

# 응용프로파일 코어 온톨로지 설계 및 구현

## A Design and Implementation of Application Profile Core Ontology

한성국(Sung-Kook Han)\*, 이현실(Hyun-Sil Lee)\*\*

### 초 록

유비쿼터스 정보 환경하에서 정보자원의 공유와 상호 교환을 위한 정보자원의 구조와 내용 기술에 표준 메타데이터 체계가 이용되고 있다. 실제 응용 도메인에서는 다수의 메타데이터 요소를 혼합-일치 방식으로 재사용하여 응용 시스템을 구축하게 되는데, 이때 메타데이터 요소의 상세화와 상호 운용성 등의 문제가 발생한다. 메타데이터 활용에서 발생하는 문제 해결에 응용 프로파일 접근 방식이 이용되고 있다. 본 논문에서는 응용 프로파일의 목적과 기능을 달성할 수 있는 응용 프로파일 코어 온톨로지를 제시하고, 이를 기반으로 한 메타데이터 응용 시스템 구축에 대하여 서술하였다.

### ABSTRACT

The standard metadata systems are very popular for the description of structures and semantics of information resources to realize sharing and exchanging information in global ubiquitous environment. In real application domain, various metadata elements are reused together with mix-and-match manner. An application system using diverse metadata systems is compelled with refinement and interoperability of metadata elements. Application profile is the general approach to resolve the various problems occurred in metadata application systems. This paper proposes Application Profile Core Ontology (APO) that can achieve the goals and functions of application profile, and describes metadata application system based on APO.

키워드 : 응용 프로파일, 코어 온톨로지, 메타데이터, 상호 운용성  
application profile, core ontology, APO, metadata, interoperability

---

\* 원광대학교 컴퓨터공학과 교수 (skhan@wku.ac.kr)

\*\* 원광대학교 도서관 사서 (hyunsil@wku.ac.kr)

■ 논문접수일자 : 2007년 8월 20일

■ 게재확정일자 : 2007년 9월 8일

## 1. 서 론

정보통신의 급속한 발달로 인하여 초고속 인터넷과 인터넷이 일상화되었다. 고성능의 컴퓨터 네트워크는 정보전달과 공유에 새로운 전기가 되었으며, 웹2.0에서와 같은 정보의 개방과 공유는 참여와 협력을 통하여 새로운 지식 정보 사회를 선도하는 기초가 되고 있다 (Maness 2006). 네트워크상에서 정보전달과 공유를 실현하기 위해서는 정보자원의 특성을 이해하기 쉽고 컴퓨터의 처리가 가능하도록 명확하게 기술할 수 있어야 한다. 즉, 정보자원이 내포하고 있는 데이터에 대한 특성을 기술할 수 있는 메타데이터가 요구된다.<sup>1)</sup> 메타데이터는 정보자원의 식별, 공유, 검색, 평가와 관리 등에서 요구되는 구조적, 의미적 정보를 명시적으로 기술하여 제공함으로써 정보처리의 핵심 요소가 되고 있다.

XML이 등장하여 정보자원의 위치, 구조, 내용과 의미 등 정보자원의 특성을 기술할 수 있는 간편한 방식이 제공됨에 따라 다양한 메타데이터들이 개발되었다. 일반 서지정보를 기술하는 더블린 코어(Dublin Core: DC)를 비롯하여 MARC21 레코드의 XML 기술을 위한 MODS, 전자 텍스트를 위한 TEI, 학습객체를 기술하는 IEEE LOM이나 SCORM, 예술품을 기술하는 CDWA, 박물관 자원을 위

한 CIMI 등 모든 분야에서 메타데이터 개발이 활발하게 진행되었다.<sup>2)</sup> 이처럼 메타데이터는 여러 분야에서 정보자원 관리에 중추적 역할을 담당하게 되었으며, 서지정보 기술 뿐 만 아니라, 자원의 통제와 관리, 사용자 프로파일, 저작권 보호 등 다양한 영역에서 활용되고 있다.<sup>3)</sup> 다양한 메타데이터 표준들이 개발되어 메타데이터 기반의 정보 활용 시대가 되었다고 볼 수 있다.

메타데이터 스키마는 일반적으로 특정분야의 이용을 목적으로 포괄적 개념 요소를 정의하고 있다. 예를 들어, IEEE LOM은 교육 콘텐츠의 학습객체 기술을 위한 상위 개념 요소를 정의하고 있다<sup>4)</sup>. 메타데이터 스키마의 이러한 특성으로 인하여, 정보시스템 구축시 하나의 메타데이터 스키마만으로 정보자원의 내용을 충실하게 기술하기가 어렵다. 따라서 여러 메타데이터 스키마를 혼합-일치(mix-and match) 방식으로 합성해서 활용하게 된다. 또한 메타데이터 스키마에서 포괄적으로 정의된 메타데이터 요소를 응용 도메인의 국소적 요구를 충족할 수 있도록 재정의해야 할 경우도 발생한다. 응용 도메인에서 필수적으로 요구되는 메타데이터의 혼합과 재정의는 표준 메타데이터 상호 운용성 문제를 만들고, 메타데이터의 상호 운용성이 보장되는 정보시스템 구축이 큰 과제이다.

1) 메타데이터의 정의나 관점에 대하여는 다양한 견해가 있으며, Duval, Smith, and Coillie(2006)의 문헌을 참고할 수 있다.

2) 다양한 분야에서 개발된 메타데이터 현황을 MetaMap에서 살펴볼 수 있다.

<<http://www.mapageweb.umontreal.ca/turner/meta/english/index.html>>

3) Gilliland Swetland는 메타데이터를 관리 메타데이터, 서지기술 메타데이터, 보존 메타데이터, 기술 메타데이터, 활용 메타데이터의 5가지 범주로 구분하고 있다. 그러나 메타데이터는 정보서비스의 모든 영역에서 활용되고 있다.

4) [http://ltsc.ieee.org/wg12/files/LOM\\_1484\\_12\\_1\\_v1\\_Final\\_Draft.pdf](http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf)

메타데이터 스키마 통합을 위한 워릭 프레임 워크(Warick Framework)가 제안된 이래로, 메타데이터 응용에서 발생하는 상호 운용성 등의 문제를 해결하기 위한 여러 방안이 제시되었다. 특히, 응용 프로파일(application profile)은 메타데이터 스키마의 특성을 실현 하면서 응용 도메인의 요구사항을 수렴할 수 있는 스키마로 인식되고 있다(Heery and Patel 2000). 응용 프로파일은 여러 메타데이터 스키마로부터 응용 도메인에서 요구되는 메타데이터 요소를 추출하여, 혼합한 응용 시스템 메타데이터 스키마로서 상호 운용성 보장 및 상세화 등의 메타데이터 기반 응용 시스템 구축에 필요한 기능을 제공한다. 응용 프로파일의 중요성이 부각됨에 따라, 정형화된 응용 프로파일 구축에 관한 여러 연구가 수행되었다. SCHEMA 프로젝트(2003)에서는 RDF 스키마를 이용하여 응용 프로파일을 기술하고 있다. RDF 스키마는 메타데이터 요소의 의미를 명확하게 표현할 수 있지만, 요소값의 특성이나 카디널리티(cardinality) 등을 표현하는 데는 한계가 있다. 이 때문에 RDF 스키마로 기술된 응용 프로파일은 효용성에 문제가 있다. XML 스키마를 이용해서 응용 프로파일을 구현한 사례도 있다(Hunter 2000). 이 경우에는 메타데이터의 구조적 특성은 명확하게 정의할 수 있지만 의미 표현에 제약이 있다. RDF 스키마와 XML 스키마를 결합하여 응용 프로파일의 유효성을 고찰한 연구도 있다(Hunter and Lagoze 2001). 이질적인 RDF

스키마와 XML 스키마의 혼합은 이들간의 인터페이스와 연결 방식이 복잡하고, 시스템 구축과 처리의 효율성에 문제가 있다. 한편, 응용 프로파일의 검증(validation)에 중점을 두어 OWL과 XDD를 혼합하여 사용한 응용 프로파일 기술 방식도 제안되었다(Ratanajaipan, Nantajeewarawat, and Wuwongse 2006a). 이 방식은 메타데이터 요소 규칙을 위해 별도의 규칙 언어 XDD를 사용함으로써 메타데이터 기반 시스템을 복잡한 규칙기반의 지식시스템으로 만들고 있다.<sup>5)</sup>

지금까지 응용 프로파일 연구는 주로 기술 방식에 대한 것들이다. XML이 정보 구조화와 정보 전달의 기본 언어로 사용되고 있고 메타데이터 스키마가 XML 형태로 표현되므로, 응용 프로파일 기술 방식은 XML 형식이어야 한다. RDF 스키마 또는 XML 스키마 형태의 표현은 궁극적으로 응용 프로파일의 온톨로지 정의 방법과 관련된 사항이라고 할 수 있다. 상호 운용성을 보장하고 메타데이터 요소의 특성을 명확하게 정의할 수 있는 응용 프로파일 코어 온톨로지(core ontology)가 개발되어야 한다. 응용 프로파일 코어 온톨로지를 이용해서 응용도메인 특성에 적합한 응용 프로파일 구축을 정형화할 수가 있다. 본 논문에서는 메타데이터 스키마 구조를 분석하여 메타데이터 요소의 특성을 명시적으로 표현하고, 메타데이터간의 상호 운용성을 정의할 수 있는 응용 프로파일 코어 온톨로지(application profile core ontology: APO)를 설계, 개발하고 온

5) Ratanajaipan 등이 제기한 DataCreated 일자가 DateIssued 일자에 선행하는가의 검증을 위해서는 XDD와 같은 복잡한 규칙언어의 도입은 시스템의 복잡성을 가중시킨다.

톨로지 표준 언어인 RDF 스키마로 그 사례를 보인다.

본 논문은 제1장 서론을 포함하여 다음과 같이 구성된다. 제2장에서는 메타데이터 특성과 상호 운용성에 대해 분석하고, 이를 실현하는 응용 프로파일의 기능과 기술방식에 대하여 고찰한다. 제3장은 응용 프로파일 코어 온톨로지 설계로 DC, IEEE LOM과 SKOS 등의 메타데이터 요소 기술방식을 분석하여 응용 프로파일에서 사용할 온톨로지를 도출한다. 또한, 도출한 온톨로지 간의 개념 모델을 설계하고 각 어휘에 대한 의미 정의를 제시한다. 제4장에서는 응용 프로파일 코어 온톨로지를 RDF 언어로 구현하고 응용 프로파일 기술 사례를 보인다. 제5장은 결론으로 본 논문의 성과와 향후 연구 방향에 대하여 논한다.

## 2. 메타데이터 응용 프로파일

응용도메인에서 다수의 메타데이터를 혼합-일치 방식으로 사용할 때 발생하는 상호 운용성 등의 문제를 해결하기 위한 도구로 응용 프로파일의 활용이 보편화되고 있다. 본 장에서는 응용 프로파일의 개념, 요구기능 및 계층모델 등 응용 프로파일 전반에 대하여 고찰한다.

### 2.1 메타데이터 상호 운용성과 응용 프로파일

멀티미디어, 인터넷 등 정보기술의 급속한

진화로 다양한 형태와 내용의 정보자원이 대량 생산되고 정보자원의 디지털화가 촉진됨에 따라, 이들을 효과적으로 기술하고 관리하기 위한 여러 메타데이터 표준안이 개발되었다. 일반적으로 메타데이터 스키마는 특정 도메인에서 범용적으로 사용되는 개념 요소를 포괄적으로 정의한다. 메타데이터의 이러한 목적 지향적 범용성으로 인하여, 응용 도메인에서는 하나의 메타데이터 스키마만으로 정보자원의 특성을 표현할 수 없다. 이 때문에 응용 도메인에서는 하나 이상의 메타데이터 스키마를 합성해서 사용하게 된다. 이러한 필요성에 의하여 DC 개발과 더불어 다수의 메타데이터 스키마를 혼합하여 사용하기 위한 워릭 프레임워크가 제안되었다. 워릭 프레임워크는 메타데이터 스키마를 패키지(package)로 구조화하여, 컨테이너(container)로 혼합하는 방식을 제시하고 있다.

다수의 메타데이터 스키마를 혼합하여 사용하는 경우, 표준 메타데이터 스키마와 호환성 유지, 메타데이터 요소간의 의미적 충돌 해소, 응용 도메인의 요구사항 수용 등 상호 운용성 문제가 있다. 상호 운용성이란 정의된 표준 메타데이터 요소의 구조적, 의미적 특성이 서로 다른 분산 시스템 간에서 그대로 유지되는 것을 뜻한다.<sup>6)</sup> 메타데이터 상호 운용성이 실현되어야 응용 시스템간 동일한 정보 해석이 가능하고 정보전달과 공유체제가 구축될 수 있으므로, 분산 네트워크 정보시스템 구현에 선행 요소가 된다.

6) INDES에서는 상호 운용성을 “어떤 환경에서 발생한 정보를 고도의 자동화된 방법으로 다른 환경에서 그대로 사용할 수 있는 것”으로 정의하고 있다(Rust and Bide 2000).

상호 운용성을 효과적으로 실현하기 위한 방안으로 워릭 프레임워크를 개선한 응용 프로파일 개념이 제시되었다.<sup>7)</sup> 워릭 프레임워크에서는 메타데이터 스키마를 패키지화하여 혼합 사용하는데 비하여, 응용 프로파일에서는 개별 메타데이터 스키마를 네임스페이스에 의해 개별적으로 혼합하는 방식을 사용하고 있다.<sup>8)</sup>

응용 프로파일은 응용 도메인에서 정보자원의 특성을 명시적으로 정의할 목적으로 여러 메타데이터 스키마에서 메타데이터 요소를 추출, 혼합한 응용 도메인 스키마이다(Heery 2000). 응용 프로파일은 표준 메타데이터를 기반으로 구성되므로, 메타데이터 상호 운용성 또는 호환성을 유지하면서 응용 도메인의 요구를 표현할 수 있어야 한다. 정의된 표준 메타데이터와 응용 도메인 요구 사이에서 발생하는 상호 운용성에는 여러 형태가 있다. 예를 들어, 음악 정보자원을 관리하는 응용 도메인에서 정보 생성자로 작곡가(Composer)를 사용하고자 하나, 표준 DC 스키마에서는 Creator만을 제공하고 있다. 이 경우, 표준 메타데이터 요소 Creator와 응용 도메인의 Composer 간 상호 운용성이 제공되어야 한다. 이 밖에도 표준 메타데이터 스키마와 응용 도메인 요구 간에는 여러 형태의 상호 운용성이 제공되어야 하는데, 본 논문에서는 이를 수직적 상호호환성(vertical interoperability)로 정의한다. 수직적 상호 운용성은 표준 메타데이터 스키마

와 응용 프로파일의 수직적 관계에서 발생하는 상호 운용성으로 다음과 같은 내용을 포함한다.

- 메타데이터 어휘의 재명명
- 메타데이터 요소간의 계층관계
- 메타데이터 요소 값의 형태(type) 및 범위(range)
- 응용도메인에서 메타데이터 요소의 제약성

한편, 응용 프로파일에 혼합되는 네임스페이스 간에도 상호 운용성이 필요하다. 예를 들어 DC의 DC.Identifier와 IEEE LOM의 General.Identity.Entry 사이에는 상호 운용성이 존재한다<sup>9)</sup>. 응용 프로파일은 네임스페이스를 중심으로 구성되므로 이와 같은 메타데이터 스키마간의 상호 운용성에는 직접적인 영향을 받지 않는다. 그러나, 이러한 운용성의 보장은 응용 프로파일의 유용성을 높이고 메타데이터 하베스팅(harvesting)에 필요하다. 본 논문에서는 메타데이터 스키마간의 상호 운용성을 수평적 상호 운용성(horizontal interoperability)으로 정의한다.

앞서 고찰한 바와 같이 응용 프로파일은 응용 도메인의 정보자원의 특성을 고려하여 수직적 상호 운용성과 수평적 상호 운용성을 실현할 수 있는 체제를 갖추어야 한다. 본 논문에서는 응용 프로파일 코어 온톨로지로서 이를 실현한다.

7) 프로파일(profile)은 메타데이터를 효과적으로 활용하기 위한 방안으로 장치 프로파일인 CC/PP, 개인정보 프로파일인 P3P 등 다방면에서 활용되고 있다.

8) 네임스페이스는 XML 기술의 일반화와 더불어 정보 식별 기능으로 널리 사용되고 있으며 메타데이터 스키마 관리 및 통제 기관 식별, 메타데이터 요소의 고유 식별자, 특정 메타데이터 어휘의 고유 특성 식별 등 3가지 개념으로 이해할 수 있다.

9) 이와 같은 메타데이터 스키마 간의 상호 운용성은 개념 매핑(mapping) 또는 크로스워크(cross work)라고도 한다.

## 2.2 응용 프로파일 사례

메타데이터 상호 운용성을 효과적으로 구현하고자, DESIRE 프로젝트, SCHEMAS 프로젝트 등에서 응용 프로파일 개발 연구가 수행된 바 있다. 이러한 시도는 응용 프로파일의 개념을 정착시키는데 공헌하였고, 새로운 응용 프로파일 개발의 방향을 제시하고 있다. 본 절에서는 대표적 응용프로파일 사례를 분석하여, 응용프로파일 코어 온톨로지 개발의 토대를 제

공하고자 한다.

SCHEMAS 프로젝트(2003)에서 메타데이터 스키마 설계를 연구하면서, 메타데이터 상호 운용성을 실현할 응용 프로파일 개념이 제시되었다. 네임스페이스를 활용하여 메타데이터 요소를 재사용하고, 특정 목적에 일치하도록 메타데이터 요소를 패키지화하는 개념, 즉 응용 프로파일의 개념이 도출되었다. 또한 응용 프로파일이 가져야 할 4가지 필요조건도 정의하였다<sup>10)</sup>. 이후 SCHEMAS 프로젝트 관

〈표 1〉 BIBLINK 응용프로파일의 코어 어휘

요소명칭	RDF 프로퍼티	의미
Title	dc:title	주타이틀
TitleAlternate	bc:titleAlternate	부타이틀
Creator	dc:creator	생성자
Creator Organisation	bc:creatorOrganisation	생성책임기관
Contributor	dc:contributor	생성지원자
Contributor Organisation	bc:contributorOrganisation	생성지원자, 소속기관
Identifier	dc:identifier	정보자원 식별자(URI, DOI, ISBN,...)
Publisher	dc:publisher	출판책임자
Date	dc:date	생성 또는 가용일자(ISO8601형식)
Format	dc:format	정보자원형식(MIME)
Subject	dc:subject	정보자원 주제(LCSH, DDC,...)
Description	dc:description	정보자원 내용 기술
Language	dc:language	내용 언어
Rights	dc:rights	지적재산권
Source	dc:source	출처
Price	bc:price	가격
Extent	bc:extent	정보자원의 크기 등
Checksum	bc:checksum	체크섬 코드
Frequency	bc:frequency	발간 빈도
Edition	bc:edition	버전 또는 판
Place Publication	bc:placePublication	정보자원의 지리적 위치
System Requirement	bc:systemRequirement	하드웨어/소프트웨어 요구사항

10) 4가지 요구사항은 Heery와 Patel(2000)의 연구를 참고할 수 있다.

런 워크(Schemas Second Workshop 2000)에서 EULER, MathNet, DCMI Education, EIONET 등의 응용 프로파일이 도출되었다. 특히, 더블린 코어의 서지정보를 확장한 BIBLINK는 응용 프로파일의 대표적 사례로 널리 알려져 있다.

BIBLINK에서는 보다 상세한 서지정보를 표

현할 수 있도록 DC의 메타데이터를 기반으로 국소적 확장 요소를 추가하여 BIBLINK 코어 어휘(core vocabulary)를 도출하였다.<sup>11)</sup> BIBLINK(2001) 코어 어휘 구성은 다음과 같이 4종류의 어휘 요소로 구성되어 있다.

- DC 메타데이터 요소
- DC 메타데이터 요소의 상세화 요소

```

<smes:NsSchema
  rdf:about="http://www.schemas-forum.org/registry/schemas/BIBLINK/1.0/bc">
  <rdf:value>The BIBLINK Vocabulary v1.0</rdf:value>
  <dc:title>The BIBLINK Vocabulary v1.0</dc:title>
  <dc:date>2000-08-22</dc:date>

  <dc:identifier rdf:resource="http://www.schemas-forum.org/.../schemas/BIBLINK/1.0/bc" />

  <dc:publisher>The SCHEMAS Project</dc:publisher>
  <dc:format>RDF/XML</dc:format>

</smes:NsSchema>
<rdf:Property ID="creatorOrganisation">
  <rdfs:label>Creator Organisation</rdfs:label>
  <rdfs:comment>The organisation responsible for the intellectual content</rdfs:comment>
  <rdfs:isDefinedBy
    rdf:resource="http://www.schemas-forum.org/registry/schemas/BIBLINK/1.0/bc" />
</rdf:Property>

<rdf:Property ID="checksum">
  <rdfs:label>Checksum</rdfs:label>
  <rdfs:comment>A hash value computed for authentication purposes</rdfs:comment>
  <rdfs:isDefinedBy rdf:resource="http://www.schemas-forum.org/.../schemas/BIBLINK/1.0/bc"/>
</rdf:Property>

<rdf:Property ID="frequency">
  <rdfs:label>Frequency</rdfs:label>
  <rdfs:comment>The frequency of issue for serials</rdfs:comment>
  <rdfs:isDefinedBy
    rdf:resource="http://www.schemas-forum.org/registry/schemas/BIBLINK/1.0/bc" />
</rdf:Property>
  
```

〈그림 1〉 BIBLINK 코어 어휘의 RDF 스키마 기술

11) 응용프로파일 요구 사항 중 하나가 새로운 메타데이터 요소를 추가하지 않는 것이다. 새로운 메타데이터 요소를 추가하면, 상호 운용성과는 대립되기 때문이다. BIBLINK의 새로운 요소 추가는 응용 프로파일 개념이 확실하게 정착되지 않은 초창기에 개발되었기 때문일 것이다.

〈표 2〉 SCHEMAS의 응용 프로파일 정의 메타데이터 요소

요소 명 칭	RDF 프로퍼티	정 의
NsSchema	smes:NsSchema	네임스페이스
ApSchema	smes:ApSchema	응용 프로파일 스키마
isProfileOf	smes:isProfileOf	메타데이터 집합
uses	smes:uses	메타데이터 요소의 재사용
label	smes:label	응용 프로파일 레이블
definition	smes:definition	부가정보
comment	smes:comment	주석
usage	smes:usage	사용가이드라인

- DC와 관련된 구조
- BIBLINK 자체 정의 요소

BIBLINK 응용프로파일의 코어 어휘를 구체적으로 살펴보면 〈표 1〉과 같다.<sup>12)</sup> 〈표 1〉의 BIBLINK 코어 어휘는 구조적 호환성을 위해서 〈그림 1〉과 같이 RDF 스키마로 기술한다.<sup>13)</sup> 〈그림 1〉에서 보는 바와 같이 코어 어휘 creatorOrganisation, checksum 등은 단순히 어휘 명칭만 정의하고 있고 구체적인 의미특성을 정의하고 있지 못하다.

한편, SCHEMAS에서는 응용 프로파일로 정의하기 위해 〈표 2〉와 같이 별도의 메타데이터 요소를 정의하고 있다.<sup>14)</sup>

〈표 1〉이 서지 도메인의 일반 메타데이터 어휘를 정의하는데 비해, 〈표 2〉는 응용 프로파일을 정의하는 메타데이터 요소를 정의하고 있다. 실제로 〈표 2〉의 메타데이터 요소만이

응용 프로파일 스키마를 정의하는데 사용된다. 〈표 2〉를 살펴보면, uses로 사용하는 메타데이터 요소를 정의하는 것 외에는 상호 운용성 등을 위한 기능을 제공하고 있지 않다.

〈표 2〉의 메타데이터 요소로 정의된 응용 프로파일 스키마 예가 〈그림 2〉다. 〈그림 2〉의 응용 프로파일 스키마에서 DCMI의 title, creator, publish, subject와 BIBLINK의 코어 어휘인 titleAlternate, price 등이 응용 도메인의 정보 자원 기술에 사용됨을 정의하고 있다. DCMI의 subject 값의 범위는 BIBLINK의 SubjectScheme으로 한정된 것도 발견할 수 있다.

〈그림 2〉에 정의된 응용 프로파일 스키마에 의해 생성된 정보 자원 기술 사례가 〈그림 3〉이다. 응용 프로파일 스키마에서 정의된 네임스페이스와 어휘로 정보자원의 내용이 표현되

12) 표에서 4가지 종류 어휘를 네임스페이스, 의미를 참조하여 구분할 수 있다.

13) BIBLINK 코어 어휘를 XML 스키마로 표현할 수 있는데, 이에 대하여는 Hunter(2000)의 연구를 참고할 수 있다.

14) SCHEMAS에서는 General, Namespace and Application Profiles, Standards Watch, Metadata Watch, Person의 5가지 메타데이터 요소를 정의하고 있다.



```

<smes:ApSchema
  rdf:about="http://www.schemas-forum.org/registry/schemas/BIBLINK/1.0/bc-ap">

  <dc:title>The BIBLINK Application Profile v1.0</dc:title>
  <dc:date>2000-08-22</dc:date>
  <dc:publisher>The SCHEMAS Project</dc:publisher>
  <dc:format>RDF/XML</dc:format>
  <dc:type>Application Profile</dc:type>

  <smes:uses>
  <rdf:description rdf:about="http://purl.org/dc/elements/1.1/title">
    <smes:comment>The title of the publication</smes:comment>
  </rdf:description>
  </smes:uses>
  <smes:uses rdf:resource="http://www.schemas-forum.org/.../BIBLINK/1.0/bc#titleAlternate"/>
  <smes:uses rdf:resource="http://purl.org/dc/elements/1.1/creator" />
  <smes:uses rdf:about="http://purl.org/dc/elements/1.1/publisher" />
  <smes:uses>
  <rdf:description rdf:about="http://purl.org/dc/elements/1.1/subject">
    <rdfs:range rdf:resource="bc:SubjectScheme" />
    <rdfs:domain rdf:resource="http://purl.org/dc/elements/1.1/subject" />
  </rdf:description>
  </smes:uses>
  <smes:uses
    rdf:resource="http://www.schemas-forum.org/registry/schemas/BIBLINK/1.0/bc#price" />
</smes:ApSchema>

```

〈그림 2〉 BIBLINK의 응용 프로파일 스키마 예

어 있음을 알 수 있다.

이상에서 고찰한 BIBLINK의 응용 프로파일 접근 방식을 요약, 정리하면, 〈그림 4〉와 같다. 응용 프로파일이 여러 네임스페이스의 메타데이터를 혼합-일치하는 도구임에도 불구하고, BIBLINK에서는 응용 프로파일 구성을 위한 별도의 코어 어휘를 정의하고 있다. DCMI 스키마를 확장하고 있기 때문에 상호 운용성에도 문제가 발생할 수 있다. 즉, 응용 프로파일 구성시 수직, 수평적 상호 운용성을 실현할 수 있는 메타데이터 요소가 고려되고 있지 못하다. BIBLINK 응용 프로파일 분석을 통해,

응용 프로파일 스키마 구성과 활용에는 〈그림 4〉와 같은 계층모델이 존재함을 알 수 있다.

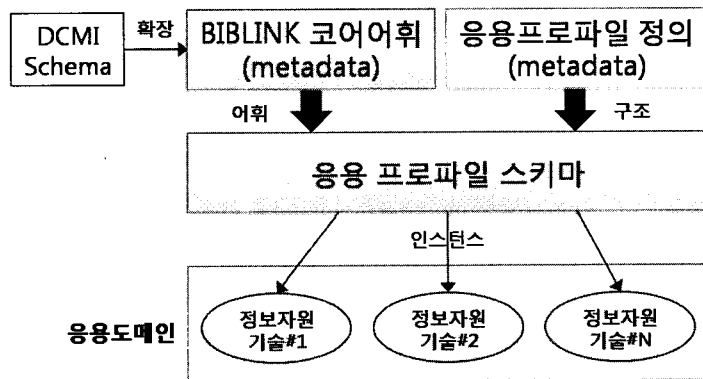
### 2.3 응용 프로파일 요구 기능과 계층구조

표준 메타데이터 스키마와 응용 프로파일 스키마의 상호 운용성을 실현하기 위해서는, 응용 프로파일 구성에 체계적인 접근 방식이 필요하다. 이에 따라, Heery and Patel(2000)은 응용 프로파일이 가져야 할 근본 특성을 4가지로 제시한 바 있다.

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/" xmlns:bc="http://www.schemas-
  forum.org/registry/schemas/BIBLINK/1.0/bc-ap#">
<rdf:Description about="urn:isbn:0-89887-113-1">
<dc:title>Patrologia Latina Database</dc:title>
<dc:creator>Jacques Paul Migne</dc:creator>
<dc:date>1993</dc:date>
<dc:language>la</dc:language>
<dc:format>CD-ROM</dc:format>
<bc:extent>2 computer laser optical disks; 4 3/4 in</bc:extent>
<dc:description>x The Patrologia Latina Database is an electronic version of the 221
  volumes of the first edition of Jacques-Paul Migne's Patrologia Latina which was
  published between 1844 and 1865.</dc:description>
<bc:systemRequirements>Multimedia PC 486x or higher, 8mb memory, CD-ROM
  drive, sound card, SVGA 256-colour monitor, Windows 95 or Windows
  3.1</bc:systemRequirements>
<dc:subject rdf:value="Christian literature, Early" bc:subjectScheme="LCSH" />
<dc:identifier rdf:value="isbn:0-89887-113-1" bc:identifierScheme="URN" />
<bc:placePublication>Cambridge</bc:placePublication>
<dc:publisher>Chadwyck-Healey</dc:publisher>
</rdf:Description>
</rdf:RDF>
  
```

〈그림 3〉 응용 프로파일 기반의 정보 자원 기술 예



〈그림 4〉 BIBLINK 응용 프로파일 구조

- ① 하나 또는 그 이상의 네임스페이스와 연 관될 수 있어야 한다.
- ② 새로운 메타데이터 요소를 도입하지 않는다.
- ③ 메타데이터 요소의 구조 특성(scheme)과 값을 지정할 수 있어야 한다.
- ④ 정의된 표준 메타데이터 요소를 상세화 할 수 있어야 한다.

①과 ②는 상호 운용성을 고려한 특성이며, ③과 ④는 특정 응용 도메인의 요구사항을 위한 특성이다. 특히, 상호 운용성 특성은 응용 프로파일에 반드시 필요한 원칙이다. NISO에서는 상호 운용성 유지를 판정하기 위하여 9가지 기준을 제시하고 있다.<sup>15)</sup> 본 논문에서는 응용 프로파일의 필수 요건을 아래와 같이 세부적으로 고찰하여, 응용 프로파일 구축 방향을 모색한다.

· **응용 도메인 적합성**

응용 프로파일의 기본 목적은 응용 커뮤니티 또는 응용 도메인의 정보 자원 기술의 국소적 요구 사항을 충족하기 위한 것이다. 응용 프로파일은 도메인 어휘와 의미 특성(semantic), 정보자원 관리를 위한 업무지침 등 응용 도메인이 필요로 하는 응용과 서비스의 기능적 요구사항을 충족하여야 한다.<sup>16)</sup> 그러므로 응용 프로파일은 다양한 응용 도메인에 적용할 수 있도록 충분한 형식 구조를 제공하여야 한다.

· **상호 운용성**

상호 운용성은 이용자가 여러 도메인 또는 기관에 분산되어 있는 정보자원을 검색하고 접근하는 데 일관된 방법을 제공할 뿐만 아니라, 정보 자원의 공유와 상호 교환 등을 지원한다. 응용 프로파일은 본 논문에서 제시한 수직, 수평적 상호 운용성의 기능을 제공할 수 있어야 한다.

상호 운용성을 실현하기 위해서는 데이터 요소와 한정사(qualifier) 등에 대한 개념 매핑 또는 크로스워크(crosswork) 등을 활용할 수 있다(Chan, Mai and Zeng 2006)(Godby et al. 2004). 응용 프로파일에서는 상호 운용성을 보장하기 위해 Open Archives Initiative(OAI)의 메타데이터 하베스팅(harvesting) 프로토콜(protocol)과 Z39.50의 메타검색 프로토콜을 지원하는 방안도 고려되어야 한다(Open Archives Initiative Protocol for Metadata Harvesting 2002).

· **상세화 및 특수화**

실제 메타데이터로 정보 자원을 기술하는 경우, 응용 도메인의 특성, 기능적 요구, 업무 특성 등에 따라, 정보자원의 기술 수준에 차이가 있다. 응용 도메인에서의 메타데이터 생성은 응용 도메인의 특수성을 표현하려는 경향이 있다. 응용 프로파일은 응용 도메인에서 요구되는 표현 상세화 및 특수화 요구를 해결할 수 있는 방안을 보유하여야 한다. DCMI에서도 이러한 문제 해결을 지원하고자 15개 데이터 요소에 대하여 기술의 풍부성과 명료성을 위해 한정사(qualifier)를 도입하여 다음과 같은 상세화 기능을 제공하고 있다.

- Date Accepted : “2006-7-8”
  - Language : “en” encoded in RFC 1766
- 첫 번째 방법은 “Date”를 한정사 “Accepted”

15) 윤세진(2002)은 이를 설계(design), 구조(organization), 변환(conversion), 조화(harmonization)와 용어(term)로 다시 정리하였다.  
 16) 응용 도메인마다 설계 또는 관리 지침을 표준화하는 경향이 있다. 기록물 관리의 경우 ISO 15489와 ISO 23081 지침을 표준화하고 있다.

로 한정하는 방법이고, 두 번째 방법은 데이터 요소값인 “en”을 “encoded in RFC 1766”으로 한정하는 방법이다. DCMI의 상세화 기능 뿐 만아니라, 네임스페이스 선언, 데이터 요소의 카디널리티 조건(cardinality constraint), 데이터 요소 값(data element value)의 범위(range)와 형식(type) 지정 등 국소적 요구 사항을 효과적으로 표현할 수 있어야 한다.

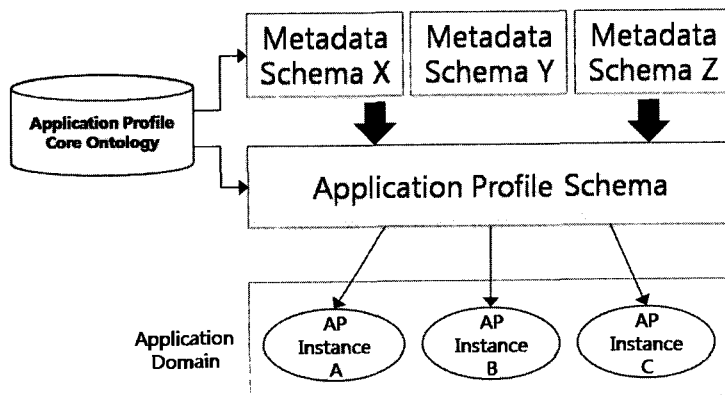
· 네임스페이스의 재사용성

응용 프로파일은 다수의 네임스페이스에 정의된 메타데이터 어휘를 조직적으로 수입(import) 할 수 있어야 한다. 메타데이터 스키마와 응용 프로파일 스키마는 수입을 통해 정의된 어휘를 재사용하는데 목적이 있으므로, 상호 운용성 등을 고려하여 새로운 어휘를 정의하지 않는다. 즉, 응용 프로파일은 새로운 어휘를 정의하는 기능을 가질 필요가 없다.

· 간편성 및 표현 기술의 중립성

단순 더블린코어의 경우, 15개의 데이터 요소, 특정값 어휘에 대한 미지정, “모든 데이터 요소는 optional이고, repeatable이다”는 규칙 집합 등의 극히 단순한 메타데이터 스키마 구조를 갖고 있다. DCMI가 지향하는 간편성으로 인하여 DCMES는 문헌정보 뿐 만 아니라, 박물관, 음악 등 50여 분야로 확산될 수가 있었다. 응용 프로파일이 응용 도메인에서 널리 활용되기 위해서는 사용자가 쉽게 활용할 수 있도록 간편한 구조를 가져야 한다. 또한, 사용 언어와 중립적으로 구성되어, 사전에 특정 언어에 대한 지식이 없어도 사용할 수 있도록 하여야 한다.

응용 프로파일이 위에서 논의한 요구 기능을 만족하려면, 응용 프로파일 활용 체계는 계층적 프레임워크를 가지게 된다. 워릭 프레임워크에서도 이러한 구조가 제시되었으며, 본 논문에서 분석한 응용 프로파일의 대표적인 사례인 BIBLINK에서도 응용 프로파일 활용의 계층적 프레임워크 구조를 발견할 수 있었다. 응



〈그림 5〉 응용 프로파일 계층 모델

용 프로파일 중심의 체계화된 계층구조에 대한 연구도 수행되었다(Sugmoto 2005)(한성국 2007). 제시된 계층 프레임워크 모델에서는 계층의 역할과 기능이 분명하게 정의되어 응용 프로파일 개발과 활용에 도움을 주고 있다. 그러나 응용 프로파일의 구성 요건을 수용하는데 미흡하고, 여러 응용 도메인에 적용할 수 있는 표준적 응용 프로파일 구성을 제시하고 있지 못하다. 본 논문에서는 응용 프로파일의 특성과 구성 요건을 고려하여 <그림 5>와 같은 계층 모델을 도출하였다.

응용 프로파일 계층모델의 핵심 구성 요소는 응용 프로파일 코어 온톨로지(application profile core ontology: APO)이다. APO는 메타데이터 요소의 어휘, 어휘 특성, 어휘 값의 형식과 범위 등을 정의할 수 있고, 네임스페이스와 스키마 관리 등 메타데이터와 응용 프로파일에서 요구되는 기능을 표준적인 방식으로 제공한다. 응용 프로파일은 APO에 의해서 생성되고, 응용 프로파일에 준거하여 인스턴스(instance)를 작성한다.

### 3. 응용프로파일 코어 온톨로지 설계

응용 프로파일을 응용 도메인과 표현 언어 중립적으로 기술하기 위해 표준 코어 온톨로지의 개념 구조를 도출하고, 코어 온톨로지 어휘 요소를 정의한다.

#### 3.1 응용 프로파일 코어 온톨로지 설계

응용 프로파일은 <그림 5>에서 보는 바와 같이 중간 연결 계층의 역할을 담당하고 있기 때문에, 표준 메타데이터와의 인터페이스, 응용 도메인과의 인터페이스 양 측면에서 인터페이스가 필요하다. 제2장에서 응용 프로파일이 표준 메타데이터와 응용 도메인의 정보 자원 기술 요구 사항간의 인터페이스를 위해 제공하여야 할 구조와 기능 특성을 서술하였다. 응용 프로파일의 코어 온톨로지 도출을 위해 세부 내용을 아래와 같이 분석한다.

##### · 표준 메타데이터 스키마와 인터페이스

응용 프로파일은 다수 메타데이터 스키마의 메타데이터 요소로 구성된다. 때문에 메타데이터 네임스페이스를 적절하게 관리하는 것이 필수적이다. 수입된 네임스페이스내의 메타데이터 요소간의 개념 매핑이나 크로스워크드 기술할 수 있어야 응용 프로파일의 가용성을 높일 수 있다.

##### · 응용 도메인과 인터페이스

응용 도메인에서는 메타데이터 요소를 상세화 또는 특수화하게 된다. 상세화의 유형과 이에 필요한 코어 온톨로지를 세부적으로 고찰하면 다음과 같다.

##### · 도메인 어휘의 재명명

음악 도메인에서는 DCMI의 Creator보다는 명시적인 Composer를 사용하는 것이 좋다. 이 경우 상위 어휘와의 계층 관계, 즉 subClassOf의 관계가 명시적으로 표출되어

야 상호 운용성을 유지할 수 있다.

· 메타데이터 요소의 선택성

응용 도메인마다 필수 어휘를 지정할 수 있고, 부가적 선택 어휘를 정의할 수 있다. 표준 메타데이터 스키마에서는 이러한 선택 특성을 정의하지 않지만, 응용 도메인의 메타데이터 요소에는 중요한 역할을 하고 있다. 선택 특성에는 mandatory, mandatory-if-applicable, optimal, required 등 여러 종류가 있다 (Ratanajaipan, Nantajeewarawat, and Wuwongse 2006b).

· 메타데이터 요소의 카디널리티 (cardinality)

카디널리티는 유효한 문서를 작성하는데 필요한 기능이다<sup>17)</sup>. 온톨로지 표준언어 OWL에 서는 cardinality, minCardinality, maxCardinality 3종류의 카디널리티 정의 방식을 제공한다.

· 메타데이터 요소 값 제한

부가적 메타데이터 요소의 경우에는, 문서의 유효성 보장이나 컴퓨터 처리 효율성을 고려하여, 메타데이터 요소값의 형식과 범위를 응용 도메인의 특성에 따라 제한하게 된다. 예를 들어, 응용 도메인에서 DCMI의 Date는 ISO 8601 또는 W3C-DTF 형식으로 기록하며, IEEE LOM의 Location은 최대 10자의 문자열을 제한할 수 있다. 메타데이터 요소값의 형태, 크기, 범위에 대한 제한은 응용 도메인

상세화에 기본 요구 사항이다.

· 어휘 구조화

응용 도메인마다 고유하게 사용하는 데이터 구조가 있다. 고유 데이터 구조는 주로 메타데이터 요소값 제한에 적용된다. 예를 들어, DCMI의 Organization을 (LOC, OCLC, DTSC)로 한정할 수 있는데, 이때 응용 프로파일에는 이 데이터 구조를 정의할 수 있어야 한다. 메타데이터 요소의 개념 매핑에서도 이러한 구조화가 필요하다. 예를 들어 Agent와 동의어 개념으로 {Actor, Contributor, Player, Worker, Performer}를 정의할 때 어휘 구조화가 필요하다. 어휘 구조화는 데이터값 또는 어휘 집합체로서, 순서를 중요한 정보로 사용하는 경우가 있으므로 이를 고려하여야 한다.

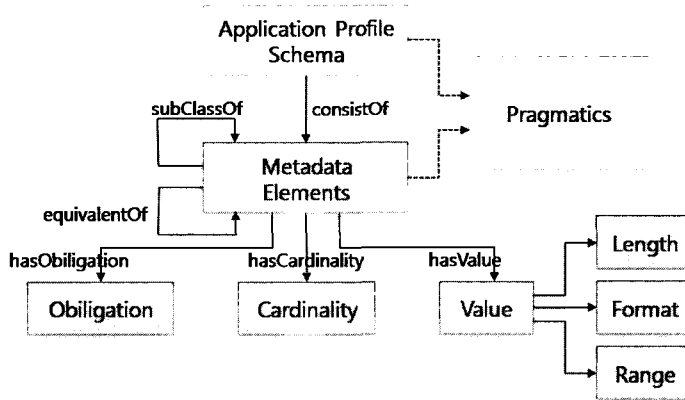
응용 프로파일 기능 특성 분석에서 도출된 응용 프로파일 코어 온톨로지의 개념 어휘간의 관계는 <그림 6>으로 표현된다.<sup>18)</sup> <그림 6>으로 응용프로파일 스키마를 구성할 코어 온톨로지를 체계적으로 정의할 수 있다.

3.2 응용 프로파일 코어 온톨로지

APO는 응용 프로파일 스키마를 구성할 메타데이터 요소를 정의하는데 사용된다. APO의 세부 온톨로지 어휘를 개발하기 위해서는 기존 메타데이터 스키마에서 메타데이터 요소를 정의할 때 사용하는 어휘체계를 참조할 필요가 있다. 본 논문에서는 대표적인 메타데이

17) 여기서 유효한 문서(valid document)는 응용 프로파일 스키마에 준거하여 생성된 인스턴스를 의미한다.

18) pragmatics는 부가적 어휘 요소의 집합으로 3.2절에서 상세하게 설명한다.



〈그림 6〉 응용 프로파일 코어 온톨로지 개념 관계

터 스키마인 DC, IEEE LOM, 시소러스 표준 스키마인 SKOS의 메타데이터 요소 정의 어휘를 응용 프로파일 관점에서 분석하고, 〈그림 6〉의 개념 모델과 관련하여 응용 프로파일 스키마 어휘를 정의한다.

· DC 스키마 정의 어휘

DCMI(2006)에서는 〈표 3〉과 같이 메타데이터 요소의 속성을 정의하는 기초 어휘와 보조 어휘를 각각 7개씩 사용하고 있다.

DCMI 어휘는 메타데이터 요소의 기본 특성과 관리 정보 정의 어휘로 구성되어 있다. 응용 프로파일은 이미 표준화된 메타데이터 요소를 대상으로 하므로 표준화 관련 정보는 DC 자체를 활용할 수 있다.

· IEEE LOM 스키마 정의 어휘

IEEE LOM(2002)에서는 계층화된 70여 개의 메타데이터 요소를 간명하게 정의하고 있다. IEEE LOM의 요소 정의 어휘는 〈표 4〉

〈표 3〉 DCMI의 스키마 정의 어휘

최소 속성 정의 어휘		보조 속성 정의 어휘	
어휘	의미	어휘	의미
Name	용어의 명칭	Comment	사용 부가 정보
URI	정의된 용어의 네임스페이스	See	연관 관리 문서
Label	사용자를 위한 명칭	Reference	참조된 정보자원
Definition	용어 의미 서술	Refines	재정의 요소
Type of Term	용어의 형식	Qualifies	한정어
Status	표준화 진행 상태	Broader Term	연관 광의어
Date Issued	용어 선정 일자	Narrower Term	연관 협의어

〈표 4〉 IEEE LOM의 스키마 정의 어휘

어휘	의미	어휘	의미
name	요소의 명칭	explanation	요소에 대한 설명
size	허용된 요소값의 수	order	요소값의 순서
example	사용예	value space	허용값의 집합
data type	요소값의 형식		

와 같다.

IEEE LOM에서는 작성된 문서의 검증(validation)을 통하여 상호 운용성을 보장할 목적으로 요소 값의 크기, 순서, 형식을 중요시하고 있다. 이러한 요소값의 상세화는 응용 프로파일의 상세화 특성과도 부합된다.

· SKOS 스키마 정의 어휘

SKOS 시소러스 용어의 BT, NT, RT 등의 계층 관계를 정의할 수 있는 다양한 어휘를 사용하고 있다. SKOS의 스키마 정의 어휘(W3C

2005)는 〈표 5〉와 같다.

SKOS의 스키마 정의에서는 〈표 5〉의 정의 어휘 이외에도 내부적으로 Concept, preLabel, altLabel, hiddenLabel, semanticRelation, broader, narrower, note, collection, orderedCollection, member, memberList 등의 다양한 어휘가 사용되고 있다. 이는 시소러스 용어 간의 관계를 명확하게 정의하고 연관 용어 집합을 정형화하는데 사용된다.<sup>19)</sup>

대표적인 메타데이터 스키마에서 사용된 어휘 분석으로 응용 프로파일에서 사용될 어휘들

〈표 5〉 SKOS의 스키마 정의 어휘

어휘	의미	어휘	의미
URI	식별자	Label	사용자를 위한 명칭
Definition	어휘 의미 설명	Comment	부가적 용어 의미
Example	사용예	Status	표준화 상태
Issued	정의된 일자	Modified	마지막 수정 일자
Super-classes	상위 클래스	Super-properties	상위 프로퍼티
Domain	프로퍼티의 도메인	Range	프로퍼티의 범위
Characteristics	프로퍼티의 함수 특성	Inverse of	역관계 프로퍼티
Replaces	사용되었던 퇴출 용어	Version info	버전정보
Replaced By	대용어	Deprecated	퇴출용어

19) 자세한 세부 어휘는 SKOS 스키마를 참고할 수 있다.



〈표 6〉 응용 프로파일 코어 온톨로지 어휘

어휘	의미	어휘	의미
· Profile Pragmatics : 응용 프로파일 기본 정보 기술 어휘			
Name	명칭	Identifier	식별자
Creator	작성자	Description	설명
SubjectDomain	적용되는 응용 도메인	DateCreated	생성일자
DateModified	최종 수정일자	Language	사용언어
Version	버전	Comment	부가정보
· Elements Pragmatics : 메타데이터 요소 관련 기본 정보 기술 어휘			
MetadataElement	요소명칭	Namespace	요소 네임스페이스
Label	사용자 명칭	Definition	요소 정의 설명
DateImported	차용일자	SeeAlso	참고요소
Example	사용예	Type	메타데이터 요소 분류
· Element Definition : 메타데이터 요소 관련 정보			
Superclass	상위 클래스 요소	Synonym	동의어 요소
Obligation	요소 선정 조건	Cardinality	카디널리티
Unit	요소값의 단위	Format	요소값의 형식
Range	요소값의 범위	Length	요소값의 길이
Collection	요소값 집합	Ordered Collection	순서 있는 요소값 집합

을 보다 분명하게 식별할 수 있다. 〈그림 6〉의 응용 프로파일 코어 온톨로지의 개념 구조를 고려하면 〈표 6〉과 같은 핵심 어휘가 도출된다. 대표적인 메타데이터 스키마 정의 어휘에서 알 수 있는 바와 같이, 스키마 기본 정보 기술에 여러 어휘가 사용되고 있다. 이를 화용 (pragmatics) 어휘<sup>20)</sup>라고 하며, APO에서는 응용 프로파일 자체를 위한 화용 어휘와 수입된 메타데이터 요소의 화용 어휘 두 종류가 있다. 각 화용 어휘는 기본 정보를 기술하는 관

점에서 유사하나, 적용 대상에 차이가 있어 별도로 구성하였다. 특히 다수의 메타데이터 스키마에서 수입된 메타데이터 요소의 네임스페이스를 Namespace로 표현하여 메타데이터 요소를 응용 도메인의 요구에 적합하도록 상세화하는 여러 어휘를 제공한다. 요소 선택 제한, 카디널리티, 요소값과 관련한 상세화를 제공한다. 메타데이터 요소 상세화와 관련된 어휘는 MetadataElement의 프로퍼티를 정의 하는데 사용된다.<sup>21)</sup>

20) UPML의 메타데이터 요소 정의를 참고하였다.

21) 메타데이터 상세화 관련 어휘와 함께 사용되는 Mandatory, Optional 등의 보조 어휘를 〈표 6〉에서는 포함 시키지 않았다.

## 4. 응용 프로파일 코어 온톨로지 구현

메타데이터 응용 시스템을 고려하여, APO를 RDF/RDF 스키마로 구현한다. 응용 프로파일 스키마 개발자와 메타데이터 문서 작성자가 네임스페이스로부터 독립될 수 있도록 구현한다.

### 4.1 응용 프로파일 코어 온톨로지의 RDF 구현

메타데이터 언어로 RDF/RDF 스키마가 표준화 되어 있다. RDF가 정보 자원간의 의미 관계를 효과적으로 표현할 수 있어 화용 어휘 개념을 명확하게 정의하는데 유용하다. 반면에, 카디널리티, 선택 제한, 요소값의 제한 등 메타데이터 상세화에서 요구되는 어휘 정의에는 RDF가 미흡한 것으로 간주되고 있다. 그러나, 메타데이터의 어휘 개념 정의는 선언적 특성을 갖고, 있기 때문에, 메타데이터를 활용하는 체계에서 일관된 의미 해석을 부여하면 해결된다.<sup>22)</sup> APO는 주로 개발자가 응용 프로파일 스키마를 정의하는데 사용하며, 응용 프로파일 스키마는 메타데이터 문서 작성의 지침과 응용 시스템에서 검증 등에 사용되므로 일관된 의미 해석이 가능하다.

APO를 RDF 스키마로 구현한 일부를 <그림 7>에 보였다.<sup>23)</sup> 실제 APO 구현에는 <표 6>

에서 제시된 코어 온톨로지 이외에 구현에 필요한 여러 어휘가 사용된다. <그림 7> 구현에서는 메타데이터 요소 상세화와 관련된 어휘, Obligation, Synonym, Cardinality 등의 구현을 보였다. Obligation 정의를 위한 Mandatory, Optional, Required 등의 어휘 정의를 볼 수 있다.

응용 프로파일 정의와 관련된 모든 어휘가 APO에 체계적으로 정의되어 있기 때문에, 응용 프로파일 스키마를 효과적으로 생성할 수가 있다. 또한 생성된 응용 프로파일 스키마로 저장고를 구축하는 것도 가능하다. RDF 관련 응용 기술이 보편화되어 있어 APO를 활용한 메타데이터 응용 시스템 개발을 용이하게 접근할 수 있다.

### 4.2 응용 프로파일 스키마 사례

응용 도메인에서 표준 메타데이터 스키마를 활용하면, <그림 8>과 같이 APO로 응용 프로파일 스키마를 작성할 수 있다. 응용 프로파일 스키마는 응용 도메인의 요구 사항을 분석하여, 적합한 메타데이터 요소를 추출하고 이를 표준 메타데이터와 연계하여 상세화 등을 실시하여 작성하여야 한다.<sup>24)</sup>

본 논문에서는 APO의 유용성과 작성된 응용 프로파일 스키마의 특징을 보이기 위해 오페라와 관련된 동영상 DVD 도메인의 간단한

22) 온톨로지 표준 언어인 OWL의 어휘도 RDF로 정의하고 있다(OWL Web Ontology Language Reference <<http://www.w3.org/TR/owl-ref>> 참고).

23) <http://sws.wonkwang.ac.kr/2007/07/apo/core/>를 참고한다.

24) 본 논문에서는 응용 프로파일 스키마 설계 방법에 대하여 구체적으로 다루지 않는다. 관련 논문(Duval, Smith, and Coillie 2006)을 참고한다.

```

<rdfs:Class rdf:ID="ApplicationProfileCoreOntology">
  <rdfs:label xml:lang="en">Application Profile Core Ontology </rdfs:label>
  <rdfs:isDefinedBy rdf:resource="http://sws.wonkwang.ac.kr/2007/07/apo/core"/>
</rdfs:Class>

<!-- Fundamental classes -->
<rdfs:Class rdf:ID="Schema">
  <rdfs:label>Metadata Schema</rdfs:label>
  <rdfs:comment>An Metadata schema</rdfs:comment>
  <rdfs:subClassOf rdf:resource="http://dublincore.org/2000/03/13/dctype#Text" />
</rdfs:Class>

<rdfs:Class rdf:ID="ApplicationProfile">
  <rdfs:label>Application Profile Schema</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Schema" />
  <rdfs:isDefinedBy rdf:resource="http://sws.wonkwang.ac.kr/2007/07/apo/core" />
</rdfs:Class>

<rdfs:Class rdf:ID="MetadataElement">
  <rdfs:label xml:lang="en">Metadata Element</rdfs:label>
  <apo:definition xml:lang="en">Metadata Elements from various Schema</apo:definition>
</rdfs:Class>

<!-- Auxiliary classes -->
<rdfs:Class rdf:ID="ObligationType">
  <rdfs:comment>A set of enumerated types for obligation specification</rdfs:comment>
</rdfs:Class>
<ObligationType rdf:ID="Mandatory" />
<ObligationType rdf:ID="Optional" />
<ObligationType rdf:ID="Required" />

<!-- Basic properties of Metadata Elements -->

<rdf:Property rdf:ID="Superclass">
  <rdfs:label xml:lang="en">superclass</rdfs:label>
  <rdfs:domain rdf:resource="#MetadataElement"/>
  <rdfs:range rdf:resource="#MetadataElement"/>
  <apo:definition xml:lang="en">A SuperClass defines an upper metadata element.</apo:definition>
</rdf:Property>

<rdf:Property rdf:ID="Obligation">
  <rdfs:domain rdf:resource="#MetadataElement" />
  <rdfs:range rdf:resource="#ObligationType" />
</rdf:Property>

<rdf:Property rdf:ID="Synonym">
  <rdfs:domain rdf:resource="#MetadataElement" />
  <rdfs:range rdf:resource="#MatadataElement" />
</rdf:Property>

<rdf:Property rdf:ID="Cardinality">
  <rdfs:domain rdf:resource="#MetadataElement" />
  <rdfs:range rdf:resource="&xsd:nonNegativeInteger" />
</rdf:Property>

```

〈그림 7〉 응용 프로코어 온톨로지의 RDF 구현(일부)

예를 보인다. APO로 응용 프로파일 스키마를 작성하면 <그림 8>과 같다.<sup>25)</sup> APO로 작성된 응용 프로파일 스키마는 <그림 2>의 BIBLINK

스키마와 비교하였을 때, 다음과 같은 특징이 있다. 첫째, 메타데이터 네임스페이스가 APO로 단

```

<apo:ApplicationSchema rdf:about="http://sws.wonkwang.ac.kr/.../sample">
  <apo:APName> Sample Application Profile Schema </apo:APName>
  <apo:APCreator> Hyun-Sil Lee </apo:APCreator>
  <apo:APDescription> Example of Application Profile Schema </apo:APdescription>
  <apo:APIdentifier> OperaSchema </apo:APIdentifier>

  <apo:MetadataElement rdf:ID="Title">
    <apo:Description> Opera DVD Title </apo:Description>
    <apo:Namespace rdf:resource="http://purl.org/dc/elements/1.1/title" />
    <apo:Obiligation> Madatory </apo:Obiligation>
  </apo:MetadataElement>

  <! ... 전체 응용 프로파일 스키마는 sws.wonkwang.ac.kr/2007/07/apo/core를 참고...>

  <apo:MetadataElement rdf:ID="Composer">
    <apo:Comment> Full name of composer </apo:Comment>
    <apo:Namespace rdf:resource="http://purl.org/dc/elements/1.1/creator" />
    <apo:Obiligation> Required </apo:Obiligation>
  </apo:MetadataElement>

  <apo:MetadataElement rdf:ID="Cost">
    <apo:Description> Purchasing cost </apo:Description>
    <apo:Namespace rdf:resource="http://www.....org/.../bc#price" />
    <apo:Unit> Won </apo:Unit>
  </apo:MetadataElement>

  <apo:MetadataElement rdf:ID="PerformanceDate">
    <apo:Description> Date of Performance </apo:Description>
    <apo:Namespace rdf:resource="http://purl.org/dc/elements/1.1/date" />
    <apo:Obiligation> Required </apo:Obiligation>
  </apo:MetadataElement>

  <apo:MetadataElement rdf:ID="Duration">
    <apo:Description> Overall playing time of opera </apo:Description>
    <apo:Namespace rdf:resource="http://ltsc.ieee.org/wg12/.../Duration" />
    <apo:DateCreated> 2007-07-07 </apo:DateCreated>
  </apo:MetadataElement>

  <apo:MetadataElement rdf:ID="PerformancePlace">
    <apo:minCardinality> 2 </apo:minCardinality>
    <apo:Description> Place or theater where opera performed </apo:Description>
    <apo:Namespace rdf:resource="http://www.metadata..../location/place" />
  </apo:MetadataElement>

</apo:ApplicationSchema>

```

<그림 8> APO에 의해 작성된 응용 프로파일 스키마 예(일부)

일화되어 있다. 응용 프로파일 스키마는 다수의 메타데이터 네임스페이스로부터 어휘를 차용하기 때문에 네임스페이스 관리는 어휘 식별에 중요하다. 기존의 응용 프로파일 스키마는 개별적인 네임스페이스를 그대로 활용하기 때문에, 스키마 개발자와 이를 사용하는 이용자 모두 다수의 메타데이터 스키마에 종속되었다. APO에서는 네임스페이스와 독립적으로 응용 프로파일 스키마를 작성할 수 있다. 둘째, 메타데이터 요소를 모듈 형식으로 재사용하고 있다. 응용 시스템 개발자는 템플레이트 형식으

로 응용 프로파일 스키마를 생성할 수 있다. 셋째, 응용 도메인에서 요구하는 상세화를 명시적으로 정의할 수 있다. 예를 들어, Obligation, minCardinality, Unit 등 메타데이터 요소의 속성을 정의하고 있다.

이 밖에도 메타데이터 요소의 계층구조, 개념 매핑 등을 정의할 수 있다. 2.3절에서 논의한 응용 프로파일 요구 기능을 이해하기 쉽게 실현하고 있다. 응용 프로파일 스키마 사례로부터 응용 프로파일 코어 온톨로지의 유용성을 확인할 수 있다.

```

<!-- Instance Example of Application Profile Schema -->
<rdf:Description rdf:about="wonkwang2345">
  <apo:Creator> Hyun-Sil Lee </apo:Creator>
  <apo:DateCreated> 2007-07-07 </apo:DateCreated>
  <apo:Version> 2.0 </apo:Version>

  <apo:Title> Tosca Opera </apo:Title>
  <apo:Composer> Giacomo Puccini </apo:Composer>
  <apo:Medium> video recording </apo:Medium>
  <apo:Series> Golden Label </apo:Series>
  <apo:Hours>123 </apo:Hours>
  <apo:Price> 22000 </apo:Price>
  <apo:Language> Italian </apo:Language>
  <apo:Publisher> Ilsong Media </apo:Publisher>
  <apo:Conductor>Erede, Alberto </apo:Conductor>
  <apo:Director>Butler, Peter </apo:Director>
  <apo:Audience> Special audience characteristics </apo:Audience>
  <apo:Player> Marton, Eva </apo:Player>
  <apo:Distribution> Spectrum DVD </apo:Distribution>
  <apo:DateOfDistribution>2004</apo:DateOfDistribution>
  <apo:Format> DVD </apo:Format>
  <apo:Size> 4 3/4 in </apo:Size>
  <apo:DatePublished> 2002-10-4</apo:DatePublished>
  <apo:SystemRequirement>DVD Player, Area Code 3, NTSC</apo:SystemRequirement>
</rdf:Decsription>

```

〈그림 9〉 응용 프로파일 스키마에서 생성된 문서의 예

〈그림 8〉로 정의된 응용 프로파일 스키마로 정보자원의 의미 특성을 기술하게 된다. 즉, 응용 도메인에서 메타데이터로 주석된 스키마 인스턴스(instance) 문서를 생성한다. 스키마 인스턴스 문서는 일반적으로 정보자원의 유지 관리를 담당하는 사용자가 작성하게 된다. 이때, 〈그림 8〉의 응용 프로파일 스키마는 인스턴스 문서 작성의 지침서로서의 역할을 하게 된다. 〈그림 9〉에 응용 프로파일 스키마로 작성한 인스턴스 문서의 예를 보였다.

〈그림 9〉 문서의 예는 쉽게 작성하고 이해할 수 있는 형태임을 보여주고 있다. 〈그림 3〉의 BIBLINK 예 등과 비교하여 복잡한 네임스페이스 체제와 거기에 종속된 어휘를 사용하지 않아도 되며, 문서 구조가 간결하다. 스키마 인스턴스 문서 작성자는 응용 프로파일에서 정의된 스키마 어휘와 어휘 특성에 따라, 인스턴스 문서를 작성할 수 있다.<sup>26)</sup> 응용 프로파일 스키마가 APO 어휘로만 작성되기 때문에, 일반적으로 사용 가능한 인스턴스 문서 편집기를 개발하여, 사용자의 인스턴스 문서 작성을 지원할 수 있다. 응용 프로파일 코어 온톨로지 APO는 메타데이터 기반 응용 시스템 구축에 필수적인 어휘 체제를 형성하고 있다.

## 5. 결 론

컴퓨터와 정보통신 기술의 급속한 발전에 따라 정보자원의 디지털화와 관리에도 새로운 체계가 마련되고 있다. 기존의 정보 시스템에서

는 MARC를 중심으로 단행본이나 정기간행물 등의 정보자원을 기술하였다. 최근에는 전자책, 웹정보, 동영상 등 다양한 디지털 정보자원이 보편화되고 있기 때문에, 정보자원의 형식과 의미 기술에 새로운 체계가 요구되고 있다. 메타데이터는 다양한 정보자원의 특성을 간단하고 편리한 방식으로 표현할 수 있어 서지정보 기술의 새로운 체계로 발전하고 있다. 또한 많은 메타데이터 스키마가 개발되어 정보자원의 메타데이터 기술을 주도하고 있다.

메타데이터를 효과적으로 활용하려면 다수의 메타데이터 요소를 혼합-일치 방식으로 합성하여 재사용하여야 한다. 이때, 메타데이터 요소간의 상호 운용성, 요소의 상세화 등의 문제가 있다. 메타데이터 재사용에서 발생하는 이러한 문제 해결에 응용 프로파일 접근 방식이 이용되고 있다. 그러나 정형화된 어휘체계와 응용 프로파일 기술 방식이 개발되어 있지 않아, 상호 운용성과 상세화 등의 목적을 실현하고 있지 못하며, 정보의 공유도 이루어지고 않고 있다. 본 논문에서는 표준적 응용 프로파일에서 요구되는 일반적 사항을 분석하여 응용 프로파일 코어 온톨로지를 APO를 제시하였다.

먼저, 응용 프로파일에서 요구되어지는 필수 기능을 분석하고, 응용 프로파일 중심의 메타데이터 응용 계층 모델을 유도하였다. 기존의 다양한 메타데이터 스키마에서 사용해 온 메타데이터 요소 기술의 어휘 체계를 분석하여, 응용 프로파일의 목적과 기능을 실현할 수 있는 코어 온톨로지 APO를 제시하였다. 다음으로, 표준 언어인 RDF/RDF 스키마로 이를 실제

26) 〈그림 9〉에서 사용된 Title, Composer, Medium 등은 응용 프로파일 스키마가 여러 메타데이터 스키마에서 수입된 어휘이다.

구현하였고, 응용 프로파일 스키마 사례를 보였다.

본 논문의 응용 프로파일 스키마 사례에서 APO로 오페라 DVD를 기술한 것처럼, 의미 기반의 응용 프로파일 코어 온톨로지 적용으로 기존의 도서관 서지정보 기술은 큰 진보를 가져올 수 있다. APO를 활용한 정보자원 기술은 기존의 기술방식에 비하여 다양한 메타데이터를 활용할 수 있으므로 풍부한 의미 표현은

물론 정보기술의 공유와 상호 호환성을 높일 수 있는 방법이다.

끝으로, 메타데이터가 정보자원의 서지정보 표현과 공유에 표준 어휘인 것처럼, 응용 프로파일 표현과 공유에 표준 어휘체계가 필요하다. 응용 프로파일 코어 온톨로지가 널리 활용되기 위해서는 APO 기반 응용 시스템 구축에 필요한 편집기와 저장고 관리 등의 도구가 개발되어야 할 것이다.

## 참 고 문 헌

- 윤세진. 2002. 메타데이터간 상호 운용성을 위한 비교 연구: MARC, Dublin Core, ONIX 메타데이터요소를 중심으로. 『한국도서관·정보학회지』, 33(2): 276-301.
- 한성국, 이현실. 2007. 시맨틱 라이브러리를 위한 아키텍처 참조 모델. 『정보관리학회지』, 24(1): 75-101.
- Bulterman, Dick. 2004. "Is it Time for a Moratorium on Metadata." *IEEE multimedia*, 2004(Oct-Dec): 10-17.
- BIBLINK CORE Metadata Set. 2001. [cited 2007. 8. 14]. <<http://www.schemas-forum.org/registry/schemas/biblink/BC-schema.html>>.
- Chan, Lois Mai, and Marcia Lei Zeng. 2006. "Metadata Interoperability and Standardization - A Study of Methodology Part I: Achieving Interoperability at the Schema Level." *D-Lib Magazine*, 12(6). [cited 2007. 8. 14]. <<http://www.dlib.org/dlib/june06/chan/06chan.html>>.
- DCMI. 2006. DCMI Metadata Terms. [cited 2007. 8. 14]. <<http://dublincore.org/documents/dcmi-terms/>>.
- Duval, E., N. Smith, M. Van Coillie. 2006. "Application Profiles for Learning." *Sixth International Conference on Advanced Learning Technologies (ICALT'06)*, 05-07: 242-

- 246.
- Godby, Carol Jean, Jeffrey A. Young, and Eric Childress. 2004. "A Repository of Metadata Crosswalks." *D-Lib Magazine*, 10(12). [cited 2007. 8. 14]. <<http://www.dlib.org/dlib/december04/godby/12godby.html>>.
- Heery, Rachel and Manjula Patel. 2000. "Application profiles: mixing and matching metadata schemas." [cited 2007. 8. 14]. <<http://www.ariadne.ac.uk/issue25/app-profiles/>>.
- Hunter, Jane. 2000. "An XML Schema Approach to Application Profiles." [cited 2007. 8. 14]. <[http://www.itee.uq.edu.au/~jane/jane-hunter/harmony/appln\\_profiles.html](http://www.itee.uq.edu.au/~jane/jane-hunter/harmony/appln_profiles.html)>.
- Hunter, Jane and Carl Lagoze. 2001. "Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles." [cited 2007. 8. 14]. <<http://www10.org/cdrom/papers/572/>>.
- Maness, Jack M. 2006. "Library 2.0 Theory: Web 2.0 and Its Implications for Libraries." *Webology*, 3(2). [cited 2007. 8. 14]. <<http://www.webology.ir/2006/v3n2/a25.html>>.
- MetaMap. [cited 2006. 8. 14]. <<http://www.dlib.org/dlib/december04/godby/12godby.html>>.
- Open Archives Initiative Protocol for Metadata Harvesting, Protocol Version 2.0 of 2002-06-14. [cited 2006. 8. 14] <<http://www.openarchives.org/OAI/openarchivesprotocol.html>>.
- Ratanajaipan, P., E. Nantajeewarawat, V. Wuwongse. 2006a. "Representing and Reasoning with Application Profiles Based on OWL and OWL/XDD." *Lecture Notes in Computer Science (LNCS)*, 4185: 256-262. Berlin: Springer.
- \_\_\_\_\_. b. "OWL/XDD: A Formal Language for Application Profiles." *IEEE Conference on Cybernetics and Intelligent Systems*, June 2006 :1-6.
- Schemas Second Workshop. 2000. [cited 2007. 8. 14]. <<http://www.schemas-forum.org/workshops/ws2/report.htm>>.
- Schemas Registry. 2003. [cited 2007. 8. 14]. <<http://www.schemas-forum.org/registry/>>.
- Sugimoto, Shigeo. 2005 "Models and Requirements of Metadata: Metadata Projects at Tsukuba and Lesson Learned." [cited



2007. 8. 14]. <<http://avalon.slis.tsukuba.ac.jp/~sugimoto/Articles/NLTConferencePaper.pdf>>.

W3C. 2005. "SKOS Core Vokabular." [cited 2006. 8. 14]. <<http://www.w3.org/TR/2005/WD-swp-skos-core-spec-20051102/Overview.html.de/>>/