 요소 기본 모델

시스템은 호주의 보건 복지부의 메타데이터 온라인 레지스트리, 캐나다의 보건 정보부의 CIHI 데이터 사전, 영국의 암 그리드 기관의 암 그리드 MDR, 미국의 국가 암 협회의 암 데이터 표준 저장소, 미국 환경 보호청의 환경 데이터 레지스트리, 미국 보건 단체의 미국 보건 정보 지식베이스, 미국의 홈랜드 보호부와 법무부의 미국 국내 정보 교환 모델, 미국 법무부의 세계의 사법 XML 데이터 모델이 있다[11-17].

### 3. 제안 모델

이 장에서는 제안 모델의 개념과 전체적인 구조, 프로세스에 대하여 기술한다.

#### 3.1 제안 모델의 개념 및 구조

그림 3은 제안 모델의 구조도를 보여준다. 제안 모델은 표준화된 공통 개념, 즉 MDR의 구성 요소들을 이용하여 온톨로지 스키마를 보다 정확하게 정의한다. 또한 MDR의 구성 요소들을 이용하여 데이터베이스 스키마 필드를 정의하고 실제 인스턴스를 생성된 온톨로지에 추가함으로써 온톨로지 클래스에 대한 인스턴스들을 풍부하게 확장한다. 제안 모델은 크게 4개 부분으로 구성되며 다음과 같다.

- (1) ISO/IEC 11179 표준 지침에 따라서 MDR의 구성 요소인 객체 클래스, 프로퍼티, 데이터 요소 개념, 값 영역, 개념 영역, 데이터 요소 등을 MDR 구현 시스템에 등록하고 관리 하는 부분
- (2) MDR 구현 시스템에 등록된 MDR 구성 요소의 표준화된 공통개념을 이용하여 데이터베이스 스키마 필드를 정의하는 부분

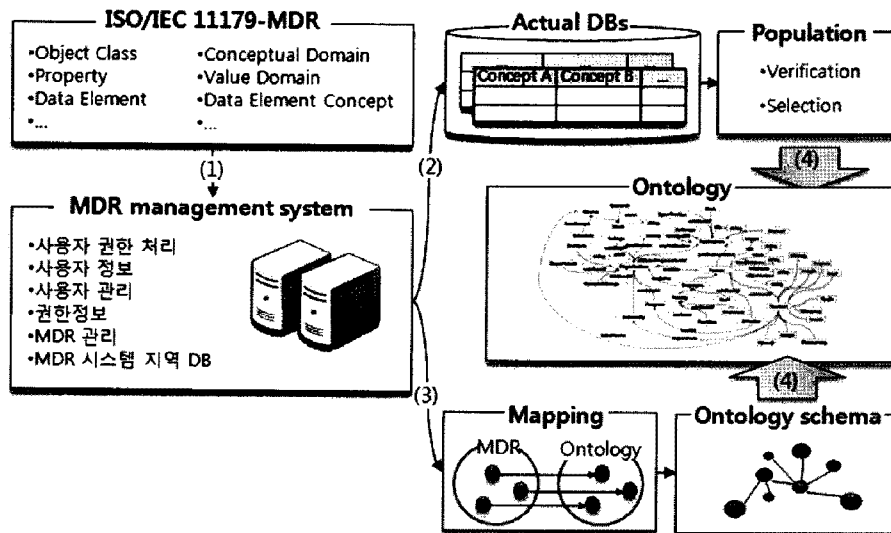


그림 3 전체적인 제안 모델의 구조도

(3) MDR 구현 시스템에 등록된 MDR 구성 요소의 표준화된 공통개념을 이용하여 온톨로지 스키마를 정의하여 생성하는 부분

(4) (2)에서 생성된 데이터베이스의 인스턴스를 (3)에서 생성된 온톨로지에 대한 클래스의 인스턴스로 추가함으로써 온톨로지를 확장하는 부분

(1)은 MDR 구현 시스템을 구축하는 부분으로 ISO/IEC 11179 표준 지침을 보면 MDR의 메타 모델이 정의되어 있다. 메타 모델에 따라 MDR 구현 시스템의 데이터베이스 스키마를 정의한다. 메타 모델에는 객체 클래스, 프로퍼티, 데이터 요소 개념, 값 영역, 개념 영역, 데이터 요소와 관리를 위한 등록 권한, 관계를 위한 객체 클래스 간 관계 등이 정의되어 있다. 따라서 MDR 구현 시스템 역시 메타 모델을 참고하여 저장 모델을 정의하여 MDR의 등록과 관리를 한다. MDR 구현 시스템은 6개 부분으로 구성되며 다음과 같다.

- 사용자 권한 처리부분(Login Processing)
- 사용자 관리 부분(User Management)
- 권한 정보 부분(Authority Information)
- 사용자 정보 부분(User Information)
- MDR 시스템 지역 DB 관리 부분(System Local DB Management Procedure)
- MDR 관리 부분 (MDR Management Procedure)

여기에서 MDR 시스템 지역 DB 관리 부분은 각 사용자들에 대한 정보, 등록 상태인 데이터 및 데이터에 대한 인증 상태를 저장하는 DB를 관리하는 부분이라고 할 수 있다.

(2)는 MDR 구현 시스템에 등록된 표준화된 공통 개념을 이용하여 실제 데이터베이스 스키마 필드를 정의하는 부분이다. 여기에서 실제 데이터베이스 스키마 필드를 정의할 경우 MDR의 구성 요소 중에서 가장 기본이 되는 데이터 요소를 이용하여 데이터베이스 스키마 필드를 정의한다. 그 이유는 MDR의 구성 요소에는 데이터요소 이외에도 다른 표준화된 공통 개념이 있지만 데이터 요소는 데이터의 형식 즉, 값의 영역, 데이터 타입, 표현 클래스 등을 포함하기 때문이다. 이는 이 논문의 제약 사항이다.

(3)은 MDR 구현 시스템에 등록되어 있는 표준화된 공통 개념을 이용하여 온톨로지 스키마를 보다 정확하게 정의하는 부분으로 먼저 MDR 구성 요소와 온톨로지 간 사상이 이루어져야 한다. 앞서 기술하였듯이, MDR의 주요 구성 요소로는 객체 클래스, 프로퍼티, 데이터 요소 개념, 값 영역, 개념 영역, 데이터 요소, 관계가 있으며 온톨로지의 구성 요소로는 개념 즉, 클래스와 프로퍼티, 데이터 타입, 관계가 있다. MDR의 구성 요소와 온톨로지의 구성 요소 간 사상 관계는 그림 4와 같

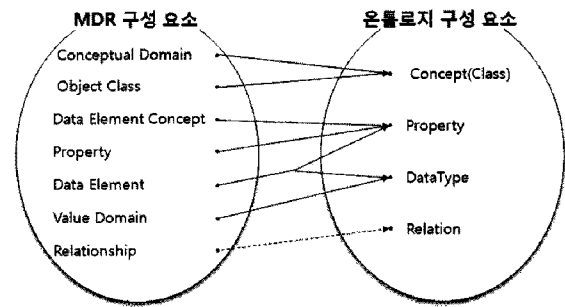


그림 4 MDR과 온톨로지 구성 요소 간 관계

다[18]. 그림 4에서 볼 수 있듯이, MDR의 개념 영역과 객체 클래스는 온톨로지의 클래스로 사상된다. 또한, MDR의 데이터 요소 개념, 프로퍼티, 데이터 요소는 온톨로지의 프로퍼티로 사상된다. MDR의 데이터 요소, 값 영역은 온톨로지의 데이터 타입으로 사상된다. 여기에서 데이터 요소는 데이터의 형식 즉, 값의 영역, 데이터 타입, 표현 클래스 등을 포함하기 때문에 프로퍼티와 데이터 타입으로 사상된다. 또한, MDR의 관계는 온톨로지의 관계로 사상된다고 정의하기는 어렵지만 의미적으로 사상될 수 있기 때문에 여기에서는 MDR의 관계와 온톨로지의 관계를 사상한다.

(4)는 실제 온톨로지의 확장이 이루어지는 부분이다. (2)에서 생성된 데이터베이스의 인스턴스를 (3)에서 생성한 온톨로지 클래스의 인스턴스로 추가하는 부분이다. 이 부분에서는 온톨로지의 스키마와 데이터베이스 스키마 필드가 같은 공통 개념인지, 즉 동일한 개념인지에 대한 규칙 기반의 학습이 이루어지며 개발자의 확인과 검증 작업이 이루어져 학습을 통하여 얻어진 결과의 정확성을 향상시킨다. 이로써 온톨로지 클래스의 인스턴스들이 확장된다.

결과적으로 제안 모델은 MDR 기반의 구조화된 데이터를 이용하여 온톨로지를 보다 정확하게 정의할 수 있으며 온톨로지를 보다 풍부하게 확장할 수 있다.

### 3.2 프로세스

그림 5는 제안 모델의 전체적인 프로세스를 보여준다. 앞서 기술하였듯이, 제안 모델은 4개 부분으로 구성되며, 전체적인 프로세스 역시 4단계로 구성된다.

1단계는 ISO/IEC 11179 표준 지침을 참고하여 MDR의 구성 요소들을 이용하여 MDR 구현 시스템을 구축하여 MDR 구현 시스템에 표준화된 공통 개념을 등록하는 단계이다.

2단계는 MDR 구현 시스템에 등록된 표준화된 공통 개념을 이용하여 실제 데이터베이스 스키마 필드를 정의하는 단계이다. 즉, 데이터베이스 스키마 필드는 MDR 구현 시스템에서 해당되는 표준화된 공통 개념을 검색하여 정의한다.

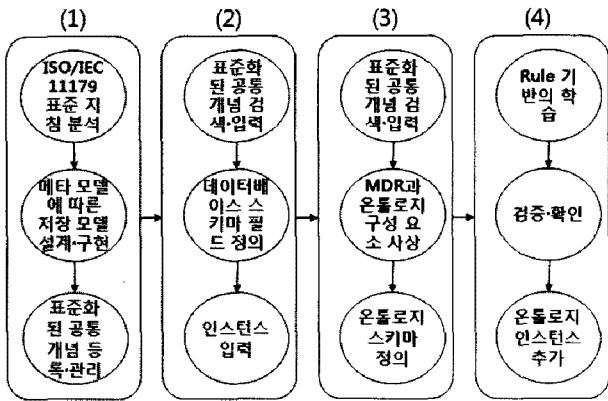


그림 5 제안 모델의 주요 프로세스

3단계는 MDR의 표준화된 공통 개념을 입력하여 MDR에 등록된 표준화된 공통 개념을 분석하고 MDR과 온톨로지 구성 요소 간의 사상을 하여 일관성 있는 즉, 보다 정확한 온톨로지를 생성하는 단계이다.

4단계는 MDR에 등록된 표준화된 공통 개념과 데이터베이스의 스키마와 데이터를 이용하여 규칙 기반의 학습을 하고 이를 통해 얻어진 결과가 맞는지에 대한 검증 및 확인 작업을 거쳐 기 생성된 온톨로지에 데이터베이스의 인스턴스를 추가하는 확장 작업을 하여 보다 풍부한 정보를 갖는 온톨로지를 생성하는 단계이다.

#### 4. 구현 및 평가

이 장에서는 제안 모델의 구현과 정성적인 평가에 대하여 기술한다.

##### 4.1 MDR 구현 시스템의 저장 모델

제안 모델을 위해서는 ISO/IEC 11179 표준 지침에 따라서 MDR를 등록 및 관리 할 수 있는 MDR 구현 시스템을 구축해야 한다. 따라서 ISO/IEC 11179 표준 지침에 따라서 MDR 구현 시스템을 구축하기 위한 저장 모델을 정의한다. ISO/IEC 11179는 재사용과 공유를 용이하게 하기 위한 메타 모델로, 실제 저장 모델은 관계형 데이터베이스에 맞게 간단하게 정의한다. 정의한 저장 모델은 그림 6과 같다.

저장 모델, S는 다음과 같은 테이블로 구성된다.

S = { Registration\_Authority, Object\_Class, Object\_Class\_Relationship, Property, Conceptual\_Domain, Conceptual\_Domain\_Relationship, Data\_Element\_Concept, Value\_Domain, Data\_Element }

Registration\_Authority는 등록된 기관에 관한 정보이며, Object\_Class, Property, Conceptual\_Domain, Data\_Element\_Concept, Value\_Domain, Data\_Element

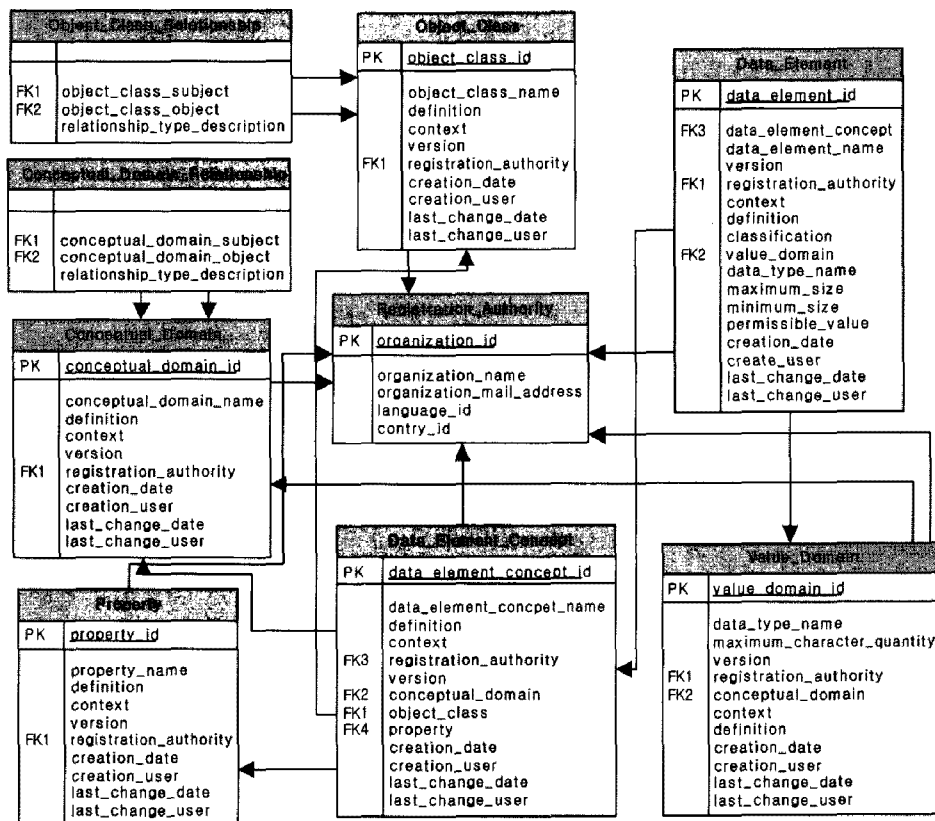


그림 6 MDR 구현 시스템의 저장 모델

는 표준화된 공통 개념이며, Object\_Class\_Relationship, Conceptual\_Domain\_Relationship은 개념 간 관계 정보이다.

그림 7은 그림 6에서의 MDR 구현 시스템의 저장 모델을 관계형 데이터베이스에 정의하고 정의된 데이터베이스의 테이블에 인스턴스를 추가하여 Data\_Element 테이블의 일부 데이터를 보여준다. 저장 모델에 저장된 인스턴스는 리하이(Lehigh) 대학의 SWAT (The Semantic Web and Agent Technologies) 연구실의 HAWK 프로젝트 연구 결과를 실험, 평가하기 위해 정의한 Univ\_bench 온톨로지를 이용한다[19]. Univ\_bench 온톨로지는 대학에 관련된 온톨로지로서 Organization, Person, Publication, Work 등의 클래스와 age, name 등의 데이터타입 프로퍼티와 degreeFrom, memberOf 등의 객체 프로퍼티 등으로 구성되어 있다.

4.2 데이터베이스 예제 생성

ISO/IEC 11179 표준 지침에 따라 MDR 구현 시스템을 구축한 후, MDR 구현 시스템에 등록된 표준화된 공통 개념을 이용하여 실제 데이터베이스 스키마 필드를 정의해야 한다. 실제 데이터베이스 스키마 필드를 정의할 때에는 표준화된 공통 개념 중에서 데이터 요소의 공통 개념만을 이용하여 데이터베이스 스키마 필드를 정의한다. 이는 데이터 요소는 데이터 타입과 데이터 형식 등을 포함하고 하고 있기 때문이다. 따라서 제안 모델의 구현을 위하여 데이터 요소의 공통 개념들을 이용하여 표 2와 같은 예제 데이터베이스 3개를 정의한다.

표 2에서 알 수 있듯이, DB1은 테이블이 데이터 요소의 공통 개념으로 모든 필드가 정의된 경우이며, DB2는 테이블이 데이터 요소의 공통 개념으로 일부 필드가 정의되고 나머지 필드들은 사용자가 정의한 필드이며,

DB3은 테이블이 데이터 요소의 공통 개념으로 필드가 정의 되어있지만, 객체 클래스의 공통 개념이 다른 필드인 것이 섞여 있다.

4.3 온톨로지 생성

MDR의 시스템에 등록된 표준화된 공통 개념을 이용하여 온톨로지 스키마를 생성하면 보다 정확한 온톨로지를 생성할 수 있다. 이를 위해서는 MDR의 구성 요소와 온톨로지 구성 요소 간 사상이 이루어져야 한다. MDR과 온톨로지 구성 요소 간 사상은 3.1절의 그림 4에서 기술한 사상 규칙에 의해서 이루어진다. 이를 통하여 생성된 온톨로지를 그래프 형태로 표현하면 그림 8과 같다.

그림 8에서, 타원은 클래스를 의미하며 ISO/IEC 11179의 개념 영역 혹은 객체 클래스에 해당한다. 개념 영역은 속성 값이나 인스턴스를 가질 수 없다. 그림에서 개념 영역에 age, name 등의 데이터 프로퍼티를 갖는 이유는 온톨

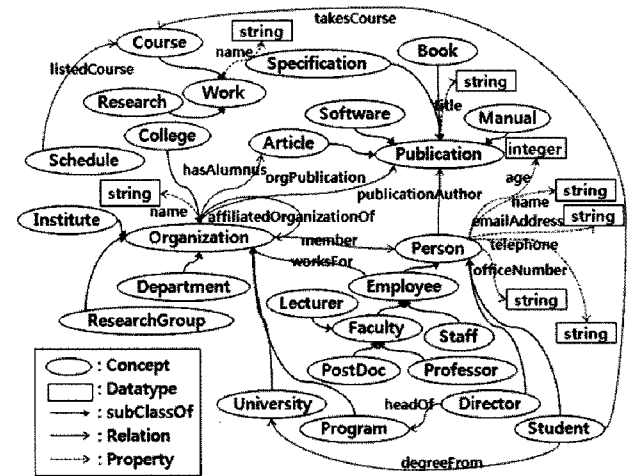


그림 8 생성된 온톨로지

DA...	DATA_ELEMENT_NAME	VERSION	REGIS...	DEFINITION	CLAS...	DATA_ELEMENT_C...	VALUE_D...	DATA_TYPE_N...
1	Professor_age_integer	1.0	1	... Professor age of the integer type	univer...	1	1	integer
2	Professor_emailAdres...	1.0	1	... Professor emailAddress of the ...	univer...	2	2	string
3	Professor_name_string	1.0	1	... Professor name of the string type	univer...	3	2	string
4	Professor_officeNumbe...	1.0	1	... Professor officeNumber of the s...	univer...	4	2	string
5	Professor_telephone_st...	1.0	1	... Professor telephone of the strin...	univer...	5	2	string
50	Professor_number_string	1.0	1	... Professor number of the string t...	univer...	50	2	string
6	PostDoc_age_integer	1.0	1	... PostDoc age of the integer type	univer...	6	1	integer
7	PostDoc_emailAddress...	1.0	1	... PostDoc emailAddress of the st...	univer...	7	2	string

그림 7 Data\_Element 테이블의 일부 인스턴스

표 2 생성된 데이터베이스 예제

DB명	설명
DB1	Data_Element 테이블의 인스턴스 중 Object_Class 필드의 값이 Student인 모든 인스턴스로 데이터베이스 스키마 필드 정의
DB2	Data_Element 테이블의 인스턴스 중 Object_Class 필드의 값이 Student인 일부 인스턴스로 데이터베이스 스키마 필드 정의
DB3	Data_Element 테이블의 인스턴스 중 Object_Class 필드의 값이 Student와 Professor인 일부 인스턴스로 데이터베이스 스키마 필드 정의

로지에서는 subClassOf 관계일 때, 속성들을 상속 받기 때문에 공통되는 속성을 상위 클래스에 표현하였기 때문이다. 또한, 객체 클래스에 해당하는 클래스는 확장될 인스턴스를 저장하기 위하여 별도의 테이블을 생성해야 한다. 이는 확장될 인스턴스가 데이터베이스의 인스턴스 이므로 데이터 사이의 관계성을 유지시키기 위함이다.

이 논문에서는 생성된 온톨로지는 대부분의 온톨로지가 현재 저장되고 있는 관계형 데이터베이스에 저장한다. 이를 위하여 표준 온톨로지 기술 언어인 OWL로 기술하였다. OWL로 기술된 온톨로지는 온톨로지 저장소이자 추론엔진인 Jena를 통하여 현재 온톨로지가 대부분 저장되고 있는 관계형 데이터베이스에 저장된다. 관계형 데이터베이스에 저장된 온톨로지는 트리플 구조이며, SUBJECT, PREDICATE, OBJECT 세 개의 필드를 가진다. 그림 9는 온톨로지가 저장된 테이블의 일부 인스턴스를 보여준다.

4.4 온톨로지 확장을 위한 사용자 인터페이스

제안 모델은 표준화된 공통 개념을 이용하여 보다 정확한 온톨로지를 생성하고 온톨로지 클래스에 대한 인스턴스를 보다 풍부하게 할 수 있는 온톨로지 확장을 위한 것이다. 앞서 ISO/IEC 11179 표준 지침에 따라 MDR 구현 시스템을 구축하고 등록된 표준화된 공통 개념을 이용하여 데이터베이스의 스키마 필드를 정의하고 온톨로지를 생성하였다. 생성된 온톨로지 클래스에 대한 인스턴스를 풍부하게 확장하기 위하여 데이터베이스의 인스턴스를 추가하게 된다. 이를 위하여 온톨로지 확장을 위한 사용자 인터페이스를 구현하였다(그림 10).

그림 10-(1)은 인스턴스를 추가할 클래스와 해당 프로퍼티를 선택하는 부분이다. 클래스를 선택하면 해당 프로퍼티를 선택할 수 있다. 그림 10-(2)는 확장할 인스턴스들을 추출할 임의의 데이터베이스에 접속하기 위하여 데이터베이스 정보를 입력한 후 리스트에 데이터베이스의 테이블 정보가 디스플레이 되면 테이블을 선택한다. 선택된 테이블의 정보는 테이블에 출력된다. 그림 10-(3)은 그림 10-(1)에서 선택한 클래스와 프로퍼티와 매핑 시킬 수 있는 그림 10-(2)의 선택된 테이블의 필

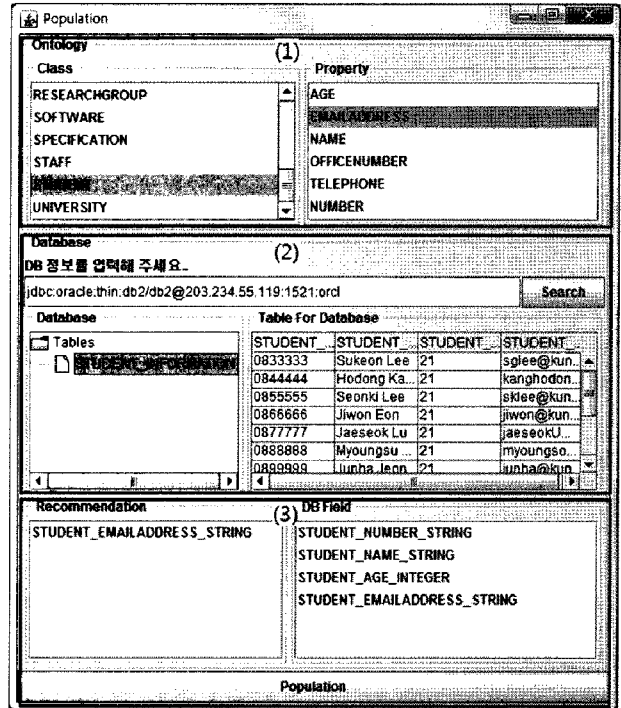


그림 10 온톨로지 확장을 위한 사용자 인터페이스 화면

드를 추천하여 디스플레이 되면 확장 버튼을 클릭하여 데이터베이스의 인스턴스를 온톨로지의 인스턴스로 확장을 하는 부분이다. 만약 추천된 필드가 없다면 직접 해당하는 필드를 선택하여 데이터베이스의 인스턴스를 온톨로지의 인스턴스로 추가할 수 있다.

온톨로지를 확장하기 위한 절차는 다음과 같다.

- Step1. 그림 10-(1)에서 해당되는 클래스와 프로퍼티를 선택
- Step2. 그림 10-(2)에서 실제 데이터베이스의 정보를 입력하고 데이터베이스의 테이블을 선택
- Step3. 그림 10-(3)에서 추천된 필드를 선택 또는 확인 작업 후 직접 필드를 선택하여 확장 버튼 클릭

Step1 → Step2 → Step3을 거치면 데이터베이스의 인스턴스가 온톨로지 클래스의 인스턴스로 추가된다. 여기에서 실제 임의의 데이터베이스의 인스턴스는 온톨로

SUBJ	PROP	OBJ
Uv::http://ist.kunsan.ac.kr/unive-bench.owl#Lecturer	Uv::http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv::http://www.w3.org/200
Uv::http://ist.kunsan.ac.kr/unive-bench.owl#Lecturer	Uv::http://www.w3.org/2000/01/rdf-schema#subClassOf	Uv::http://ist.kunsan.ac.kr/
Uv::http://ist.kunsan.ac.kr/unive-bench.owl#Lecturer	Uv::http://www.w3.org/2000/01/rdf-schema#label	Lv:0:39:http://www.w3.org,
Uv::http://ist.kunsan.ac.kr/unive-bench.owl#listedCourse	Uv::http://www.w3.org/2000/01/rdf-schema#domain	Uv::http://ist.kunsan.ac.kr/
Uv::http://ist.kunsan.ac.kr/unive-bench.owl#listedCourse	Uv::http://www.w3.org/2000/01/rdf-schema#range	Uv::http://ist.kunsan.ac.kr/
Uv::http://ist.kunsan.ac.kr/unive-bench.owl#listedCourse	Uv::http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv::http://www.w3.org/200
Uv::http://ist.kunsan.ac.kr/unive-bench.owl#listedCourse	Uv::http://www.w3.org/2000/01/rdf-schema#label	Lv:0:39:http://www.w3.org,
Uv::http://ist.kunsan.ac.kr/unive-bench.owl#Department	Uv::http://www.w3.org/1999/02/22-rdf-syntax-ns#type	Uv::http://www.w3.org/200
Uv::http://ist.kunsan.ac.kr/unive-bench.owl#Department	Uv::http://www.w3.org/2000/01/rdf-schema#label	Lv:0:39:http://www.w3.org,

그림 9 저장된 온톨로지 일부 인스턴스

지에 추가 할 때 URI 정보가 있어야 한다. 그러므로 인스턴스의 URI 정보를 'HTTP://DB의 IP 정보/DB 사용자명/테이블명'의 형태로 기술한다. 이는 확장되는 온톨로지의 인스턴스들은 데이터베이스의 인스턴스를 이용하여 확장되었기 때문이다.

#### 4.5 평가

이 논문에서는 MDR의 장점과 특징을 이용하여 온톨로지 인스턴스를 확장할 수 있는 모델을 제안하였다. 현재 MDR에서 관리하는 표준화된 공통 개념을 이용한 온톨로지를 스키마를 정의하기 위한 연구가 진행되고 있다. 이러한 연구 결과는 온톨로지 스키마를 구성하는 클래스 및 프로퍼티에 대한 이해를 용이하게 하고 온톨로지 간 상호운용성을 향상시킨다. 그러나 보다 중요한 사항은 온톨로지 인스턴스를 추가하고 확장하는 방법에 대한 연구이며, 따라서 MDR 기반의 온톨로지 스키마 명세 방법을 보완하여 인스턴스를 확장할 수 있는 모델에 대한 연구가 요구된다. 이 논문에서는 이와 같은 배경에서 MDR 기반의 온톨로지 인스턴스 확장 모델을 제안하였으므로 이러한 측면을 고려하여 제안 모델에 대한 평가를 수행한다.

##### 4.5.1 평가 대상 및 평가 항목 정의

온톨로지 스키마가 정의되면 인스턴스를 추가하는 방법은 크게 수동으로 새로운 인스턴스를 직접 추가하는 방법과 이미 존재하는 정보를 인스턴스로 추가하는 방법으로 분류된다. 새로운 인스턴스를 직접 추가하는 방법은 많은 노력과 시간을 필요로 하며 이미 존재하는 정보도 그 양이 매우 풍부하기 때문에 제안 모델은 후자의 방법을 택한다.

기 생성된 정보를 이용하여 온톨로지를 확장하는 일반적인 방법은 웹상에 존재하는 자원으로부터 인스턴스를 획득하여 취득하는 방법이다. 그러나 이 논문에서의 제안 모델은 데이터베이스로부터 인스턴스를 획득하여 추가하는 접근 방법이다. 즉, MDR과 데이터베이스 그리고 MDR을 기반으로 생성된 온톨로지 스키마 간의 관계성을 이용하여 온톨로지를 확장하는 방법으로써, 일반적인 온톨로지 확장 모델이 고려하는 환경에 비해 특수한 환경을 위한 모델이라 할 수 있다.

이와 같은 환경을 고려할 경우, 웹 자원을 이용한 일반적인 온톨로지 확장 모델과의 직접적인 비교 평가는 부적절해 보이며, 가장 타당한 비교 평가 대상은 제안 모델에서 고려하는 환경 하에서 인스턴스를 확장하기 위해 제안된 모델이라 할 수 있다. 그러나 현재까지 제안된 확장 모델이 부재하여 비교 대상 선정이 용이하지 않으며, 아울러 MDR을 기반으로 정의된 온톨로지 스키마의 인스턴스를 웹 자원을 이용하여 추가 및 확장할 수 있다. 따라서 MDR을 기반으로 정의된 온톨로지 스

키마에 인스턴스를 추가한다는 가정 하에서, 제안 모델과 일반적인 웹 자원 기반의 온톨로지 확장 모델을 비교 평가한다.

온톨로지 인스턴스 확장에서 가장 중요한 요소는 정확한 인스턴스를 추출하여 추가하는 부분이다. 이를 위해서는 검색된 후보 인스턴스를 대상으로 확인 작업을 수행하게 된다. 이러한 확인 작업에 대한 자동화를 위한 연구가 진행되어 왔지만 현재까지 수동 혹은 반자동으로 확인 작업을 수행한다. 무엇보다도 정확한 인스턴스를 획득하기 위해서는 온톨로지 개발자에 의한 수동 확인 작업이 필수적이다. 이 작업은 많은 비용, 즉 시간이 요구되는 작업으로 비용을 줄이기 위해서는 가능한 정확한 후보 인스턴스만을 검색해서 반환해 주어야 한다. 이 논문에서는 평가 항목과 그 의미를 정리하면 다음과 같다.

- 처리 성능 : 온톨로지를 구축하기 위해서는 많은 정보가 필요하며 그 정보의 신뢰성, 즉 정확성을 보장해야 한다. 이를 위해 온톨로지 개발자의 확인 작업에 매우 많은 비용이 소요된다. 확인 작업에 가장 결정적인 영향을 미치는 요인은 후보 인스턴스의 개수이며 이를 통해 온톨로지 개발자의 확인 작업에 소요되는 시간을 산출할 수 있다. 확인 작업에 관련하는 후보 인스턴스의 수에 대한 비교 연산을 통해 두 모델의 전체적인 성능을 비교할 수 있다. 이 논문에서는 제안 모델과 일반적인 웹 자원을 이용한 확장 모델에서 결정적인 영향을 미치는 주요 처리 연산들을 중심으로 처리 성능을 평가한다. 또한 처리 성능을 보다 세부적으로 평가하기 위해 확장을 위해 관여하는 (1)인스턴스 개수와 이를 토대로 정의한 전체적인 (2)처리 시간에 대한 비교 평가를 수행한다.

비교 평가를 수행하기 위해서는 각 모델의 전체적인 처리 알고리즘에 대한 정의가 요구된다. 제안 모델과 일반적인 확장 모델의 알고리즘을 의사 코드로 기술하면 표 3과 같다. 두 모델의 전체적인 처리 알고리즘은 기본적으로 유사한 패턴을 지닌다. 주어진 키워드를 이용하여 후보 인스턴스를 검색한 후, 각 후보 인스턴스를 확인하여 유효한 후보만을 인스턴스로 선택하여 추가한다. 그러나 두 모델 간에는 기본적인 접근 방법에서 오는 차이점이 존재한다. 웹 기반의 온톨로지 확장 모델은 실제 값 레벨에서 확인 작업이 이루어지는 반면 제안 모델은 속성 레벨에서 이루어진다. 다시 말해서 일반적인 확장 모델의 경우에는 웹상의 자원에 대한 키워드 검색을 수행하여 후보 인스턴스를 포함하는 문서를 반환하게 되고(retrieveInfo(k) 함수), 온톨로지 공학자가 해당 사이트를 방문(candidates.getLink() 함수 및 visit(c) 함수)하여 유효한 인스턴스인지를 판단하게 된다. 그러나 제안 모델은 MDR의 데이터 요소를 이용하여 정의



표 3 모델에 따른 처리 알고리즘

제안 모델의 처리 알고리즘	일반적인 확장 모델의 처리 알고리즘
<pre> K : INPUT candidates, selectedInstances : LIST; c : STRING; BEGIN_PROPOSED   candidates = getSchemaInfo(K<sub>i</sub>);   WHILE(candidates.next()!=NULL)     c = candidates.getMetadata();     IF (select(c) == TRUE))       selectedInstances.add(c);   END_IF END_WHILE populate(selectedInstances); END_PROPOSED                     </pre>	<pre> K : INPUT candidates, selectedInstances : LIST; c : STRING; BEGIN_PROPOSED   candidates = retrieveInfo(K<sub>i</sub>);   WHILE(candidates.next()!=NULL)     c = candidates.getLink();     visit(c);     IF (select(c) == TRUE))       selectedInstances.add(c);   END_IF END_WHILE populate(selectedInstances); END_PROPOSED                     </pre>

한 데이터베이스 테이블로부터 후보 인스턴스를 추출하게 되며, 따라서 실제 인스턴스를 직접 후보 인스턴스로 검색하지 않고 이를 대표하는 데이터 요소, 즉 데이터베이스 속성명을 검색하고(getSchemaInfo(k) 함수) 이에 대한 확인 작업(candidates.getMetadata())을 통해 인스턴스를 결정하게 된다.

4.5.2 처리 성능 평가

이 논문에서의 평가 항목은 처리 성능 평가이며 이를 위해 두 개의 세부 평가 항목을 정의한다. 첫 번째는 확장을 위해 이용되는 인스턴스 개수에 대한 평가이며 두 번째는 첫 번째 평가 항목의 평가 모델을 기반으로 한 처리 시간에 대한 평가이다.

성능 평가를 위해 데이터베이스와 웹상에 동일한 인스턴스가 분포하고 후보 인스턴스 검색 시에 원하는 모든 유효 인스턴스들이 반환된다고 가정한다. 이와 같은 가정을 전제로, 두 모델을 통해 주요 프로세스별 처리되는 인스턴스 수를 평가하기 위한 계산 모델은 다음과 같이 정의된다.

먼저, 제안 모델을 위한 계산 모델, CMI(Proposed)는 다음과 같다.

$$CMI(Proposed) = Cal(getSchemaInfo(K_i)) + Cal(candidates.getMetadata())$$

이 때, Cal(getSchemaInfo(k))의 결과가 최악, 즉 모든 정보를 검색한다고 가정하면,

$$CMI(Proposed) = n(F_A) + n(F_A) = 2*n(F_A)$$

가 된다. 실제 검색되어야 하는 검색 결과(F<sub>S</sub>)를 포함하여 전체 필드(F<sub>A</sub>) 개수가 검색됨을 의미하며, 따라서 최악의 경우 검색되는 경우의 개수는 n(F<sub>A</sub>)가 된다. 값 레벨에서의 검색이 아닌 스키마 레벨에서의 검색이 수행되므로 전체 값이 아닌 필드 정보가 이용된다. 따라서 분포되어 있는 인스턴스의 총 개수를 I<sub>A</sub>이라고 할 때, 제안 모델의 가능한 후보의 총 개수는 I<sub>A</sub>가 아닌 F<sub>A</sub>가 된다.

다음은 일반적인 웹 기반 확장 모델을 위한 계산 모델, CMI(General)은 다음과 같다.

$$CMI(General) = Cal(retrieveInfo(K_i)) + Cal(candidates.getLink())$$

이 때, Cal(retrieveInfo(K<sub>i</sub>))의 결과가 최적, 즉 후보 인스턴스가 검색하고자 하는 정확한 인스턴스 집합이라고 가정하면,

$$CMI(General) = n(I_S) + n(I_S) = 2*n(I_S)$$

가 되며, 이 때 n(I<sub>S</sub>)는 사용자가 추가하고자 하는 인스턴스의 개수를 의미한다. 만일 후보 인스턴스 개수가 최악이라면 계산 모델은 다음과 같다. 이 때, n(I<sub>A</sub>)는 분포된 전체 인스턴스의 개수로서 후보 개수의 최대 개수를 의미한다.

$$CMI(General) = n(I_A) + n(I_A) = 2*n(I_A)$$

앞서 기술한 내용, 즉 정의한 각 계산 모델을 기초로 주요 연산과 조건에 따른 인스턴스의 개수 등을 정리하면 표 4와 같다.

웹 기반 온톨로지 확장 모델의 경우, n(I<sub>S</sub>)개의 인스턴스를 추가하기 위해 검색된 후보의 개수는 n(I<sub>S</sub>)+a이다. 여기서 a는 0 이상으로 후보 인스턴스가 정확하게 검색되었다고 가정할 때(최상의 정확도), 즉 a=0이므로 확인 연산을 위해 전달되는 후보 인스턴스의 개수는 n(I<sub>S</sub>)이다.

값 레벨에서 후보 인스턴스 검색 및 확인 연산이 이루어지는 일반적인 온톨로지 확장 모델과는 달리, 제안 모델은 스키마 레벨, 즉 메타데이터(데이터베이스의 속성명)를 이용하여 실제 추가될 인스턴스를 판단하여 추출하게 된다. 따라서 검색 대상 후보의 총 개수는 n(I<sub>A</sub>)가 아닌 n(F<sub>A</sub>)가 된다. 따라서 사용자가 요구하는 인스턴스를 획득하기 위해 검색되는 후보 개수는 n(F<sub>S</sub>)이며, 최악의 경우 n(F<sub>A</sub>)를 사용자가 확인하게 된다.

주요 요인에 대한 인자 값에 대한 정의를 요약하면

표 4 주요 연산과 조건에 따른 인스턴스의 수

주요 연산 & 조건	웹 기반 확장 모델	제안 모델
전체 인스턴스 개수	$n(I_A)$	$n(I_A)$
검색 후보 개수	$n(I_S) + a$ ( $0 \leq a \leq n(I_A - I_S)$ )	$n(F_S) + \beta$ ( $0 \leq \beta \leq n(F_A - F_S)$ )
검색 성능(정확도) 조건	최상의 정확도 조건 부여 : $a=0$ 인 경우	최악의 정확도 조건 부여 : $\beta=n(F_A)-n(F_S)$ 인 경우
확인할 후보 개수	최상의 정확도 부여하면, $n(I_S) + a$ $= n(I_S) + 0 = n(I_S)$	최악의 정확도를 부여하면, $n(F_S) + \beta$ $= n(F_S) + (n(F_A) - n(F_S))$ $= n(F_A)$

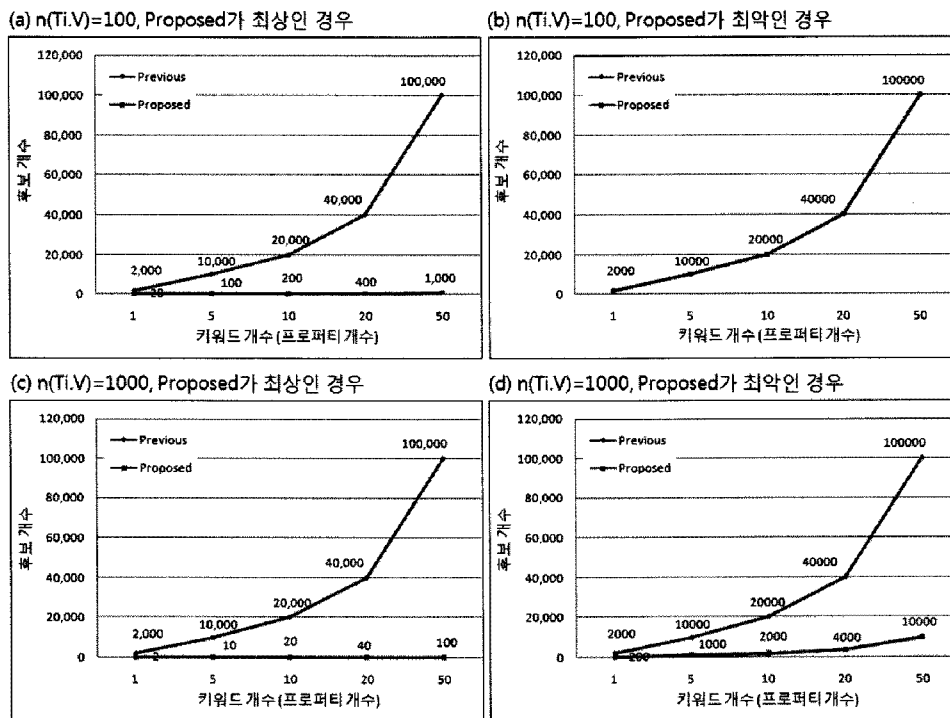


그림 11 이용되는 인스턴스 수 비교 평가 결과

다음과 같으며, 그림 11은 위의 정의를 바탕으로 얻어진 결과를 보여준다.

- $n(I_A)$  : 모든 인스턴스 개수이며 고정 값을 이용한다.
  - $n(K)$  : 인스턴스를 추가하기 위해 주어진 키워드의 입력 회수로서, 인자 값의 집합은 {1, 5, 10, 20, 50}이다. 즉  $n(K)$ 는 이 값들 중 하나임
  - $n(T_i.V)$  : 각 속성에 대한 값의 개수로서, 특정 속성 값의 집합  $T \subset I_A$ 이며 따라서  $n(T_i.V) < n(I_A)$ 이다. 이 논문에서는 모든 속성이 지니는 값의 개수가 동일하게 분포한다고 가정하며,  $n(T_i.V)$ 의 값을 {100, 1000}으로 설정
  - $n(F)$  : 속성의 개수로서  $n(F) = n(I_A) / n(T_i.V)$ 로 정의
  - $n(I_S)$  : 실제 사용자가 원하는 인스턴스  $I_S$ 의 개수를 의미하며  $n(I_S) = 1000$
- 후보군 검색 및 확인 작업을 통해 온톨로지 인스턴스

를 추가하는데 소요되는 처리 시간 평가를 위한 계산 모델을 정의한다. 처리 시간 계산 모델은 처리되는 인스턴스 수를 위한 계산 모델을 기반으로 후보 인스턴스 획득 연산을 추가하여 정의한다. 다른 연산의 경우, 그 영향이 미비하기 때문에 계산 모델에서 제외한다. 계산 모델에서, 중요한 단위 연산은 단일 후보를 검색하는 연산, 각 후보를 사용자가 확인하는 연산이다. 각 연산에 걸리는 시간  $t$ 는  $0 < t < 1$  범위 내로 제한한다.

먼저, 제안 모델을 위한 계산 모델, CMT(Proposed)는 다음과 같이 정의된다. 이 때,  $n(T_i.V)$ 의 값에 따라  $Cal(getSchemaInfo(K_i))$ 의 결과를 최악과 최상의 경우로 구분하여 평가한다. 예를 들어,  $n(T_i.V)$ 의 값을 1000으로 설정하고 최악의 경우를 고려할 때,  $Cal(getSchemaInfo(K_i))$  값은  $n(F_A)$ 이다. 그러나 최상의 경우를 고려할 때의 값은  $n(F_S)$ 가 된다. `candidates.getMetadata()`

연산과 select(c) 연산에 대해서 최악의 경우에는  $n(F_A)$  만큼 수행되고 최상의 경우에는  $n(F_S)$ 만큼 반복 연산을 수행한다.

$$CMT(Proposed) = T.getS + \sum_{i=1}^{n(F_A)} (T.getM + T.selectI)$$

- T.getS : getSchemaInfo(K<sub>i</sub>) 연산을 위해 소요되는 시간
- T.getM : candidates.getMetadata() 연산 처리 시간
- T.selectI : select(c) 연산 처리 시간

다음은 일반적인 웹 기반 확장 모델을 위한 계산 모델, CMT(General)은 다음과 같다. 이 때, Cal(retrieve-Info(K<sub>i</sub>))의 결과는 최상, 즉 후보 인스턴스가 검색하고자 하는 정확한 인스턴스 집합인 경우만을 고려하여 평가를 수행한다. 따라서 값이  $n(I_S)$ 이므로 WHILE 문 내의 연산들은  $n(I_S)$ 만큼 반복된다.

$$CMI(General) = T.getInfo + \sum_{i=1}^{n(I_S)} (T.getL + T.visit + T.selectI)$$

- T.getInfo : retrieveInfo(K<sub>i</sub>) 연산을 위해 소요되는 시간
- T.getL : candidates.getLink() 연산 처리 시간
- T.visit : visit(c) 연산 처리 소요 시간
- T.selectI : selected(c) 연산 처리 시간

위 계산 모델과 앞서 정의한 인자 값 설정을 바탕으로 얻어진 결과는 그림 12와 같다.

### 4.5.3 평가 결과 분석

그림 11의 평가 결과를 보면, 대부분의 경우에 제안 모델의 성능이 우수함을 알 수 있다. 앞서 언급하였듯이, 일반적인 확장 모델은 최상의 경우만을 고려하였음에도 제안 모델의 최악의 경우보다 성능이 저하됨을 알 수 있다. 그림 11(b)의 결과는 두 모델이 동일한 성능을 보인다. 이 결과는  $n(T_i.V)=100$ 으로 설정했을 때, 제안 모델의 최악의 경우와 일반적인 확장 모델의 최상의 경우를 비교한 결과이다. 비록 두 결과가 동일하지만, 실질적으로 최악의 경우는 확률적으로 일어나기 드물기 때문에 전반적으로 제안 모델이 더 높은 성능을 보인다는 결론을 도출해 낼 수 있다.

그림 12의 처리 시간에 대한 성능 평가 결과에서, 전체적으로 제안 모델이 나은 성능을 보였다. 특히 그림 12(b)와 그림 12(d)는 제안 모델은 최악의 경우, 그리고 일반적인 모델은 최상의 경우를 고려한 결과이다. 그럼에도 제안 모델이 나은 성능을 보임을 알 수 있다. 또한 하나의 인스턴스 확인 시간,  $t$ 를  $0 < t < 1$  사이의 난수를 이용하여 두 모델 간 확인 작업에 걸리는 단위 시간을 거의 동일하다는 조건 하에서 얻어진 결과로서, 실질적으로 제안 모델의 경우 데이터베이스에 인스턴스들이 한 필드에 있는 값의 성격이 같기 때문에 이를 확인하는 시간은 웹 문서에 흩어져 있는 값을 확인하는 시간보다 더 적게 소요된다. 이는 제안 모델과 일반적인 확장 모델 간 처리 성능 차이가 더욱 커질 수 있음을 의미한다.

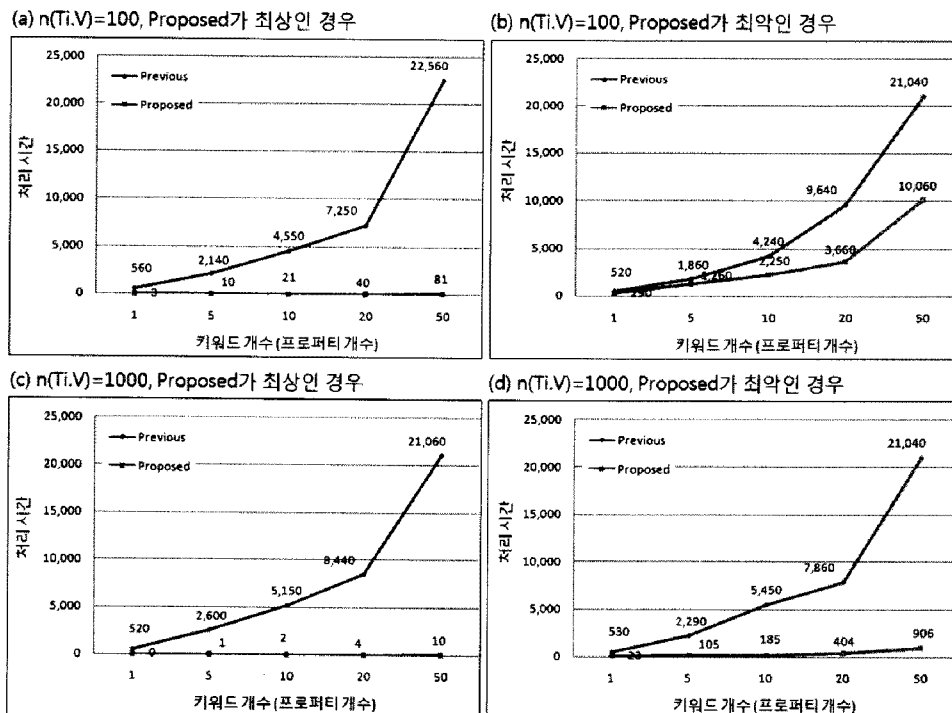


그림 12 처리 시간 비교 평가 결과

이러한 평가 결과를 바탕으로 이 논문에서 제안한 MDR 기반의 온톨로지 확장 모델과 프로토타입은 다음과 같은 장점 및 활용성을 지닌다.

- 사용자 인터페이스를 제공하여 온톨로지 인스턴스 구축용이
- 표준화된 공통 개념을 이용함으로써 온톨로지의 신뢰성 보장
- 확인 작업을 통한 인스턴스의 정확성 보장
- 온톨로지 인스턴스 확장을 위한 비용 절감
- 풍부한 온톨로지 구축을 위한 모델로서 활용

그러나 제안 모델 또한 다음과 같은 몇 가지 제약사항 및 한계점을 지니며, 향후 이러한 문제들이 해결될 수 있도록 제안 모델의 확장이 요구된다.

- 제안 모델은 MDR과 데이터베이스 스키마 간 관계 정보 구축 필요
- MDR 기반 온톨로지 정의 시 발생 가능한 다양한 경우를 고려한 모델 확장 필요
- 일반적인 확장 모델과의 연계를 통한 온톨로지 확장이 가능한 혼합 모델 개발 필요

## 5. 결론 및 향후 연구

이 논문에서는 구축된 온톨로지를 응용 시스템에 활용하기 위하여 온톨로지 클래스의 인스턴스를 풍부하게 확장하기 위한 모델을 제안하였으며 프로토타입을 구현하여 실제 온톨로지를 확장하는 방법을 보이고 정량적인 평가를 수행하였다. 제안 모델은 사용자 인터페이스를 제공하여 온톨로지 인스턴스 구축을 용이하게 하며 확인 작업을 통하여 인스턴스의 정확성을 보장할 수 있다. 또한 온톨로지 인스턴스 확장을 위한 비용을 절감할 수 있으며 풍부한 온톨로지 구축을 위한 모델로서 활용할 수 있다.

제안 모델은 MDR을 기반으로 온톨로지를 정의할 때, 객체 클래스를 온톨로지 개념(클래스)으로 사상시키는 경우만 고려하였다. 그러나 향후에는 MDR을 기반으로 온톨로지를 정의할 때 발생하는 다양한 경우를 고려하여 제안 모델을 확장할 필요가 있다. 또한 제안 모델과 일반적인 확장 모델은 상호 보완적으로 활용되어야 한다. 따라서 일반적인 확장 모델과의 연계를 통한 온톨로지 확장이 가능한 혼합 모델 개발이 이루어져 한다.

## 참고 문헌

- [1] J. Hendler, "Web 3.0 Emerging," *Journal of Computer*, vol.42, no.1, pp.111-113, Jan. 2009.
- [2] T. Berners-Lee, "The World Wide Web: Past, Present and Future," *IEEE Computer Magazine*, vol.29, no.10, pp.69-77, Oct. 1996.
- [3] A. Gomez-Perez and O. Corcho, "Ontology Languages for the Semantic Web," *Journal of IEEE Computer Society*, vol.17, no.1, pp.54-60, Feb. 2002.
- [4] World Wide Web Consortium, <http://www.w3.org/>, 2009.
- [5] W3C, Resource Description Framework, <http://www.w3.org/RDF/>, 2009.
- [6] W3C, Web Ontology Language, <http://www.w3.org/2004/OWL/>, 2009.
- [7] Jena Semantic Web Framework, <http://jena.sourceforge.net/>, 2009.
- [8] The Protege Ontology Editor and Knowledge Acquisition System, <http://protege.stanford.edu/>, 2009.
- [9] S.G. Oh, "Designing a Metadata Registry Using Semantic Web Technology," *Journal of LISS*, vol.36, no.3, pp.109-136, Sep. 2005. (in Korean)
- [10] D.-K. Baik, "MDR:MetaData Registry," *Journal of TTA*, vol.2000, no.71, Oct. 2000. (in Korean)
- [11] METeOR, Australian Institute of Health and Welfare, <http://meteor.aihw.gov.au/>, 2009.
- [12] ICIS CIHI, Canadian Institute for Health Information, <http://secure.cihi.ca/ddexternal/welcome.do>, 2009.
- [13] NCICB: Cancer Data Standards Registry and Repository(caDSR), US National Cancer Institute, [http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore\\_overview/cadsr](http://ncicb.nci.nih.gov/NCICB/infrastructure/cacore_overview/cadsr), 2009.
- [14] Data Registry Services, US Environmental Protection Agency, <http://www.epa.gov/edr/>, 2009.
- [15] United States Healthcare Information Knowledge-base, US Health organizations, <http://ushik.ahrq.gov>, 2009.
- [16] National Information Exchange Model, U.S. Department of Homeland Security (DHS) and U.S. Department of Justice (DOJ), <http://www.niem.gov/>, 2009.
- [17] Global Justice XML, US Department of Justice, [http://www.it.ojp.gov/topic.jsp?topic\\_id=43](http://www.it.ojp.gov/topic.jsp?topic_id=43), 2009.
- [18] D. Jeong, J.-D. Kim, J. Son, J. Kim, and D.-K. Baik, "A Study on Creation of Web Ontology based on the Metadata Registry for the Semantic Web," *Proc. of the 2008 KSCI Winter Conference*, vol.16, no.2, pp.19-24, Jan. 2009. (in Korean)
- [19] Univ-bench Ontology, Lehigh University, <http://www.lehigh.edu/~zhp2/2004/0401/univ-bench.owl>, 2009.
- [20] H. Jeong, D.-K. Baik, and D. Jeong, "Web Ontology Learning and Population Model using Structured Data Based on MDR," *Proc. of the 2009 KITS Spring Conference*, pp.393-396, May. 2009. (in Korean)



## 정혜진

2008년 군산대학교 수확정보통계학부 정보통계학전공(이학사). 2008년~현재 군산대학교 정보통계학과 석사과정. 관심분야는 시맨틱 웹, 데이터베이스, 모델링, 정보보안

## 백두권

정보과학회논문지 : 데이터베이스  
제 36 권 제 2 호 참조



## 정동원

1997년 군산대학교 컴퓨터학과 이학사  
1999년 충북대학교 전산학과 이학석사  
2004년 고려대학교 컴퓨터학과 이학박사  
1998년 전자통신연구원 위촉연구원. 1999년~2000년 ICU 부설 한국정보통신교육원 GIS 분원 전임강사. 2000년~2001년 (주)지구넷 부설 연구소 선임연구원. 2002년~2005년 라임미디어 테크놀로지 부설 연구소 연구원. 2004년~2005년 고려대학교 정보통신기술연구소 연구조교수. 2005년 Pennsylvania State University PostDoc. 2002년~2004년 TTA 표준화위원회-데이터연구회(SG08.02) 특별위원. 2004년~현재 TTA 표준화위원회-메타데이터표준화 프로젝트 그룹(PG406) 위원. 2005년~현재 군산대학교 정보통계학과 교수. 2006년~현재 데이터관리서비스 전문위원회 (ISO/IEC JTC1/SC32 국내위원회) 위원. 2008년~현재 지리정보 전문위원회(ISO/TC211 국내위원회) 위원. 2009년~현재 TTA 표준화위원회-NGIS 프로젝트그룹(PG409). 관심분야는 데이터베이스, 시맨틱 웹, 시맨틱 GIS, 유비쿼터스 컴퓨팅, 정보보안