

# 시멘틱 웹 기반 개방형 전자도서관 모델에 관한 연구

## A study of Semantic Web based Open Digital Library Model

황상규\*, 오경묵\*\*

(Hwang, Sang-Kyu; Oh, Kyung-Mook)

### 목 차

- |                          |                           |
|--------------------------|---------------------------|
| 1. 시멘틱 웹 환경하에 전자도서관 구축   | 3. 응용프로파일 메타데이터 구조        |
| 2. 메타데이터 기반 시멘틱 웹 구축     | 3.1. 시멘틱 웹에서 메타데이터의 활용    |
| 2.1. 의미정보교환을 위한 시멘틱 웹 표준 | 3.2. 개방형 전자도서관 모델         |
| 2.2. 시멘틱 웹에서 XML의 역할     | 3.3. 응용프로파일 메타데이터 자동변환    |
| 2.3. 시멘틱 웹에서 RDF의 필요성    | 3.4. 응용프로파일 자동변환 알고리즘의 활용 |
| 2.4. 시멘틱 웹에서 웹 온톨로지의 필요성 | 4. 결론 및 향후발전방향            |

### 초 록

최근에 이르러 차세대 웹 아키텍처인 시멘틱 웹에 관한 연구와 투자에 대한 관심이 계속해서 증대되고 있다. 정보학적인 관점에서, 차세대 웹 아키텍처인 시멘틱 웹은 하나의 거대한 메타데이터 조직으로 볼 수 있다. 시멘틱 웹을 거대한 메타데이터 조직으로 볼 수 있는 가장 큰 이유는, 시멘틱 웹을 구축과정에서 가장 중요한 단계 중 하나가 웹 정보자원에 대한 정형화된 메타데이터를 작성하는 것이기 때문이며, 이용자는 메타데이터를 이용하여 보다 쉽게 자신이 원하는 정보를 찾을 수 있다. 본 논문에서는, 시멘틱 웹 환경 하에서 서로 다른 정보체계구조를 지닌 개방형 전자도서관의 상호운영성을 제공하기 위하여 새로운 방식의 응용프로파일 메타데이터구조를 개발하였다. 새로운 방식의 응용프로파일 메타데이터구조를 토대로, 개방형도서관모델에서 서로 다른 형태의 대규모 메타데이터를 통합하기 위한 공동서지목록 자동생성 및 통합검색 알고리즘을 개발하였다.

### Abstract

Recently there has been a growing interest in the investigation and development of the next generation web - the Semantic Web. From the perspective of a information science, the next generation web - Semantic Web is a metadata initiative. It is reason that One of important stage of Semantic Web Construction is adding formal metadata that describes a Web resource's content and so people can find easy material using metadata. In this paper, We designed new application profile metadata architecture as a way to serve as interoperability between various open digital libraries using different information architecture in Semantic Web environment. Based on new application profile metadata architecture, we developed union catalog automatic generation and union search algorithm to integrate heterogeneous huge-scale metadata in open digital library.

키워드: Open Digital Library, Semantic Web, Application Profile, Metadata

\* 한국국방연구원(Korea Institute for Defense Analyses)

\*\*숙명여자대학교 정보과학부(Division of Information Science, Sookmyung Women's Univ)

- 논문집수일 : 2004. 2. 26.
- 게재확정일 : 2004. 3. 10.

## 1. 시멘틱 웹 환경하에 전자도서관구축

지금까지의 웹 정보체계는 주로 사람과 컴퓨터간의 정보교환을 위한 것이고, 이를 위한 도구인 웹 마크업 언어(HTML)는 눈으로 보여지는 부분에 대한 표현기술에 초점을 맞추고 있다. 웹 마크업 언어는 웹 콘텐츠 자체의 의미전달을 위한 구체적인 해결방안을 제시하지 못하며, 웹 마크업 언어의 표현방식으로는 문서의 내용과 의미를 나타내는 의미정보(semantic information)를 제대로 표현하기가 힘들다. 웹 프로그래머는 웹 마크업 언어의 태그(tag)라는 명령어를 통해 글자의 크기, 색깔 등을 지정할 수 있어도, 해당문서의 주제는 무엇이고, 언제 누구에 의해 쓰여졌는지를 명시적으로 표현할 수는 없다. 이러한 문제점을 해결하기 위해, 웹을 최초로 만든 팀버너스리(Tim Berners-Lee)는 컴퓨터간의 정보교환이 가능하게 하며 웹 상의 데이터의 의미를 컴퓨터가 이해하여 처리할 수 있는 새로운 정보기술로써 시멘틱 웹(Semantic Web)을 고안해 내었다(Tim Berners-Lee, James Hendler and Ora Lassila, 2001).

[그림 1] 전자도서관 운영환경의 변화

의미정보를 제대로 전달하기 힘든 기존의 웹 정보체계는 서로 다른 도서관시스템간의 의미 정보교환을 위한 표준 규약(protocol)을 제공하지 못한다. 이것은 마치 전세계 어디라도 국제전화로 연결이 가능하다하더라도 한쪽은 영어로 나머지 한쪽은 한국어로 이야기한다면 서로간의 의사소통이 불가능한 것과 마찬가지이다. 대부분의 도서관시스템이 동일한 통신규약(network protocol)으로 웹 프로토콜(HTTP)을 채택한다하더라도, 서로간의 의미정보 표현 방식이 다르다면 이기종 도서관시스템간의 직접적인 상호자료연동은 원천적으로 불가능하게 된다. 하지만 모든 도서관시스템들이 동일한 정보시스템 체계를 유지하기는 현실적으로 불가능하며, 결국 서로 다른 시스템의 상호 의미전달을 위한 표준을 필요로 하게 된다. 시멘틱 웹 연구자들은 서로 다른 시스템간의 상호 의미정보교환을 위한 표준을 제공하기 위한 연구를 계속 진행해오고 있으며, 이러한 표준들을 토대로 서로 다른 도서관시스템간의 상호 의미정보교환을 위한 다양한 시도가 이루어지고 있다. 새로운 전자도서관 운영환경인 시멘틱 웹에서 XML, RDF, 웹 온톨로지와 같은 의미정보표현에 관한 표준들에 대해서는 다음 2장에서 구체적으로 살펴보기로 하겠다. 3장에서는 메타데이터를 웹 온톨로지로 활용하는 방안과 차세대 전자도서관인 개방형 전자도서관모델에 대해서 살펴보고, 서로 다른 정보체계구조를 지닌 개방형 전자도서관간의 상호운영성을 제공하기 위한 구체적인 방안을 제시하였다. 마지막 4장에서는 상호운영성을 제공하기 위한 구체적인 방법인 응용프로파일 자동변환 알고리즘의 장점 및 활용방안과 후속 연구계획에 대해 설명하였다.

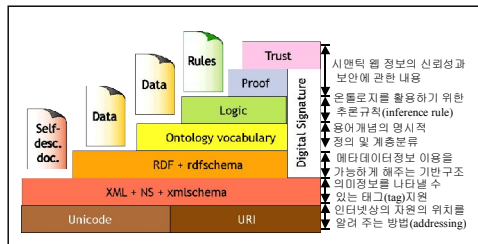
## 2. 메타데이터기반 시멘틱 웹 구축

시멘틱 웹을 구축과정에서 가장 중요한 단계 중 하나는 웹 정보자원에 대한 정형화된 메타데이터를 작성하는 것이며, 메타데이터의 집합체인 시멘틱 웹은 하나의 거대한 메타데이터 저장소로 볼 수 있다. 시멘틱 웹은 메타데이터의 개념과 유사하게 웹 문서에 의미정보를 덧

КСІ

붙이고, 이를 이용하여 컴퓨터가 이 의미정보를 자동으로 추출할 수 있는 패러다임을 제공하게 된다. 새로운 전자도서관 운영환경인 시멘틱 웹에서 XML, RDF, 웹 온톨로지와 같은 의미정보표현에 관한 표준들에 대해 구체적으로 살펴보기로 한다.

## 2.1. 의미정보교환을 위한 시멘틱 웹 표준



[그림 2] 시멘틱 웹의 웹 계층구조

시멘틱 웹의 계층구조는 맨 아래 웹 통신프로토콜(HTTP)에서 특정 리소스의 물리적 위치를 지정하기 위한 주소체계인 URI계층이 존재하며, 이를 토대로 XML계층, RDF계층, 온톨로지(Ontology)계층, 로직(Logic)계층, 증명(Proof)계층, 신뢰(Trust)계층 순으로 구성되어 있다(송중철, 이동일, 문병주, 2002). 시멘틱 웹의 계층구조에서 가장 하위 레벨인 UR는 웹 프로토콜에서 자원을 지칭하기 위한 주소지정(addressing) 방법을 제공한다. 흔히 웹 브라우저 주소창에서 'http://'로 시작하는 URL주소는 여러 가지 URI방식중 하나이다. XML은 HTML의 태그의 한계와 SGML의 문법의 복잡성 등을 개선한 표준 마크업언어이다. XML에서는 의미정보를 나타낼 수 있는 태그 이름을 사용자가 자유롭게 정의할 수 있기 때문에 의미정보 전달을 가능하게 해주는 기반을 제공하게 된다. RDF는 메타데이터를 처리하기 위한 언어로써, 메타데이터 정보이용을 가능하게 해주는 기반을 제공하고 있다. 시멘틱 웹의 온톨로지계층에서 사용하는 웹 온톨로지(Web Ontology)는 어떤 사용되어지는 범위를 한정 시킨 일종의 어휘사전이다. 어휘가 가지는 모든 의미를 처리할 수 있는 만능시스템을 구현하기란 매우 어렵기 때문에, 해당 어휘가 사용되는 각 주제도메인의 국한된 범위 내에서 가질 수 있는 의미를 제한하기 위하여 웹 온톨로지를 사용하게 된다. 정보시스템에서 웹 온톨로지는 사람이 직관적 또는 목시적으로 해당 어휘의 의미를 판단하여 처리하는 부분을 컴퓨터가 대신 처리할 수 있도록 하는데 사용되는 용어(vocabulary)사전에 역할을 수행하게 된다. 웹 온톨로지 언어로는 OIL(Ontology Inference Layer), DAML(DAR PA Agent Markup Language), SHOE(Simple HTML Ontology Extensions)와 최근에 W3C에서 개발중인 Notation 3(N3)가 있다. OIL(Adam Pease, 2002)과 DAML(Frank van Harmelen and Ian Horrocks, 2000)은 RDF/RDF Schema를 기반으로한 마크업 언어이며, SHOE(Sean Luke and Jeff Heflin, 2000)는 HTML에서 온톨로지 표현이 가능한 언어이다. N3(Tim Berners-Lee, 2001)는 RDF와 매핑을 통하여 상호보완적 역할을 수행하는 것을 목적으로, RDF와 로직, 그리고 데이터를 하나의 언어로 표현하기 위한 논리언어이다. 로직계층은 온톨로지를 이해하고 해석하기 위한 추론규칙(inference rule)을 제공하며, 최상위 계층인 증명계층과 신뢰계층은 시멘틱 웹 정보의 신뢰성과 보안에 관한 내용을 다루게 된다(Hewlett-Packard Company,2003).

## 2.2. 시멘틱 웹에서 XML의 역할

XML(eXtensible Markup Language)이란 SGML의 일부분(subset)으로 구성된 마크업 언어로써, 기존의 HTML의 태그의 한계와 SGML의 문법의 복잡성 등을 개선하여 W3C에서 제안한 표준 마크업언어이다. 기존의 HTML은 정보가 기록된 데이터영역과 눈으로 보여지는 뷰(view)영역을 하나의 문서로 표현하였다. 하지만 XML은 눈으로 보여지는 부분은 XML의 데이터영역으로부터 XSL이라는 스타일시트(style sheet)로 분리하여 보다 명확한 의미정보전달이 가능해졌다. 또한 XML은 시멘틱 웹과 관련하여 태그의 이름을 사용자가 자유롭게 정의할 수 있기 때문에 의미정보를 나타낼 수 있는 태그 이름을 사용자가 지정할 수 있다는 점에서 의미정보 전달을 가능하게 해주는 기반체계를 제공한다.

```
<note>
<date> 2004-02-10 </date>
<to> 현희 </to>
<from> 미노 </from>
<heading> 전체공지 </heading>
<body> 오전 9시부터 전체회의 시작</body>
</note>
```

[그림 3] 의미정보를 나타낼 수 있는 태그

[그림 3]의 XML 문서는 간단한 메모의 작성을 표현하기 위한 것으로서, 태그의 이름만 살펴봐도 메모를 보낸 사람(from), 메모를 받는 사람(to), 제목(heading), 메모내용(body) 등의 의미를 쉽게 파악할 수 있다. 하지만 이러한 방법은 웹 프로그래머마다 같은 의미를 뜻하면서도 각기 서로 다른 형태의 용어(tag)를 정의하여 사용할 수 있다는 문제점을 가지게 된다. [그림 3]에서 메모의 제목을 <heading>으로 표기하였지만 개발자에 따라 같은 의미의 <subject>라는 다른 태그로 정의하여 사용할 수도 있다. 따라서 이러한 상호운용성문제를 해결하기 위해서는 근본적으로 메타데이터 또는 웹 온톨로지를 통해 어휘의미중의성문제(word sense ambiguity)를 해결해야 한다(황상규, 오경목, 변영태, 1999). 그밖에 또 다른 문제점으로, 같은 내용에 대해서도 여러 가지 서로 다른 구조를 가질 수 있는 XML문서의 다형성이 있다. [그림 3]의 메모 내용 중에서 날짜를 연월일로 구분해서 표현하면 [그림 4]와 같은 다른 형태의 구조가 된다. XML문서의 내용과 구조가 복잡해질수록 이러한 XML문서의 다형성은 사람과 시스템 모두 XML문서의 내용을 이해하고, 처리하는 과정을 힘들게 하는 요인으로 작용하게 된다.

```

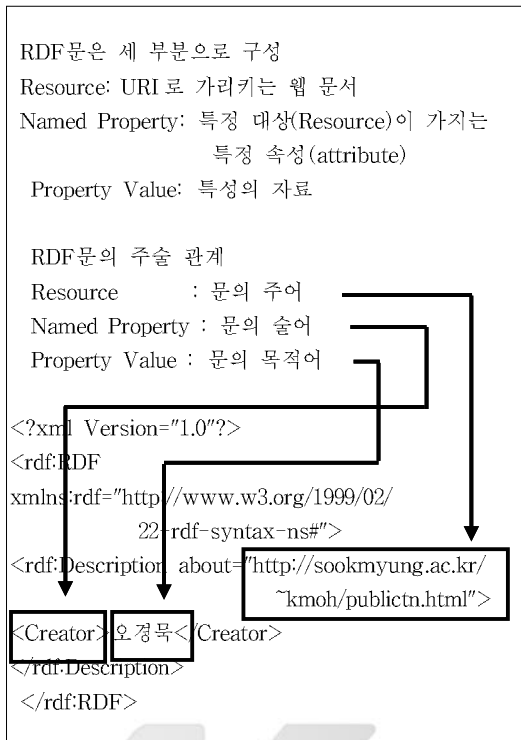
<note>
<date>
<day>17</day>
<month>10</month>
<year>2003</year>
</date>
<to> 현희 </to>
<from> 미노 </from>
<heading> 전체공지 </heading>
<body> 오전 9시부터 전체회의 시작 </body>
</note>

```

[그림 4] XML문서의 다형성

### 2.3. 시멘틱 웹에서 RDF의 필요성

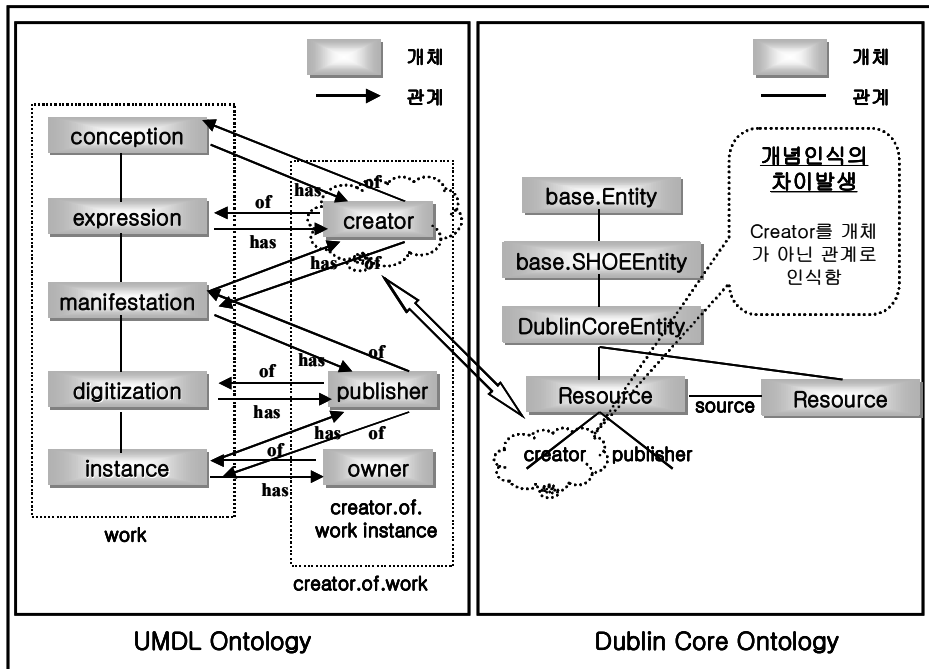
시멘틱 웹에서 RDF는 복잡한 XML문서의 구조를 간소화시키고 정형화하여, 사람과 시스템이 XML문서의 내용을 이해하고, 처리하는 과정의 편리성을 제공하는 역할을 하게 된다. RDF는 XML형태로 표기되어지는 대상자원의 의미정보를 표현하기 위한 언어로써, 웹 문서 등 특정 대상자원(Resource)이 가지는 특정 속성(attribute)에 대하여 특정 값(value)을 기술하기 위해서 일반 문장의 주어, 동사, 목적어에 해당하는 3가지 정보로 구성되어 있다. [그림 5]의 RDF문장은 “<http://sookmyung...> has <creator> <오경목>”이라는 의미를 가지며, “<주어>는 <술어><목적어>를 갖는다.”로 읽어나가면 된다. 기본적으로 RDF 모델에서는 대상의 의미가 항상 리소스(주어)와 속성(술어), 속성값(목적어)으로 함께 묶어 표현되므로, XML과는 달리 같은 내용에 대해서는 항상 동일한 구조를 가지게 된다. 또한 XML에서와 같이 서로 다른 구조를 가진 여러 가지 표현방법이 존재하지 않기 때문에 문서의 내용에 대한 이해가 쉽다. 하지만 RDF에서도 XML의 문제점 중 하나였던 태그 이름의 중첩성과 모호성은 여전히 존재하게 된다. 즉 서로 다른 태그이지만 실제로는 같은 의미일 수 있고, 반대로 같은 태그이지만 사용자에 따라서 다른 의미로 쓰일 수도 있다. 이러한 문제를 해결하기 위해서는 메타데이터 또는 웹 온톨로지정보를 사용해야 하며, RDF에서는 웹 온톨로지와 유사한 RDF스키마가 존재한다. RDF스키마는 일종의 어휘사전으로, RDF에서 특정태그의 의미는 스키마 참조를 통해 이루어진다. 여기서, 스키마 참조란 일종의 RDF에서 사용하게 되는 어휘의미사전의 위치를 지정해주는 것이다. 서로 다른 사용자가 특정 태그용어에 대해 동일한 의미로 이해하기 위해서는 사용자 누구나 항상 동일한 어휘의미사전을 사용해야 혼란을 막을 수 있기 때문이다. RDF에서는 XML의 네임스페이스기능을 이용하여, 웹 문서상의 특정 RDF태그의 사용용도가 정의된 사전(스키마)의 위치를 알려주게 된다. 하지만, RDF에서는 여러 개의 스키마를 동시에 사용할 수 있기 때문에 잘못된 스키마운영에 따른 중첩과 모호함이 발생할 수 있는 가능성을 완전히 배제할 수 없다. 이는 마치 사용자가 임의로 영어단어장을 만들어 사용할 때 검증된 영어사전과는 달리 오류발생의 빈도가 높은 것과 마찬가지로이다. 따라서, 어휘의미중의성문제를 해결하기 위해 사용하는 웹 온톨로지는 반드시 믿을 수 있는 기관에서 만든 것만을 이용해야하며, 이와 동시에 많은 사용자가 쉽게 접근할 수 있도록 이미 널리 알려진 웹 온톨로지를 사용해야 한다.



[그림 5] RDR의 기본구성

#### 2.4. 시멘틱 웹에서 웹 온톨로지의 필요성

시멘틱 웹에서 웹 온톨로지를 사용함으로써 XML에서 의미정보를 나타낼 수 있는 태그 이름을 사용자정의가 가능함으로써 발생하는 어휘의미중의성문제를 해결하게 된다. 하지만, 일반적으로 웹 온톨로지는 단순히 단어의 개념정의뿐 아니라 단어들 간의 계층/관계정보, 추론규칙 등을 함께 정의해야 함에 따라 구현과정이 복잡해지며 쉽게 활용하기도 힘들다는 단점을 가지고 있다. 이러한 웹 온톨로지 구축과 활용상의 제약은, 시멘틱 웹 활성화의 가장 큰 장애요인 중 하나로 여겨지고 있다. 실제 웹 온톨로지 기반 전자도서관 운영시 문제점을 살펴보기 위하여, MARC을 기초로 하여 전자도서관을 위해 구축된 UMDL Ontology (Peter Weinstein, 1998)와 SHOE방식으로 표현한 Dublin Core Ontology(Sean Luke,2000) 간의 상호운영성 방안을 검토해보기로 하겠다. UMDL Ontology를 표현한 [그림 6]은, 사각형으로 표시된 개체(entity)와 화살표로 표시된 관계(relation)로 구성되어 있다. UMDL Ontology에서 행위(act) 또는 행위의 주체(actor)는 개체로 표현되며, 개체와 개체간에는 관계(relation)가 성립된다. 행위의 주체인 '창작자(creator)'는 '표현물(expression)의 창작자(of 관계)'이며, "창작자는 표현물을 소유한다(has관계)."라고 정의할 수 있다. 하지만, Dublin Core Ontology에서 '창작자'는 UMDL Ontology와는 달리 하나의 독립된 '개체'가 아닌 '관계'로 정의된다. Dublin Core Ontology설계자는 '창작자'를 Resource라는 개체가 가지는 하나의 속성 관계라고 개념을 정의한 것이다.



[그림 6] UMDL Ontology와 Dublin Core Ontology간의 개념차이 비교

이는 근본적으로 Dublin Core Ontology설계자와 UMDL Ontology설계자간에 사물을 보는 근본적인 시각차를 보여주고 있는데, 정보학적 관점에서 서지정보를 다루는 MARC과 Dublin Core에서 '창작자'는 거의 유사한 개념임에도 불구하고, 웹 온톨로지를 모델링하는 설계자의 관점에 따라 한쪽은 '개체'로 한쪽은 '관계'로 인식하고 있다. 이 경우, 모델링 개념이 서로 다른 두 웹 온톨로지 간에는 직접적인 상호연동이 불가능하며, 상호연동을 위한 별도의 변환규칙을 필요로 하게 된다. 결론적으로 웹 온톨로지 구성은 웹 온톨로지를 모델링하는 설계자의 관점에 따라 동일한 내용도 전혀 다른 형태로 표현 될 수 있으며, 서로 다른 웹 온톨로지 기반 시스템간의 상호연동 역시 매우 힘든 일이 아닐 수 없다. 이러한 문제점의 근본원인은 웹 온톨로지서 사물에 대한 정의를 내릴 때, 항상 명시적 정의(explicit definition)를 통해 수학기호처럼 명확한 개념정의를 만들려는 특성에서 기인한다. 사물에 대한 기본 정의가 명확할 때 웹 온톨로지를 토대로 추론규칙의 적용이 가능해지며, 잘못된 추론의 발생빈도도 낮아지게 된다. 하지만, 웹 온톨로지 설계자는 세부적 사항까지도 명시적으로 정의해야 함에 따라, 서로 다른 웹 온톨로지 기반시스템 간의 상호연동시 유연성 결핍의 문제를 발생시킨다.

### 3. 응용프로파일 메타데이터구조

웹 온톨로지를 이용하여 의미정보의 중첩과 모호성을 해결하더라도 서로 다른 웹 온톨로지 기반 시스템간 상호연동은 쉽게 이루어질 수 없다. 이러한 웹 온톨로지의 단점을 보완하기 위해 메타데이터를 웹 온톨로지로 활용하는 방안에 대해 검토해 보았으며, 이를 토대로 시멘틱 웹 환경하에서 이기종 전자도서관 시스템간에 상호운영성 문제를 해결할 수 있는 새로운 방식의 응용프로파일 메타데이터구조를 개발하였다. 응용프로파일(application profile)이란 사용자가 기존에 널리 알려진 메타데이터표준들로부터 자신에 사용목적에 부합되는 메타

데이터요소들을 추출하여 자신만의 새로운 메타데이터를 만드는 사용자정의 메타데이터 생성방법이다(이수상, 2002). 응용프로파일 메타데이터구조에서는 이미 기존에 널리 알려진 메타데이터표준들로부터 XML은 의미정보를 나타낼 수 있는 태그 이름을 지정함으로써, 사용자임의로 태그이름을 부여함으로써 발생하는 어휘의미중의성문제를 사전에 예방한다.

### 3.1. 시멘틱 웹에서 메타데이터의 활용

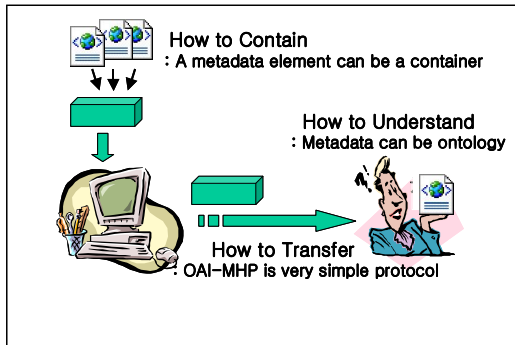
웹 온톨로지를 활용하기 위한 로직단계에서 사용되는 추론규칙은 수학에서의 정확성과 엄격성이 요구되며, 이러한 정확성과 엄격성은 웹 온톨로지 설계 시에도 동일하게 적용되어진다. 더구나, 모든 경우에 통용되는 추론규칙을 만들어내기가 어려운 만큼, 모든 추론규칙에 사용될 수 있는 웹 온톨로지 표준을 만들어내는 것 역시 매우 어려운 일이다. 이러한 문제를 해결하기 위한 방안으로 시멘틱 웹 환경하에 전자도서관 운영시 필요로 하는 웹 온톨로지의 최소기능을 검토해 보았으며, 이 과정에서 메타데이터가 웹 온톨로지의 역할을 대신 수행 할 수 있음을 확인하였다. 웹 온톨로지의 기본특성과 메타데이터의 주요기능을 비교, 정리한 결과([표 1])를 보면, 메타데이터의 주요기능들은 웹 온톨로지가 갖추어야 할 5가지 기본특성을 모두 만족시키고 있음을 확인할 수 있다. 따라서 메타데이터는 시멘틱 웹 환경하에 전자도서관 운영시 웹 온톨로지의 기능을 대신하여 수행할 수 있다.

기본특성	웹 온톨로지의 기본특성	메타데이터의 주요기능
명료성	어휘의미중의 성해소를 통한 상호운영성 보장	어휘의 사용범위와 의미가 명확한 한정어 (qualifier) 표시의 특징
단순성	의미를 이해하기 위한 최소한의 추론규칙	인코딩스킴 (통제어휘집에서 선정된 요소명, 날짜에 경우, 통상적인 표준규칙을 적용한 일반적 표기법을 그대로 사용함) ex. 날짜 = YYMMDD 또는 YYYY-MM-DD
신뢰성	제공되는 의미정보의 신용도 보장	해당분야 전문가집단의 논의와 투표를 거쳐 한정어를 확정지음
인지도	접근의 용이성	대부분의 메타데이터는 이미 여러분야에서 널리 활용되어 왔음
유연성	추가 및 확장의 용의성	기존 메타데이터 요소들을 조합하여 새로운 메타 데이터 생성

[표 1] 웹 온톨로지의 기본특성과 메타데이터의 주요기능간의 비교

### 3.2. 개방형 전자도서관모델

전세계적으로 전자도서관의 발전추세는 다양한 디지털자원에 대한 통합에서, 분산환경의 전자도서관들 간의 통합과 연동으로 관심의 초점이 바뀌어가고 있다. 차세대 전자도서관 모델은 개방성, 표준성, 상호운용성을 요구하는 개방형 전자도서관 모델로서, 아직까지는 현재 실험중이거나 서비스 개발중인 초기단계에 머물고 있다. 따라서, 개방형 전자도서관표준모델은 아직까지 존재하지 않으며, 일반적인 개방형 전자도서관모델의 기본운영개념을 [그림 7]과 같이 정리하였다.



[그림 7] 개방형 전자도서관모델의 기본운영개념

개방형 전자도서관모델의 기본운영개념에서 정보의 표현방법(How to Contain)은 메타데이터요소를 <dc:title>처럼 RDF태그명으로 선언함으로써 해당 태그 속성값의 정의와 용도가 한정되게 된다. 일반적으로 RDF문서는 XML문법에 따라 시작태그와 끝태그로 둘러싼 데이터로 구성되며, 해당 태그는 데이터의 속성을 표시하게 된다. 결국 태그는 '단일 데이터를 담는 그릇(container)'의 역할을 수행하게 된다. 만약 어떤 태그가 특정 메타데이터의 요소로 태그이름이 정의되었다면, 메타데이터의 요소로 정의된 특정태그로 둘러싼 데이터는 해당 메타데이터의 요소의 특성을 그대로 상속받게 된다.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf=
    "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1">
  <rdf:Description about=
    "http://www.arXiv.org/ea">
    <dc:title>My Example</dc:title>
  </rdf:Description>
</rdf:RDF>
```

[그림 8] 메타데이터 기반 RDF 태그

[그림 8]은 더블린 코어의 어휘가 RDF상에서 어떻게 태그로 정의되어 '단일 데이터를 담는 그릇'으로 사용될 수 있는지를 보여주는 예이다. <dc:title>라는 메타데이터 기반 RDF태그가 선언되어 사용됨으로써 해당 태그의 속성값인 'My Example'은 더블린 코어의 메타데

이터요소인 표제(Title)라는 속성 그대로 상속받아 'My Example'의 정의와 용도가 한정되게 된다. 'My Example'은 [그림 5]의 사례에서처럼 RDF 모델에서 대상의 의미가 리소스와 그 속성값으로 표현되므로, 같은 내용에 대해서는 항상 동일한 의미를 가지게 되며, 의미정보의 중첩과 모호성을 해결하게 된다.

정보의 전달방법(How to Transfer)은 개방형 전자도서관을 위한 통신프로토콜 표준에 하나인 OAI-MHP프로토콜을 사용하여 표현하였는데, OAI-MHP프로토콜은 전자도서관들간 표준적인 상호운용성을 제공을 목적으로 전자도서관 구축을 위한 표준모델인 ISO의 OAIS 참조모델에서 채택한 표준통신규약이다(이수상, 2002). OAI-MHP프로토콜은 매우 단순한 구조로 구성되어 있는데, 클라이언트는 데이터제공자의 OAI 서버에게 HTTP의 GET또는 POST 방식을 사용하여 6가지 명령(Identify, GetRecord, ListIdentifiers, ListRecords, ListSets, ListMetadataFormats)으로 요청(request)을 전달하며, 요청을 받은 데이터제공자의 OAI 서버는 각 요청명령에 대한 응답(response)데이터를 규정된 XML스키마에 유효하도록 레코드를 재구성하여 보내게 된다. [그림 9]는 OAI-MHP프로토콜의 6가지 명령 중 레코드 검색(GetRecord)을 보여주고 있는데, 검색대상인 특정 레코드 검색키(oai:eg:001)와 레코드의 메타데이터 형식(oai\_dc)을 지정하여 서비스를 요청하면 해당 레코드가 검색되어 응답데이터가 전송되게 된다.



[그림 9] OAI-MHP 프로토콜을 이용한 의미정보전달

전달된 정보의 이해(How to Understand)는 메타데이터를 웹 온톨로지로 활용함으로써 해결될 수 있는데, [그림 8]과 같은 검색결과를 넘겨받은 사용자 혹은 시스템이 더블링크어 메타데이터에 대해 알고 있다면 'My Example'은 '해당 콘텐츠의 표제'라고 명확히 이해할 수 있다. 다음은 서로 다른 정보체계를 가진 시스템간의 전달된 의미정보를 이해하는 방법에 대해 살펴보기로 하겠다. [그림 10]은 UIUC에서 개발한 RDF형식의 UIUC메타데이터스키마(Timothy W. Cole, et. al, 2003)와 응용프로파일스키마를 비교해서 보여주고 있다. 응용프로파일스키마는 기존 UIUC 메타데이터스키마에 단순히 전자명함의 역할을 수행하는 메타데이터인 vCard(Roland Alden, et.al, 1998)를 추가하여, 새로운 응용프로파일 메타데이터를 정의하였다.

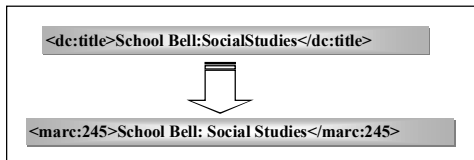
	UIUC 메타데이터스키마		응용프로파일스키마
의미 정보 이해	<pre> &lt;rdf:RDF xsi:schemaLocation= "http://.../uiuc_dcq_rdf.xsd" xmlns:xsi="http://...XMLSchema-instance" xmlns:rdf="http://...rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcq="http://purl.org/dc/terms/"  &lt;rdf:Description about = 'http://.../view.asp?ID=1133352537'&gt; &lt;dc:title&gt;School Bell: Social Studies&lt;/dc:title&gt; &lt;dc:creator&gt; &lt;rdf:Description&gt; Joe Smith joe@my.com My Company, Inc. &lt;/rdf:Description&gt; &lt;/dc:creator&gt;  &lt;dc:subject&gt;Social sciences&lt;/dc:subject&gt; &lt;dc:subject&gt;geography&lt;/dc:subject&gt; &lt;/rdf:Description&gt; &lt;/rdf:RDF&gt; </pre>	↔	<pre> &lt;rdf:RDF xsi:schemaLocation= "http://.../uiuc_dcq_rdf.xsd" xmlns:xsi="http://...XMLSchema-instance" xmlns:rdf="http://...rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcq="http://purl.org/dc/terms/"  <b>xmlns:vcard= "http://.../vcard/3.0/"</b>  &lt;rdf:Description about = 'http://.../view.asp?ID=1133352537'&gt; &lt;dc:title&gt;School Bell: Social Studies&lt;/dc:title&gt; &lt;dc:creator&gt; &lt;rdf:Description&gt; Joe Smith &lt;vcard:email&gt; joe@my.com &lt;/vcard:email&gt; &lt;vcard:org&gt; My Company, Inc. &lt;/vcard:org&gt; &lt;/rdf:Description&gt; &lt;/dc:creator&gt;  &lt;dc:subject&gt;Social sciences&lt;/dc:subject&gt; &lt;dc:subject&gt;geography&lt;/dc:subject&gt; &lt;/rdf:Description&gt; &lt;/rdf:RDF&gt; </pre>

[그림 10] 서로 다른 메타데이터스키마간의 의미정보이해

[그림 10]에서 어렵게 표시된 부분은 <dc:creator>를 보다 구체적으로 기술하기 위해 또 다른 메타데이터인 vCard의 <vCard:email>과 <vCard:org>라는 태그를 추가한 것이다. 이 경우, vCard를 지원하는 응용프로파일스키마기반 도서관시스템과 vCard를 지원하지 않는 UIUC스키마기반도서관시스템간의 OAI-MHP2.0프로토콜기반으로 상호연동을 할 경우, UIUC도서관시스템은 vCard를 지원하지 않기 때문에 전달된 의미정보를 이해하는 과정에서 <dc:title>시작태그와 </dc:title>끝태그 사이에 모든 문자열을 단순히 dc:title의 속성값으로 인식하게 된다. 이는 “일반적인 메타데이터 운영개념에서, 메타데이터요소의 상세구분을 정의하지 않는 경우 가장 최상위의 메타데이터요소의 개념으로 대상을 이해하면 된다.”는 암시적 정의(implicit definition)를 추론규칙으로 이용하여 이기종 도서관시스템간의 정보아키텍처의 구조적 차이를 극복할 수 있기 때문이다 이처럼 비교적 유사한 형태의 메타데이터 구조를 지닌 이기종 도서관시스템간의 상호연동은, 약간의 구조적 차이가 발생하더라도 기존 웹 온톨로지 방식과는 달리 별도의 자료변환을 위한 추론규칙 없어도 어느 정도 상호운영성이 보장될 수 있다.

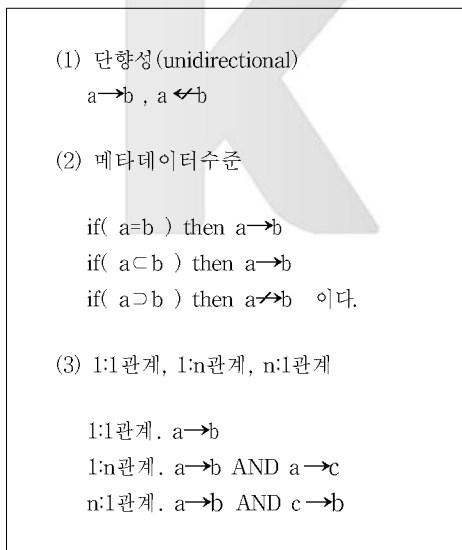
### 3.3. 응용프로파일 메타데이터 자동변환

지금까지 메타데이터를 웹 온톨로지로 활용하여 서로 유사한 정보체계를 가진 전자도서관 간의 상호운영성 문제를 해결하는 방안에 대해 살펴보았다. 다음은 서로 상이한 정보체계를 가진 전자도서관간의 상호운영성 문제해결방안을 살펴보고, 서로 다른 정보체계를 가진 전자도서관시스템간의 상호운영성방안에 대해 살펴보기로 한다.



[그림 11] 다른 메타데이터요소간 변환

일반적으로 서로가 상이한 정보체계를 가진 경우, 서로 다른 시스템간에 상호연동이 이루어지기 위해서는 별도의 변환프로그램을 필요하게 된다. 하지만 메타데이터를 웹 온톨로지로 활용하여 XML이나 RDF태그명으로 정의하는 경우, [그림 11]에서처럼 엘리먼트 단위로 보다 쉽게 서로 다른 정보체계간의 자동변환이 이루어질 수 있다.



[그림 12] 메타데이터 연관정보의 기본적 특성

```

(1) 단방향성
표제 (dc:title) → 설명 (dc:description)
표제 (dc:title) ↗ 설명 (dc:description)

(2) 메타데이터수준
if(표제 (dc:title) = 표제 (marc:245) )
then 표제 (dc:title) → 표제 (marc:245)
if(표제 (marc:245) ⊂ 설명 (dc:description))
then 표제 (marc:245) → 설명 (dc:description)
if(설명 (dc:description) ⊃ 표제 (marc:245))
then 설명 (dc:description) ↗ 표제 (marc:245)

(3) 1:1 관계, 1:n 관계, n:1 관계
1:1 관계.
표제 (dc:title) → 표제 (marc:245)
1:n 관계.
주제 (marc:600) → 주제 (dc:subject) AND
주제 (marc:600) → 설명 (dc:description)
n:1 관계.
표제 (marc:245) → 설명 (dc:description) AND
주제 (marc:600) → 설명 (dc:description)

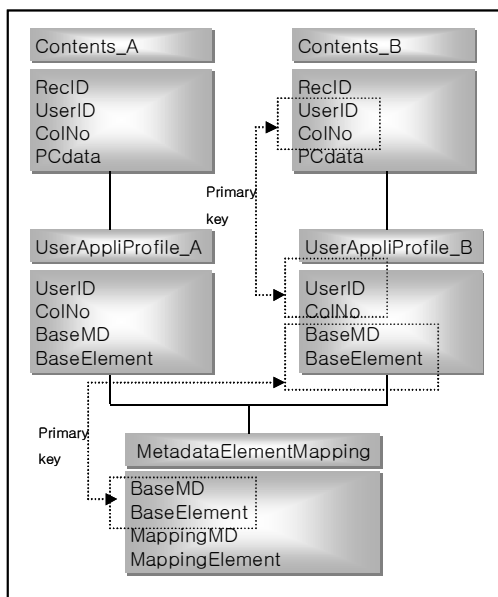
```

[그림 13] 연관정보 기본적 특성의 예

다음은 서로 다른 메타데이터요소간 매핑을 위한 메타데이터 연관정보(metadata element mapping)의 기본적 특성에 대해 살펴보기로 하겠다. 앞으로 소개할 메타데이터 연관정보의 3가지 기본적 특성은 메타데이터를 웹 온톨로지로 활용할 때 기본적으로 적용되는 추론규칙으로 활용할 수 있다. 일반적으로 서로 다른 메타데이터요소간 연관정보 구축시 고려해야 할 기본적 특성은 [그림 12]와 같다. [그림 13]의 예제에서 연관정보의 기본적 특성 중 첫 번째 특성은, 서로 다른 메타데이터의 요소들간 매핑시 이행관계는 단방향성(unidirectional)을 지닌다는 점이다. 메타데이터항목 중 표제(title)는 다른 메타데이터항목인 설명(description)의 일부로 포함될 수 있으나, 그 역관계는 성립되지 않는다. 따라서 메타데이터간 연관정보 구축시에는 반드시 서로 다른 메타데이터 항목간의 관계를  $a \rightarrow b$ 관계와  $b \leftarrow a$ 관계로 따로 분리하여 검토해보아야 한다. 두 번째 특성으로, 메타데이터항목구분의 수준에 따라 이행관계가 성립될 지 여부를 검토해 보아야 한다. 먼저 두 메타데이터항목의 수준이 같을 경우를 생각해 볼 수 있는데, 'dc:title'과 'marc:245' 모두 메타데이터항목 중 표제를 가리키는 요소로 서로간에 대응관계가 성립함에 따라 'dc:title → marc:245'라는 이행관계가 성립하게 된다. 그 다음으로 한쪽이 다른 한쪽에 포함되는 경우로, 'marc:245'와 'dc:description'간의 관계를 생각해볼 수 있다. 이 경우, '표제'는 다른 메타데이터항목인 '설명'의 일부로 포함될 수 있으므로 'marc:245 → dc:description'관계는 성립하나 그 역관계는 성립하지 않게 된다. 마지막 특성으로, 서로 다른 메타데이터의 요소들간 매핑 시 대응관계는 1:1 관계, 1:n 관계, n:1 관계라는 3가지 종류로 구분할 수 있다. 이러한 대응관계는 기존 MARC과 더블린코어간 상호운영성 연구사례에서 그 예를 찾아볼 수 있는데(Library of Congress,2001), 더블린코어

의 'Subject'는 MARC필드 중 '600, 610, 611, 630, 650, 653'에 대응되며 이에 따라 1:n관계 (n:1관계)가 성립하게 된다. 지금까지 살펴본 연관정보의 기본적 특성들은 일반적 통념상 허용될 수 있는 메타데이터의 묵시적 정의로부터 유도될 수 있는 추론규칙이다. 만약 서로 다른 시스템간의 상호운영성을 지원하기 위해 웹 온톨로지를 사용할 경우, 위와 같은 3가지 특성들은 반드시 문서화된 명시적 규칙으로 정의되어야 하나, 메타데이터를 웹 온톨로지로 활용할 경우 메타데이터를 이용하는 모든 도서관시스템들은 이미 위와 같은 3가지 추론 규칙을 묵시적으로 이해하고 사용할 수 있다.

다음은 위와 같은 3가지 묵시적 정의에 의한 추론규칙을 토대로, 서로 다른 응용프로파일 정보구조를 가진 도서관시스템간의 메타데이터 자동변환 알고리즘에 대해 살펴보기로 하겠다. 일반적으로 대량의 메타데이터자료를 처리하기 위한 가장 보편적인 방법으로 데이터베이스(database)를 사용하게 되며, 데이터베이스에서 가장 사용하는 질의처리 프로그램으로 SQL이라는 프로그램언어가 있다. HTML 혹은 XML문서로 표현되는 메타데이터자료들은 보다 효과적인 정보처리를 위해 시스템 내부적으로 데이터베이스의 형태로 존재하게 된다. 이러한 특성을 고려하여 새로 개발한 메타데이터 자동변환 알고리즘은 보다 효율적인 정보처리를 위해 SQL언어를 이용하여 알고리즘을 구현하였다. SQL언어는 데이터베이스에서 정보를 얻거나 갱신하기 위한 표준화된 프로그램언어로서 SQL을 지원하는 대화형 프로그램을 이용하거나, 별도의 프로그램 소스코드 내에 삽입하여 쓸 수 있다. SQL언어는 데이터베이스 종류와 상관없이 실행이 가능하다는 장점과 데이터베이스를 이용함에 따라 대량의 데이터 처리가 가능하다는 이점을 가지고 있다. 따라서, 응용프로파일 자동변환 알고리즘을 SQL언어로 표현함으로써 전자도서관시스템의 데이터베이스 종류에 무관하게 모든 데이터베이스에서 실행이 가능하며, 수십만건 이상의 대용량 메타데이터 처리시에도 해당 프로그램을 그대로 사용할 수 있다.



[그림 14] 자동변환을 위한 스키마 구조

	항목	용도		
테이블명	Contents_A	실제 메타데이터가 저장되는 부분(기준테이블)		
	Contents_B	실제 메타데이터가 저장되는 부분(변환대상테이블)		
	UserApplicationProfile_A	기준테이블의 응용프로파일 구성		
	UserApplicationProfile_B	변환대상테이블의 응용프로파일 구성		
	MatadataElementMapping	응용프로파일 메타데이터요소간 매핑정보		
컬럼명	UserID	관리자번호	UserID와 colNo을 이용하여 Contents 테이블	
	colNo	컬럼번호	과 UserApplicationProfile 간의 1:1 관계가 성립	
	RecNo	레코드번호	Contents 테이블에 PCdata항목에 실제 메타데이	
	PCdata	해당레코드값		터요소값이 저장됨
	BaseMD	기준메타데이터명	메타데이터명이 DC,marc이고, 대응메	
	BaseElement	기준메타데이터요소명		타데이터요소명이 각각 title, 245일때,
	MappingMD	대응메타데이터명		기준레코드와 대응레코드간의 비교
	MappingElement	대응메타데이터요소명		예) DC.title = marc.245

[표 2] 자동변환을 위한 스키마 구조의 항목별 내용

[그림 14]에서는 서로 다른 응용프로파일 정보구조를 지원하기 위해 새로 설계한 데이터베이스 스키마 구조를 보여주고 있다(데이터베이스 스키마 항목에 대한 구체적 설명은 [표 2]를 참조). 메타데이터 자동변환 알고리즘은 총 5개의 테이블로 구성되어 있는데, UserApplicationProfile\_A(또는 B)이라는 테이블은 사용자가 정의한 해당 도서관시스템의 응용프로파일 정보구조를 정의하게 되며, Contents\_A(또는 B)이라는 테이블에는 실제 응용프로파일형식에 따른 실제 메타데이터자료가 들어가게 된다. 테이블명에서 ‘\_A’는 기준테이블을 의미하며, ‘\_B’는 변환대상테이블임을 의미한다. 마지막으로 MatadataElementMapping이라는 테이블은 서로 다른 메타데이터의 요소들간 매핑정보가 저장된다.

응용프로파일 자동변환 알고리즘	
설명	<ul style="list-style-type: none"> <li>■ 기준테이블의 응용프로파일 (UAP_A)과 응용프로파일 메타데이터요소간 매핑정보 (MEM)로부터</li> </ul>
	<ul style="list-style-type: none"> <li>■ 기준레코드와 대응레코드간의 매핑관계가 성립할 때(예. DC.title = marc.245)</li> </ul>
	<ul style="list-style-type: none"> <li>■ 매핑관계가 성립하는 메타데이터요소 (MD)와 메타데이터요소명(Element)을 조회하라.</li> </ul>
■	<pre>select MEM.BaseMD, MEM.BaseElement, MEM.MappingMD, MEM.MappingElement</pre>
■	<pre>from UserAppliProfile_A UAP_A, MetadataElementMapping MEM</pre>
■	<pre>where UAP_A.BaseMD= MEM.BaseMD and AP_A.BaseElement= MEM.BaseElement and UAP_A.UserID='admin01'</pre>

[그림 15] 응용프로파일 자동변환 알고리즘

[그림 15]는 서로 다른 응용프로파일의 메타데이터항목간 자동변환 알고리즘을 SQL언어를

통해 표현하고 있다. 일반적으로 SQL언어는 크게 Select절-From절-Where절이라는 세부분으로 구성된다. 먼저 SQL의 From절에는 질의처리대상인 테이블명을 기술하고, Where절에는 질의의 검색조건이 기술되며 Select절에서는 검색결과로 조회되는 컬럼명이 기술된다. From절의 테이블들로부터 Where 절에 기술된 검색조건이 만족되는 자료들 중 검색을 원하는 테이블의 필드명을 Select절에 기술함으로써 하나의 SQL문이 완성되는 것이다. 또한 SQL의 의 From절에는 테이블명이 길어질 경우, 단축이름(alias)을 선언할 수 있다. [그림 15]의 예제에서는 From 절에서 'UserAppliProfile\_A' 라는 테이블명 대신 단축이름인 'UAP\_A'를 선언하고 있으며, 나머지 Select절과 Where절에서는 실제 테이블명 대신 단축이름인 'UAP\_A'를 사용하게 된다. 응용프로파일 자동변환 알고리즘은 해당 도서관시스템의 기준 응용프로파일 (UAP\_A)과 메타데이터간요소간의 상호매핑정보(MEM)를 이용하여 기준 응용프로파일과 매핑 가능한 모든 메타데이터요소들을 추출하게 된다. 실제 하나의 서버에 여러 개의 전자도서관시스템 프로그램을 따라 운영할 수 있으며, 이 경우 하나의 데이터베이스안에 응용프로파일의 기준메타데이터가 운영되는 프로그램 개수만큼 존재할 수 있다. 보통 하나의 시스템의 관리자는 한명이므로, 특정 시스템을 구분하기 위해 where절에서 조회조건으로 관리자ID(UAP\_A.UserID = 'admin01')를 추가하였다.

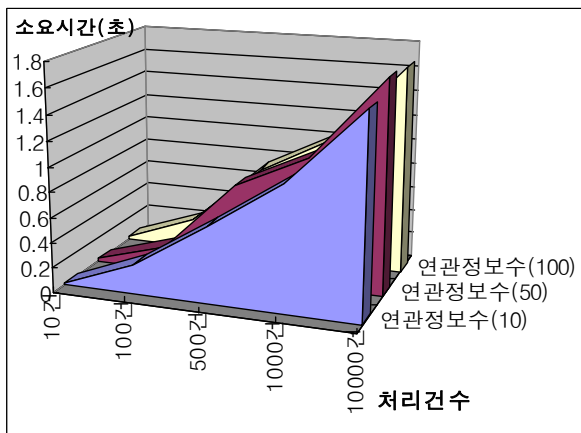
### 3.4. 응용프로파일 자동변환 알고리즘의 활용

지금까지 메타데이터를 웹 온톨로지로 활용하는 방법과 메타데이터의 묵시적 정의로부터 유도될 수 있는 추론규칙에 대해 살펴보았다. 또한, 메타데이터의 묵시적 정의에 의한 추론규칙을 토대로, 서로 다른 응용프로파일 정보구조를 가진 도서관시스템간의 메타데이터 자동변환 알고리즘을 개발하였다. 다음은 응용프로파일메타데이터 자동변환 알고리즘을 응용하여 실제 시스템에 적용시킨 하나의 사례로, 공동서지목록(Union Catalog)을 자동생성하는 방법에 대해 살펴보기로 한다.

[그림 16]에서는 응용프로파일메타데이터 자동변환 알고리즘을 응용하여 실제 시스템에 적용시킨 하나의 사례로 공동서지목록을 자동생성하는 방법을 SQL언어로 표현하였다. SQL은 크게 Select-From-Where절이라는 세 부분으로 구성되는데, 복잡한 질의를 처리하기 위해 From절과 Where절 안에 또 다른 Select-From-Where절을 선언하는 중첩구조를 허용하고 있다. [그림 16]의 예제에서는 중첩된 구조 안에 또 다른 중첩이 존재하며, 중첩된 구조를 보기 쉽도록 중첩 수준(depth)을 '0','1','2' 단계로 나눠서 정의하였다. SQL처리시 중첩된 구조에서 가장 수준이 낮은 라벨 AAA부분(depth=2)이 가장 먼저 처리되는데, AAA는 [그림 15]에서 설명한 응용프로파일의 메타데이터항목간 자동변환 알고리즘에 해당한다. 그 다음 단계로 라벨 BBB부분(depth=1)이 처리되는데, 라벨 BBB는 라벨 AAA의 처리결과와 테이블 UAP\_B를 이용하여 기준테이블의 응용프로파일과 다른 서버로부터 수집된 변환대상 응용프로파일간의 상호 매핑관계를 계산하게 된다. 마지막으로 가장 바깥에 위치한 라벨 Result(depth=0)가 수행되며, 최종 조회결과로 기준 메타데이터요소와 대응될 수 있는 연관 메타데이터명(BBB.MD\_A), 연관메타데이터요소명(BBB. Ele\_A)와 연관메타데이터요소의 실제값(C\_B.PCdata)이 조회되었다.

라벨명 (label)	수준 (depth)	공동서지목록 자동생성 알고리즘
라벨 설명	2	AAA = {기준 UserAppliProfile A와 메타데이터간 요소간의 상호매핑정보를 이용하여 기준 UserAppliProfile A와 매핑가능한 모든 메타데이터요소를 추출, AAA는, [그림 15]의 응용프로파일의 메타데이터항목간 자동변환 알고리즘임} UAP_B={다른 서버로부터 수집된 AppliProfile B}
	1	BBB = {AAA와 UAP_B를 이용하여 기준 UserAppliProfile_A 와 다른 서버로부터 수집된 UserAppliProfile_B간의 상호 매핑관계를 계산함}
	0	C_B = {다른 서버로부터 수집된 컨텐츠정보} Result = {BBB와 C_B를 이용하여 다른 서버로부터 수집된 컨텐츠정보를 AppliProfile A 형식으로 자료변환한 결과}
Result	0	select BBB.MD_A, BBB.Ele_A, C_B.PCdata from
BBB	1	(select AAA.BaseMD MD_A, AAA.BaseElement Ele_A, UAP_B.BaseMD MD_B, UAP_B.BaseElement Ele_B, UAP_B.UserID UserID_B, UAP_B.colNo colNo_B from
AAA	2	(select MEM.BaseMD, MEM.BaseElement, MEM.MappingMD, MEM.MappingElement from UserAppliProfile_A UAP_A, MetadataElementMapping MEM where UAP_A.BaseMD= MEM.BaseMD and AP_A.BaseElement= MEM.BaseElement and UAP_A.UserID='admin01') AAA,
UAP_B	2	UserAppliProfile_B UAP_B
BBB	1	where AAA.MappingMD=UAP_B.BaseMD and AAA.MappingElement=UAP_B.BaseElement) BBB,
C_B	1	Contents_B C_B
Result	0	where BBB.UserID_B = C_B.UserID and BBB.colNo_B=C_B.colNo

[그림 16] 응용프로파일 자동변환 알고리즘 기반 공동서지목록 자동생성



[그림 17] 공동서지목록 자동생성 평균 소요시간

[그림 17]은 한 레코드당 공동서지목록 자동생성시 평균소요시간을 측정결과를 그래프로 표현하였다. 실험을 위한 테스트환경으로 Enterprise450서버를 사용하였으며, SQL언어를 테스트를 위한 데이터베이스시스템으로는 Oracle 7.3을 사용하였다. 본 실험에서 비교대상 레코드수는 10000건까지 증가시키면서 처리속도를 측정하였으며, 비교대상 레코드수보다 상대

적으로 적은 개수인 연관정보수(MetadataElementMapping테이블의 레코드 수)는 100개까지 증가시키면서 처리속도를 비교해 보았다. 비교대상 레코드수 변화에 따라 1000건을 비교처리하는데 대략 1초가 소요되었으며, 10000건의 데이터를 처리하는데 대략 2초가 소요되었다. 대체적으로 공동서지목록 자동생성 평균 소요시간은 대상 레코드 수가 증가함에 따라 소요시간도 증가하였으나, 메타데이터항목간 연관정보수 증가는 전체처리속도에 별로 영향을 미치지 않음을 확인할 수 있다. 실험결과를 살펴보면, 특정 메타데이터항목과 수집된 다른 메타데이터항목들 간의 연관비교가 대략 1~2초 이내에 이루어짐을 확인할 수 있다. 이는 공동서지목록자동생성 알고리즘을 활용하여 인터넷을 기반으로 자동화된 시스템을 구현시, 실시간처리시스템 구현이 가능하다는 장점을 가지게 된다.

	처리단계	
	구분	주요사항
질의		수집된 다양한 메타데이터들 가운데 주제가 'Z39.50'인 자료조회
	조회식	해당 시스템에서 주제필드로 marc:600을 사용하는 경우, query( marc:600 = 'Z39.50')
SQL 처리		수집된 다양한 메타데이터들 가운데 marc:600에 대응되는 레코드셋조회
	중간 처리 결과 (A)	marc:600 marc:610 marc:611 ... dc:subject dc:decription
		매핑된 레코드셋 중 레코드값에 'Z39.50'이라는 단어가 나오는 최종레코드조회
	최종 처리 결과 (B)	marc:600 = '서지학' marc:610 = 'Z39.50' marc:611 = '연구방법론' ... dc:subject = 'Z39.50 연구' dc:decription = '도서관정보 시스템에서 Z39.50의 적용사례'

[그림 18] 응용프로파일 자동변환기반 통합검색 알고리즘의 수행방식

다음은 응용프로파일메타데이터 자동변환 알고리즘의 또 다른 응용 사례로, 서로 다른 메타데이터 포맷간의 자료변환 및 통합작업이 필요 없는 통합검색 알고리즘을 개발하였다. [그림 18]은 통합검색 알고리즘의 수행방식을 처리단계별로 소개하고 있는데, “수집된 다양한 메타데이터들 가운데 주제가 'Z39.50'인 자료를 조회하라.”라는 질의요청에 대한 SQL처리과정을 단계별로 보여주고 있다. 중간처리결과는 수집된 다양한 메타데이터요소들 가운데 marc:600에 대응되는 레코드셋 조회결과를 보여주고 있으며 최종처리결과는 중간처리 후

언어인 레코드셋에서 해당 레코드의 내용이 'Z39.50'인 자료만을 조회하게 된다. [부록. 1]은 통합검색질의에서 중간처리결과를 보여주기 위한 SQL조회식과 통합검색질의를 위한 실제 SQL조회식을 보여주고 있다. 통합검색 알고리즘은 서로 다른 메타데이터 포맷간의 자료변환 없이 메타데이터연관정보만을 이용하여 이기종 전자도서관시스템들로부터 수집된 자료에서 직접 통합검색이 가능함에 따라, 서로 다른 이기종 전자도서관시스템들로부터 수집된 자료의 통합을 위한 재가공 비용을 최소화할 수 있다.

#### 4. 결론 및 향후발전방향

지금까지 우리는 새로운 웹 정보체계인 시멘틱 웹 환경하에서, 전자도서관 구축방안의 새로운 운영패러다임에 대해 검토해보았다. 시멘틱 웹은 기존의 HTML기반 웹 정보체계와는 달리 웹 상의 데이터의 의미를 사람이 아닌 컴퓨터가 이해하여 처리할 수 있도록 지원해주는 새로운 정보 기술으로써, 이를 토대로 이기종 전자도서관간의 상호운영성문제를 데이터수준에서 해결할 수 있는 실마리를 얻게 되었다. 기존 HTML기반 웹 정보체계하에서 전자도서관간의 상호운영성문제를 해결하기 위해서는 서로 다른 시스템 각각마다 별도의 변환프로그램을 필요로 하였으나, 시멘틱 웹 환경 하에서는 XML과 RDF, 웹 온톨로지등을 활용하여 시멘틱 웹 표준규약에 따라 상호간 의미정보교환이 가능하게 되었다. XML과 RDF, 웹 온톨로지등을 활용하여 상호간 의미정보를 교환하는 과정에서 해결해야할 문제점들과 해결방안에 대해 살펴보았으며, 서로 다른 웹 온톨로지기반 시스템간의 상호연동시 발생하는 유연성 결핍의 문제를 해결하고자 메타데이터를 웹 온톨로지로 활용하는 방안을 제시하였다. 그밖에 메타데이터를 웹 온톨로지로 활용할 때 메타데이터의 묵시적 정의로부터 유도될 수 있는 추론규칙에 대해 정의하였으며, 이를 토대로 서로 다른 응용프로파일의 메타데이터항목간 자동변환 알고리즘을 새로 개발하였다.

지금까지 살펴본 메타데이터를 웹 온톨로지로 활용한 응용프로파일 자동변환 알고리즘의 장점은 다음과 같다. 첫째, 짧은 기간 내에 생성되고, 갱신되며 소멸하는 웹 리소스에 특성상 수시로 공동서지목록을 갱신해야하는 전자도서관 혹은 검색엔진의 디지털저장소관리에서, 사람의 수작업을 대신해줄 수 있는 응용프로파일 자동변환 알고리즘을 이용한 공동서지목록 자동생성기능은 매우 빠르고 효과적인 디지털 저장소 관리기법으로 이용될 수 있다. 둘째, 서로간 메타데이터 구성체계가 틀리더라도 최소한의 메타데이터 항목간 매핑정보만 얻을 수 있다면, 이기종 시스템간의 상호운영성을 지원할 수 있다. [그림 18]의 통합검색질의사례에서처럼 서로 다른 메타데이터 포맷간의 자료변환없이 수집된 자료로부터 직접 통합검색이 가능하다. 셋째, 새로운 형태의 메타데이터나 새로운 정보체계를 가진 시스템이 개발되더라도 기존 메타데이터 항목과 매핑정보만 추가하면 시스템간의 상호운영성을 지원할 수 있음으로써 확장성과 유연성이 뛰어나다. 마지막으로, 응용프로파일 자동변환 알고리즘을 SQL언어로 표현함으로써 전자도서관시스템의 데이터베이스 종류에 무관하게 모든 데이터베이스에서 응용프로파일 자동변환 알고리즘 수행이 가능하다.

응용프로파일 자동변환 알고리즘을 실제 적용한 공동서지목록 자동생성에서 10000건의 데이터를 비교하는 과정이 불과 1~2초 사이에 신속히 이루어짐을 확인하였다. 또한 본 논문에서 제시한 메타데이터를 웹 온톨로지로 활용하는 방안은 전자도서관 구축 외에 다른 분야에

도 적용가능한데, 대표적인 예로 온라인상에서 서적, 음반 등 디지털콘텐츠 판매하는 전자상거래시스템 분야의 카달로그 관리나 경영정보를 제공 위한 지식관리시스템의 핵심인 지식베이스 관리 등에 활용할 수 있다. 응용프로파일기반 통합검색 알고리즘은 서로 다른 메타데이터 문서포맷 간의 자료변환없이 메타데이터연관정보만을 이용하여 수집된 자료로부터 직접 통합검색이 가능하며, 자료의 통합을 위한 재가공 비용을 최소화할 수 있음에 따라 정보통합시스템이나 통합전자도서관 구축 등의 분야에 널리 활용될 수 있다.

응용프로파일 자동변환 알고리즘을 실제 전자도서관시스템에 적용하기 위한 후속 연구로, 서로 다른 메타데이터의 요소들간의 자동변환시 매핑관계에 대한 신뢰도를 수치화하여 알고리즘의 신뢰도를 향상시키는 방안에 대한 연구를 계속해서 진행해 나갈 예정이다. 매핑관계에 대한 신뢰도를 수치화하는 가장 큰 이유는, 현재의 알고리즘으로는 메타데이터 요소간 매핑정보테이블에는 매핑관계에 대한 신뢰도를 수치화 할 수 없기 때문에 자동변환시 우선순위선택이 불가능하다. 일례로 'marc:710'에서 'dc:creator'로의 매핑신뢰도가 90%이고 정의할 수 있고, 'marc:710'에서 'dc:creator'로의 매핑신뢰도가 70%라면, 1:n 관계가 성립할 때 신뢰도 우선순위에 따라 매핑신뢰도가 가장 높은 'dc:creator'를 먼저 선택하게 될 것이다. 이러한 매핑신뢰도의 수치화를 현실적으로 적용하기 위해서는 “정상적인 신뢰도를 어떠한 기준에 의해 정량화 할 것인가”라는 문제와 “자동변환을 여러 번 반복할 때 신뢰도 값이 계속 줄어드는 편향성 현상”과 같은 문제점들을 계속해서 해결해나가야 한다. 또한, 집합연산에 비해 복잡한 수식연산이 어려운 SQL언어의 특성상, 우선순위 계산 과정에서 복잡해지는 수식연산을 SQL언어로 표현하는 방안에 대한 연구도 계속해서 진행해나갈 예정이다.

#### \*참고문헌\*

이수상, 2002, “전자도서관의 최근 동향”, 데이터베이스 연구 18 권 3 호  
<<http://www.sigdb.or.kr/publish/18-3/0003-0017.pdf>>

송중철, 이동일, 문병주, 2002, “Semantic Web의 표준화 및 요소기술 개발 동향”, 정보통신연구진흥원 IT정보단. <[ttp://kidbs.itfind.or.kr:8888/WZIN/jugidong/1064/106402.htm](http://kidbs.itfind.or.kr:8888/WZIN/jugidong/1064/106402.htm)>

황상규, 오경목, 변영태, 1999, “어휘의미 중의성이 인터넷 정보검색 효율에 미치는 영향에 관한 연구”, 한국정보관리학회지 제16권 제3호. <<http://sookmyung.ac.kr/~kmoh>>

Tim Berners-Lee, James Hendler and Ora Lassila, May 2001, “The Semantic Web” Scientific American .  
<  
<http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2>>

Hewlett-Packard Company, 2003, “Introduction to Semantic Web Technologies”.  
<<http://www.hpl.hp.com/semweb/sw-technology.htm>>

Adam Pease, 2002, “Why Use DAML?”. <<http://www.daml.org/2002/04/why.html>>

Frank van Harmelen and Ian Horrocks, 2000, "Questions and answers on OIL : the Ontology Inference Layer for the semantic web", IEEE Intelligent Systems volume 15, number 6, pages 69-72. <<http://www.ontoknowledge.org/oil/oil-faq.html>>

Sean Luke and Jeff Heflin, 2000, "SHOE 1.01", SHOE Project. <<http://www.cs.umd.edu/projects/plus/SHOE/spec.html>>

Tim Berners-Lee, 2001, "Notation 3". <<http://www.w3.org/DesignIssues/Notation3.html>>

Peter Weinstein. 1998. "Ontology-Based Metadata: Transforming the MARC Legacy" Proceedings of the Third ACM Digital Library conference, Pittsburgh, PA, USA, June. <<http://www-personal.umich.edu/~peterw/Ontology/Beethoven/demo.html>>

Sean Luke, 2000, "Dublin Core Ontology", <<http://www.cs.umd.edu/projects/plus/SHOE/onts/dublin.html>>

Catherine C. Marshall and Frank M. Shipman, 2003, "Which Semantic Web?", Hypertext'03 Conference. <[www.csdl.tamu.edu/~marshall/ht03-sw-4.pdf](http://www.csdl.tamu.edu/~marshall/ht03-sw-4.pdf)>

Thomas G. Habing, et. al, 2002, "The University of Illinois Open Archives Initiative Metadata Harvesting Project", <<http://oai.granger.uiuc.edu/ProviderTools/TemplateBReadme.htm>>

Timothy W. Cole, et. al, 2003, "Open Archives Initiative Metadata Harvesting Project", <<http://oai.granger.uiuc.edu/index.htm>>

Roland Alden, et.al, 1998, "RFC 2426, vCard MIME Directory Profile", <<http://www.ietf.org/rfc/rfc2426.txt>>

Library of Congress, 2001, "MARC to Dublin Core Crosswalk", <<http://www.loc.gov/marc/marc2dc.html>>

부록. 1. 응용프로파일 자동변환기반 통합검색질의

A. 중간처리결과 조회식

- 통합검색질의에서 중간처리결과를 보여주기 위한 조회식임

중간처리결과 조회식	
설명	① 응용프로파일 자동변환 알고리즘 실행 후
	② UserAppliProfile_B 테이블로부터
	③ marc:600 에 대응되는 레코드셋 얻음
질의식	<pre> select AAA.BaseMD MD_A,        AAA.BaseElement Ele_A,        UAP_B.BaseMD MD_B,        UAP_B.BaseElement Ele_B,        UAP_B.UserID UserID_B,        UAP_B.colNo colNo_B from (select MEM.BaseMD, MEM.BaseElement,        MEM.MappingMD, MEM.MappingElement from UserAppliProfile_A UAP_A , MetadataElementMapping MEM where UAP_A.BaseMD= MEM.BaseMD and UAP_A.BaseElement=MEM.BaseElement and MEM.BaseMD='marc' and MEM.BaseElement='600' and UAP_A.UserID='admin01') AAA, UserAppliProfile_B UAP_B where AAA.MappingMD=UAP_B.BaseMD and AAA.MappingElement=UAP_B.BaseElement                     </pre>
	①
②	

조회식 실행 후, 중간처리결과					
MD_A	ELE_A	MD_B	ELE_B	USERID_B	COLNO_B
marc	600	dc	description	user_e	2
marc	600	dc	subject	user_e	1
marc	600	marc	600	user_b	1
marc	600	marc	610	user_c	1
marc	600	marc	611	user_d	1

**B. 최종처리결과 조회식**

- 통합검색질의를 위한 실제 조회식임

중간처리결과 조회식	
실 명	① 응용프로파일 자동변환 알고리즘 실행 후
	② UserAppliProfile_B 과 Contents_B 테이블로부터
	③ marc:600에 대응되며, 해당 레코드의 내용이 'Z39.50'인 자료만을 조회
질 의 식	<pre> select AAA.BaseMD MD_A,        AAA.BaseElement Ele_A,        UAP_B.BaseMD MD_B,        UAP_B.BaseElement Ele_B,        UAP_B.UserID UserID_B,        UAP_B.colNo colNo_B,        C_B.PCDATA from           </pre>
	<pre> ( select MEM.BaseMD, MEM.BaseElement,       MEM.MappingMD, MEM.MappingElement from       UserAppliProfile_A UAP_A,       MetadataElementMapping MEM where       UAP_A.BaseMD= MEM.BaseMD and       UAP_A.BaseElement= MEM.BaseElement       and MEM.BaseMD='marc'       and MEM.BaseElement='600'       and UAP_A.UserID='admin01' ) AAA,           </pre>
	<pre> UserAppliProfile_B UAP_B, Contents_B C_B where AAA.MappingMD=UAP_B.BaseMD and       AAA.MappingElement=UAP_B.BaseElement       and UAP_B.UserID=C_B.UserID       and UAP_B.colNo=C_B.colNo       and C_B.PCDATA like '%Z39.50%'           </pre>

- 통합검색질의 후 최종처리결과

MD_A	ELE_A	MD_B	ELE_B	USERID_B	COLNO_B	PCDATA
marc	600	dc	description	user_e	2	도서관정보시스템에서 Z39.50의 적용사례
marc	600	dc	subject	user_e	1	Z39.50 연구
marc	600	marc	610	user_c	1	Z39.50