메타데이터의 상호운용성 보장을 위한 요소기술

The Basic Component Technologies for the Guarantee of Metadata Interoperability

이 수 상(Soo-Sang Lee)*

〈목 차〉

Ⅰ. 서론
Ⅱ. 응용 프로파일
1. 기본 특성
2. 응용 프로파일 사례
Ⅲ. 모델링
1. 시맨틱스
2. 데이터 모델
3. 신택스
Ⅳ. 구현
1. 메타데이터 스키마 편집기
2. 메타데이터 레코드 입력기
3. 메타데이터 관리기
Ⅴ. 자원탐색
1. 통합검색
2. 자원탐색
Ⅵ. 포탈 서비스
1. 기본 서비스
2. 포탈 아키텍처
Ⅶ. 결론

초 록

최근 전자도서관 구축과 관련된 주요 이슈는 상호운용이 가능한 개방형 전자도서관을 어떻게 구현할 것인가 하는 문제에 집중되고 있다. 따라서 이를 위한 여러 가지 개념, 기반기술, 프로토콜 등이 연구개발되고 있다. 본 논문은 이러한 이슈의 핵심 논점을 '상호운용성을 보장하는 메타데이터(metadata interoperability)'로 파악하고, 이를 기반으로 하여 5가지의 핵심 요소기술을 다음과 같이 정리하였다 - 응용 프로파일, 모델링, 구현, 자원탐색, 포탈 서비스. 본 논문은 이 5가지 핵심 요소기술에 대한 개념, 특성, 사례, 표준화 및 기술적 동향 등을 심도있게 분석하였다.

주제어: 전자도서관, 메타데이터 상호운용성, 응용 프로파일, 모델링, 메타데이터 편집기, 자원탐색, 서비스 포탈

Abstract

One of the main issues in developing a digital library is how to implement an open digital library which can be interoperable, and diverse concepts, technology and protocols have been introduced to deal with this issue. In this study, it is analysed in terms of 'metadata interoperability', and based on the analysis, 5 basic component technologies are defined as follows: application profiles, modeling, implementation, resource discovery, portal service. Therefore it explains concepts of those core features, characteristics, cases in point, standardization and technical trends.

Key Words: application profiles, digital library, metadata interoperability, portal service, resource discovery

* TG인포넷 정보기술연구소 부장(sslee@tginfonet.com)
• 접수일 : 2003. 2. 6 • 최초심사일 : 2003. 2. 21 • 최종심사일 : 2003. 3. 4

I. 서 론

자원의 특성을 기술하는 구조화된 데이터(structured data)로서 메타데이터는 전자도서관에서 다루게 되는 다양한 정보자원들을 기술, 조직하는 역할뿐만 아니라 이용자들에게 그 자원들을 검색하도록 하며, 각종 부가적인 서비스를 제공하는 기반이 된다. 즉, 디지털 정보콘텐츠에 대한 통로(pathway) 역할을 한다는 것이다. 전자도서관에서 메타데이터는 가장 기본적인 용도인 자원의 관리 및 탐색을 포함하여, 안전, 개인정보관리, 내용등급, 저작권관리, 보존 등의 다양한 용도에도 사용되어 진다.

일반적으로 메타데이터의 유형모델은 계층(hierarchy) 모델과 객체관계(entity-relationship) 모델의 2가지로 구분할 수 있다. 계층모델은 Dublin Core(unqualified)와 같이 계층이 없는 단순한 메타데이터와 복수의 계층을 갖는 복수계층 메타데이터가 있다. 경우에 따라서는 그룹 또는 세트에 의해 메타데이터 요소들을 분리하고 집단화할 수 있다. 객체관계모델은 IFLA의 FRBR(Functional Requirements for Bibliographic Records) 모델[1]을 준수하는 메타데이터 스키마이며, 박물관, 문화예술자료 등 객체간의 관계를 표시할 필요가 있는 멀티미디어 자원에 대한 메타데이터 기술에 유리하다.

한편, 메타데이터는 작성방법에 따라 몇 가지로 구분할 수 있다. 우선, 기계처리에 의해 메타데이터를 자동생성하는 경우이다. DC-dot와 같이 웹사이트의 HTML문서를 분석하여 메타데이터를 자동으로 생성하는 경우이다. 이렇게 생성되는 메타데이터에는 질적인 문제가 발생할 수 있다. 다른 방법으로는 전문가에 의해 수동으로 생성하는 경우이다. 과도한 비용이 유발될 소지가 있다. 전문가가 생성하는 메타데이터의 사례는 도서관 사서가 생성하는 MARC 데이터를 포함하여, 자원의 저작자가 직접 생성하는 메타데이터, 저작자와 사서가 공동으로 생성하는 경우, 메타데이터 생성기관(agencies)에서 생성하는 경우 등이 있다.

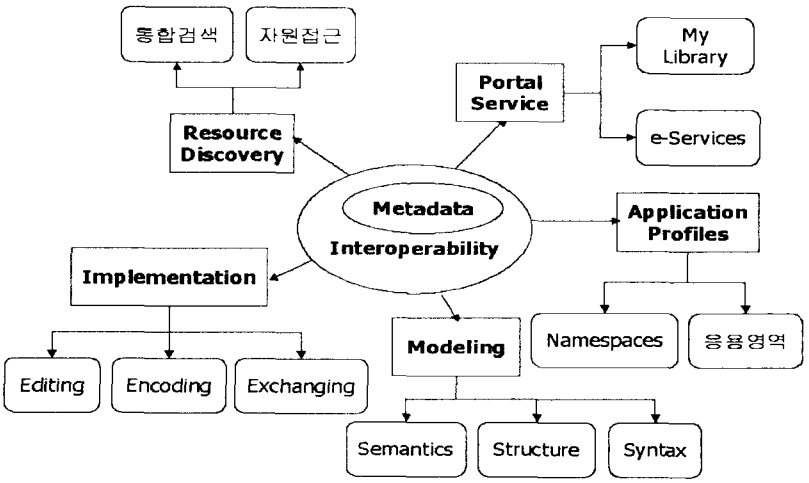
도서관 시스템의 경우, 그 동안 대표적인 메타데이터로 MARC을 사용해 왔다. MARC은 모든 유형의 자료에 포괄적으로 적용되는 범용의 메타데이터 표준이었다. 그러므로 스키마구조가 방대하고 복잡할 수밖에 없었다. 반면, 최근의 경향은 응용 영역마다 특화된 메타데이터 표준을 허용하는 추세로 변하고 있다. 응용 영역마다 적합한 표준 메타데이터를 사용한다는 이점이 있지만, 상호 연동을 하고자 할 경우 복수의 스키마간 상호운용의 문제가 발생하게 된다.

베이커(Thomas Baker)는 개별 시스템마다 독자성을 유지하면서, 메타데이터의 상호운용성을 보장하는 방안으로서 피진이론(pidgin theory)을 제안하였다.[2] 서로 다른 메타데이

1) IFLA, Study Group on the Functional Requirements for Bibliographic Records. *Functional Requirements for Bibliographic Records, Final Report*, 1998. <http://www.ifla.org/VII/s13/frbr/frbr.htm>.

터 언어를 사용하는 공동체 집단들간 의사소통을 가능케 하는 메타데이터 언어의 필요성을 역설하였는데, 그는 이것을 피진 메타데이터(pidgin metadata)라 불렀다. 통합검색 등 메타데이터간 상호운용성을 보장하기 위한 최소한의 메타데이터 요소(기본요소)가 바로 피진 메타데이터인데, 베이커는 이러한 피진 메타데이터로서 Dublin Core(unqualified)를 제안하였다. 따라서 특정한 응용영역은 고유한 메타데이터 세트(이후 언급하게 되는 응용 프로파일)를 선정할 때, 상호호환을 위하여 각 요소들은 미리 정의된 피진 메타데이터로 변환(이것을 dumb-down이라고 함)할 수 있어야 한다.

결국 전자도서관 시스템의 상호운용성(interoperability)은 이기종간 분산환경의 전자도서관들을 연동하여 단일의 통합환경에서 서비스를 제공하는 개념이며 동시에 기능적 요구사항이다. 상호운용성을 위한 전자도서관의 기능요구 수준은 크게 3가지로 구분할 수 있다. 첫째는 연합(federation) 수준으로서, AACR2, Z39.50 등과 같이 엄격한 표준 프로토콜을 상호간에 준수하여 달성된다. 둘째는 다소 느슨한 연합 형태인 수확(harvesting) 수준으로서 각 전자도서관은 기본적인 메타데이터를 교환함으로써 상호연동이 가능하도록 한다. 셋째, 수집(gathering) 수준으로서 각 전자도서관은 프로토콜 수준에서 협력하지 않고, 서비스하고자 하는 정보를 웹크롤러(web crawler)와 같은 검색엔진을 사용하여 수집하는 경우이다.[3)]



〈그림 1〉 전자도서관의 5가지 핵심요소 로드맵

통합환경의 정보자원 서비스를 위한, 상호운용성이 보장되는 전자도서관 구축의 핵심요

2) Thomas Baker, "A Grammar of Dublin Core," *D-Lib Magazine* 6(10) (October 2000), <http://www.dlib.org/dlib/october00/baker/10baker.html>.

3) 이수상, "OAI (Open Archives Initiative) 프로토콜을 이용한 개방형 전자도서관 구축 사례," KERIS 세미나 발표자료(2002. 11.6).

소 기술은 <그림 1>과 같이 '상호운용성을 보장하는 메타데이터(metadata interoperability)'라는 기반개념에서 파악되어야 한다. 이러한 기반개념으로부터 응용 프로파일, 모델링, 구현, 자원탐색, 포탈 서비스의 5가지 핵심 요소기술을 구분할 수 있다. 그러므로 본 논문에서는 개방형 전자도서관 구축을 위한 이 5가지 요소기술에 대한 개념, 특성, 하위기술, 표준화 및 기술적 동향 등을 심도있게 살펴보게 된다.

Ⅱ. 응용 프로파일(Application Profiles)

1. 기본 특성

응용 프로파일은 하나 이상의 네임스페이스(namespace)에서 추출한 메타데이터 요소들로 구성된 메타데이터 스키마(metadata schema)를 말한다. 응용영역의 목적에 부합하는 새로운 시스템을 개발하기 위해서는 메타데이터 스키마에 대한 결정이 우선되어야 한다. 예를 들어, 박물관, 교육, 웹자원, 전자저널 등과 같은 영역에서 전자도서관을 구축할 경우, 이러한 특정 도메인에 필요한 메타데이터 스키마를 정의하는데 사용되는 개념이 바로 응용 프로파일이다.

응용 프로파일 즉, 응용시스템에서 사용되는 각종 용어들을 기술하는 메타데이터 스키마를 결정하기 위해서는 메타데이터가 웹에서 사용가능해야 하며, 가능한 기존의 메타데이터 요소들을 재활용해야 하며, 시스템의 서비스는 메타데이터가 상호운용성을 가지도록 하는 등의 요건이 필요하다.[4] 이 중에서도 가장 중요한 요건은 기존의 메타데이터 표준을 재활용한다는 점이다. 물론 재활용을 위해서는 유용한 메타데이터 스키마를 조사분석해야 하며, 현재의 사용현황에 대해서도 검토해야 한다. 스키마를 이러한 응용 프로파일의 기본적인 특성을 정리하면 다음과 같다.[5]

① 하나 이상의 기존 네임스페이스에서 메타데이터 요소들을 추출한다.
② 가능한 새로운 메타데이터 요소의 도입이 없도록 한다.
③ 인증된 스킴(schemes)과 값(values)을 가진다.
④ 표준 메타데이터 요소를 구체화한다.

정리하면, 응용 프로파일은 기존의 네임스페이스에서 정의된 용어들을 재활용하여 작성된다. 다양한 네임스페이스 스키마들로부터 필요한 용어들을 추출하고, 각 용어들의 구문

4) Eric Duval, et. al., "Metadata Principles and Practicalities," *D-Lib Magazine* 8(4) (April 2002). <http://www.dlib.org/dlib/april02/weibel/04weibel.html>

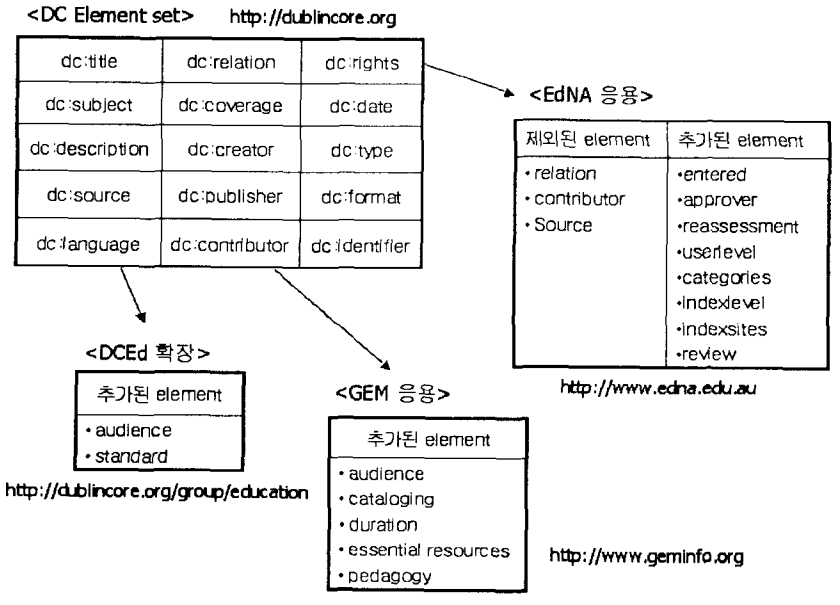
5) Rachel Heery, & Manjula Patel, "Application Profiles: mixing and matching metadata schemas," *ariadne* 25 (Sep. 2000). <http://www.ariadne.ac.uk/issue25/app-profiles/>

적 정의(semantic definitions)를 수정, 보완하여 결정한다. 이러한 작업으로 데이터 요소들의 집합, 사용방법에 대한 규정 및 지침 등이 작성된다.

2. 응용 프로파일 사례

응용 프로파일을 구축하고 관리하는 주체는 2부류의 사람들로 구성된다. 첫 번째 부류는 표준 메타데이터 작성자이며, 응용 영역의 표준이 될만한 데이터 요소들을 도출하고, 선정하게 된다. 두 번째 부류는 시스템 개발자이다. 시스템 개발자는 제공되는 서비스를 기반으로 기존의 메타데이터 세트에서 원하는 요소들을 선정하고, 그것을 원하는 용도에 맞도록 변형하며, 그리고 관련된 스킴들을 관리한다. 서로간 입장이 다른 것은 단지 참가하는 사람들의 역할을 구분한 것뿐이다. 실질적으로 응용 프로파일을 결정하는 사람들은 이 두 가지 관점에서 필요한 역할을 수행하게 된다.

예를 들어 교육영역에 대한 응용 프로파일을 작성한다고 가정하자. 그러면 우선적으로 교육영역에 있어 표준 메타데이터 스키마가 있는지 조사하여야 한다. 교육영역의 표준 메타데이터는 DC기반의 메타데이터 스키마 또는 IMS기반의 메타데이터 스키마를 많이 사용한다는 것을 알 수 있게 된다. <그림 2>는 DC기반의 메타데이터 스키마 사례이다.



〈그림 2〉 DC기반의 메타데이터 스키마 사례

그림에서처럼 DC기반의 메타데이터 스키마의 대표적인 사례는 1) DC 메타데이터 세

트, 2) DC의 교육영역을 위한 확장, 3) GEM 메타데이터 세트, 4) EdNA 메타데이터 세트 등이 있다. 교육영역 표준 메타데이터는 IMS기반의 메타데이터 스키마가 대표적이다. IEEE LOM(Learning Object Metadata)이라고도 불리우며, 이로부터 파생된 관련 메타데이터는 CanCore, iLumina 등이 있다. 결국 교육영역에 대한 응용 프로파일은 DC기반의 메타데이터 스키마와 IMS기반 메타데이터 스키마 중에서 유용한 데이터요소들을 추출하여 작성하면 된다.

최근 응용 프로파일을 작성에는 다음과 같은 3가지 경향을 나타내고 있다. 첫째, DC 메타데이터를 기반으로 하고, 주요한 데이터요소를 다른 메타데이터 스키마에서 추출하여 사용하는 경우이다. 엄격히 따지면, 1) DC 기반의 메타데이터 응용 프로파일에 DC 메타데이터 스키마 이외의 데이터요소들을 사용하는 경우와 2) DC 메타데이터 스키마를 다른 메타데이터 프로파일에 혼합하여 사용하는 경우로 나눌 수 있다. 둘째, 응용 영역에 대한 표준 메타데이터 스키마가 있는 경우, 그것을 기반으로 작성하는 경우이다. 즉, 위 사례에서 IMS 메타데이터 스키마를 활용하는 경우이다. 세 번째, 독자적으로 표준 메타데이터 스키마를 정의하는 경우이다. 물론 시간과 노력이 많이 요구된다.[6]

한편, ISO 표준에 기반한 메타데이타 레지스트리(Metadata Registry)를 통해 네임스페이스 스키마들과 응용 프로파일들을 체계적으로 관리할 수 있다. 새로운 데이터요소나 한정어에 대한 정의, 용례, 변환방법 등을 등록하고 관리함으로써, 원하는 응용 프로파일을 작성하는데 활용할 수 있다. 대표적인 참조사례로는 SCHEMAS Registry를 들 수 있다. 여러 종류의 메타데이터 세트들을 갖추고 있어 응용 프로파일을 작성하는데 지원하고 있다.[7]

Ⅲ. 모델링(Modeling)

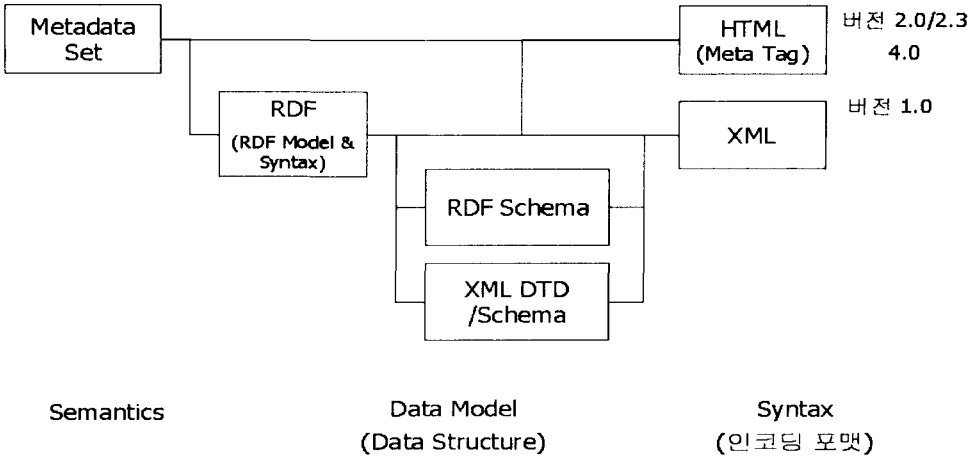
응용 프로파일에서 정의된 메타데이터 요소를 실제의 시스템으로 설계하는 것을 데이터모델링이라 한다. 다양한 메타데이터간 연결을 위해 메타데이터에 대한 시맨틱스(semantics), 데이터 모델(data model), 신택스(syntax)에 의한 공통적인 규칙을 지원하기 위한 고려사항이 요구된다. 정의된 메타데이터 세트에 대한 의미를 어떻게 기술할 것인가? 데이터 모델로서 데이터의 구조를 어떻게 할 것인가? 데이터의 내부 인코딩 구문(syntax)을 어떻게 할 것인가? 하는 문제인 것이다.

응용 프로파일 영역에서 언급하였듯이 데이터 모델링의 최근 동향은 <그림 3>에서 보

6) Jane Hunter & Carl Lagoze, "Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles", <http://archive.dstc.edu.au/RDU/staff/jane-hunter /www10/paper.html>.

7) Peter Johnston, "The role of metadata schema registries", (2001). <http://avalon.ulis.ac.jp/~sugimoto/RPs/dc 2001.pdf>.

듯이 RDF와 XML을 중심으로 논의되고 있다.



〈그림 3〉 메타데이터 모델링 영역

1. 시맨틱스(Semantics)

시맨틱스 영역은 응용 프로파일에서 선정된 메타데이터 세트의 논리적 또는 의미적 프레임웍을 표시한 것이다. 메타데이터 요소명, 정의, 네임스페이스, 스킴, 반복여부, 표준 메타데이터(DC, MARC 등)와 연동정보, 사례, 구문정보 등이 표시되는 메타데이터 명세서(metadata specification)로 정리된다. 한마디로 말하면, 시맨틱스 영역은 메타데이터 세트를 이루는 각 요소들의 의미를 정의하는 부분이다. Dublin Core를 사례로 설명한다면, Dublin Core(버전 1.1)는 ISO/IEC 11179(Specification and Standardization of Data Elements, Parts 1-6) 권고에 따라 각 데이터요소를 아래와 같은 10가지 속성(attributes)으로 표시한다.

- 이름(name) : 데이터요소의 이름
- 식별자(identifier) : 데이터요소에 부여된 유일 식별자
- 버전(version) : 데이터요소의 버전
- 등록기관(registration authority) : 해당 데이터요소를 등록한 기관
- 언어(language) : 데이터요소를 기술하는 언어
- 설명(definition) : 데이터요소의 개념과 특성을 명확하게 표현하는 문장
- 제약조건(obligation) : 데이터요소가 어떤 값으로 표현될 수 있음을 지정
- 데이터형식(datatype) : 데이터요소의 값을 나타내는 형식을 지정
- 최대출현수(maximum occurrence) : 데이터요소의 반복제한 범위를 지정

· 부가정보(comment) : 데이터요소를 적용하는데 필요한 정보 기술

결국 <그림 4>는 Dublin Core Metadata Set(DCMS)의 'Creator'와 'Date' 데이터요소 명세의 일부분이다.

Element : Creator
- Name : Creator
- Identifier : Creator
- Definition : An entity primarily responsible for making the content of the resource.
- Comment : Examples of a Creator include a person, an organisation, or a service. Typically, the name of a Creator should be used to indicate the entity.

Element: Date
- Name : Date
- Identifier : Date
- Definition : A date associated with an event in the life cycle of the resource.
- Comment : Typically, Date will be associated with the creation or availability of the resource. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 [W3CDTF] and follows the YYYY-MM-DD format.

〈그림 4〉 DC의 시맨틱스 정의 일부분

2. 데이터 모델(Data Model)

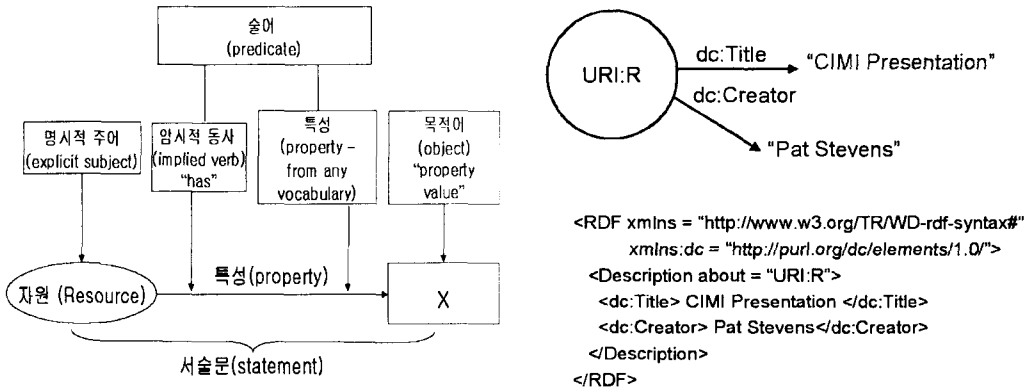
데이터 모델은 처리 대상인 자원을 표현하는 데이터 구조(data structure)를 말한다. 데이터 구조는 각 요소들을 문장형식, 트리 형식, 다이어그램 형식, XML 형식 등 다양한 방법으로 표현할 수 있다. 즉, 메타데이터 정보 표현방법이 데이터모델이며, 데이터구조이다. 어떠한 방법을 선택하더라도, 표현력이 풍부해야 한다. 예를 들어, Dublin Core의 데이터 모델은 각종 스킴(scheme), 형식(type), 하위요소(sub-element), 언어(language) 등을 표현할 수 있는 방법이 있으면 된다. 메타데이터에서 데이터 모델의 표현방법에 대한 논의는 몇 가지로 정의할 수 있다. RDF 모델, RDF Schema 모델, XML DTD 모델, XML Schema 모델, 그리고 최근 논의되고 있는 RDF Schema와 XML Schema를 혼용하는 경우 등 다양한 방법이 존재한다.[8)]

RDF(Resource Description Framework)는 정보자원의 특성(properties)을 기술하는 포맷이다. W3C의 최근 표준인 RDF는 시맨틱웹(semantic web) 및 전자도서관 영역의 핵심

8) Jane Hunter & Carl Lagoze, 전게논문.

기술의 하나이며, 전자도서관의 정보자원 또는 웹 페이지의 의미(semantics)를 메타데이터에 표현하는 수단을 제공하고 있다. 그래서 RDF는 웹을 통해 다양한 응용영역간 상호운용적 메타데이터의 기술과 교환의 프레임웍 역할을 한다.

RDF 모델은 각종 자원들간의 관계를 <resource-property-value>의 3가지(triple) 객체로 구성하는 서술문(statement)을 표현하는 문법이며, 기술언어이다. 결국 RDF 모델은 RDF 서술문(RDF statement)으로 표현되며, RDF 서술문의 일반적인 패턴은 <그림 5>와 다음과 같이 묘사할 수 있다.[9]



〈그림 5〉 RDF 서술문의 그래픽 형식과 DC의 Syntax 사례

RDF를 기반으로 하는 데이터모델은 다음의 2가지 규격을 사용할 수 있다.[10] 첫째는, RDF Model & Syntax(Resource Description Framework Model and Syntax Specification, 1999)이며, 줄여서 그냥 RDF라고 한다. RDF Model은 응용영역의 메타데이터에 대한 인코딩, 교환, 처리 등을 위한 구조적 제한을 기술하는 명세서이다. 즉, <resource-property-value>의 공식적인 관계를 정의하는 역할을 한다. 반면, RDF Syntax는 그렇게 정의된 RDF Model을 XML로 저장하고 교환하는데 필요한 명세서이다. <그림 5>의 우측 그림은 RDF로 표현된 DC모델에 대한 예시이다.

둘째는, RDF Schema이다. RDF Schema는 RDF Model & Syntax와 달리 RDF를 위한

9) Thomas Baker, "A Grammar of Dublin Core," *D-Lib Magazine* 6(10) (October 2000), <http://www.dlib.org/dlib/october00/baker/10baker.html>.

10) Resource Description Framework Model and Syntax Specification. 1999. <http://www.w3.org/TR/REC-rdf-syntax/>. Resource Description Framework Model and Syntax Specification. 2000. <http://www.w3.org/TR/REC-rdf-syntax/>. Resource Description Framework Schema Specification 1.0. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>, XML Schema Part 0: Primer, W3C Working Draft, 22 September 2000. <http://www.w3.org/TR/xmlschema-0/>, XML Schema Part 1: Structures, W3C Working Draft, 22 September 2000. <http://www.w3.org/TR/xmlschema-1/>, XML Schema Part 2: Datatypes, W3C Working Draft, 22 September 2000. <http://www.w3.org/TR/xmlschema-2/>.

각종 어휘(vocabulary)를 이용자가 직접 정의하여 사용 가능하다는 점에서 차이가 난다.

RDF와 달리 XML언어를 통해 데이터모델을 표현할 수 있다. XML DTD와 XML Schema를 사용하는 2가지 경우가 있다. XML DTD는 전자책(e-book)과 같은 고정된 메타데이터 기술에 적합하다고 한다. 반면, XML Schema는 XML DTD보다 융통성이 많고, 재사용이 가능한 이점이 있다. 또한 네임스페이스를 지원하는 기능도 있어 최근에는 XML Schema의 사용이 많이 권장되고 있다.[11]

3. 신택스(Syntax)

마지막으로 메타데이터의 신택스는 메타데이터의 기술과 교환을 위한 인코딩 포맷과 같은 구문을 말한다. 관계형 DB, HTML 메타태그(meta tag), 술어논리언어 등 여러가지 인코딩 포맷을 사용할 수 있지만, 최근의 경향은 표준성과 호환성을 위하여 XML을 많이 사용하고 있다. 아래 <그림 6>은 메타데이터의 기술, 저장, 교환 등을 위한 XML 표현(XML binding)에 따라 표기된 XML 레코드의 사례이다.

```
<?xml version="1.0"?>
<record
  xmlns="http://example.org/learningapp/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://example.org/learningapp/schema.xsd"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:ims="http://www.imsglobal.org/xsd/imsmd_v1p2">

  <dc:title>Frog maths</dc:title>
  <dc:identifier>http://somewhere.com/frogmaths/</dc:identifier>
<dc:description>Simple maths games for 5-7 year olds.</dc:description>
  <ims:typicallearningtime><ims:datetime>0000-00-00T00:15</ims:datetime>
  </ims:typicallearningtime>
</record>
```

〈그림 6〉 XML로 표기된 메타데이터 사례
(출처 : http://www.ukoln.ac.uk/metadata/dcmi/dc-xml-guidelines/)

11) Andy Powell & Pete Johnston, "Guidelines for implementing Dublin Core in XML". < http://www.ukoln.ac.uk/metadata/dcmi/dc-xml-guidelines/>.

Ⅳ. 구현(Implementation)

메타데이터의 구현 영역은 전자도서관 내부에서 메타데이터를 실질적으로 처리하는 편집, 인코딩, 교환 등의 기능을 가진 시스템을 제작하는 것을 말한다. 프로그램 코딩 영역으로서, 궁극적으로 메타데이터 편집기(metadata editor)의 형태로 구현된다. 앞서 언급된 개념과 모델을 시스템에 실제로 적용하게 되므로, 기술적인 한계, 성능, 안정성 등의 요건들을 충분히 검토하는 것이 중요하다.

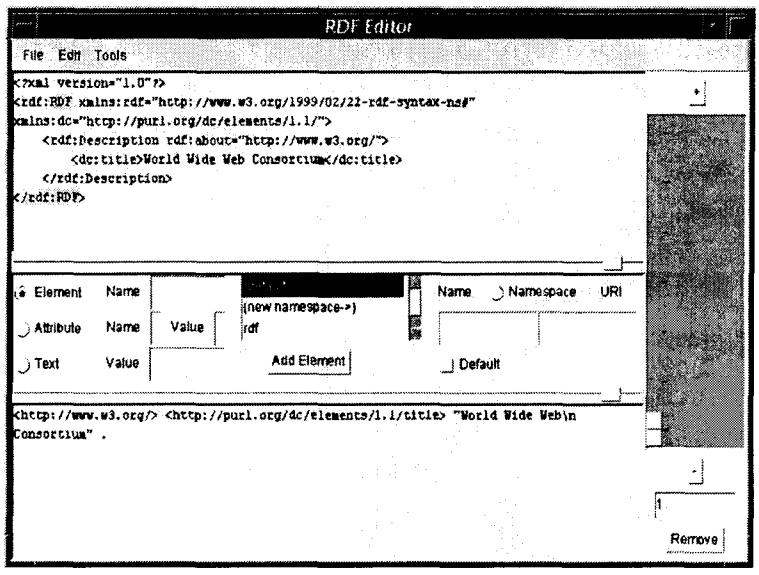
우선, 메타데이터 편집기의 주요 요구사항부터 정리하고자 한다. 메타데이터 편집기는 주된 사용자에 따라 웹기반의 메타데이터 편집기(web-based metadata editor)와 웹기반이 아닌 관리자용 메타데이터 편집기의 2가지로 구분할 수 있다. 웹기반의 메타데이터 편집기는 협력환경에서 저작자가 직접 디지털자원을 전자도서관에 입력(제출)하는 경우에 사용한다. 학위논문 전자도서관인 경우, 학위취득자인 연구자가 직접 메타데이터를 입력하고 원문을 제출하도록 하는 도구이다. 관리자용 메타데이터 편집기는 디지털자원의 저작자보다 사서와 같은 메타데이터 관리자가 사용하는 편집기라 할 수 있다. 후자의 편집기는 전자의 편집기를 통해 저작자가 입력한 메타데이터나 원문을 수정/보완하는 기능뿐만 아니라 보안, 저장소관리, 검색환경 설정 등 시스템관리를 위한 각종 기능들을 제공하여야 하므로, 다소 복잡하게 구현될 수 있다.

한편, 메타데이터 편집기는 입력대상의 메타데이터에 따라 전용편집기와 범용편집기로 구분할 수 있다. 전용편집기는 단일의 메타데이터 세트 즉, DC, MARC, LOM 등 전문영역의 응용 프로파일(표준 메타데이터 스키마를 가진)을 위한 편집기를 말한다. 즉, DC Editor, MARC Editor, LOM Editor 등 표준 메타데이터 또는 단일 응용영역의 응용 프로파일을 위한 메타데이터 편집기가 된다. 범용편집기는 다양한 메타데이터 스키마에 대한 메타데이터를 입력할 수 있는 통합환경의 편집기가 된다. 매우 이상적인 편집기이며, 하나의 시스템에 다양한 메타데이터 스키마를 가지는 디지털자원을 통합관리 할 요구가 있을 때 필요하다. 반면 전용편집기는 단일의 메타데이터 스키마를 가지는 메타데이터를 관리하는 경우에 사용된다. 이처럼 응용 영역에 따라 편집기의 기능요구사항이 달라질 수 있다.

일반적으로 메타데이터 편집기가 갖추어야 할 기본 기능은 크게 메타데이터 스키마 편집기, 메타데이터 레코드 편집기, 그리고 메타데이터 관리기의 3가지 영역으로 나눌 수 있다.

1. 메타데이터 스키마 편집기(metadata schema editor)

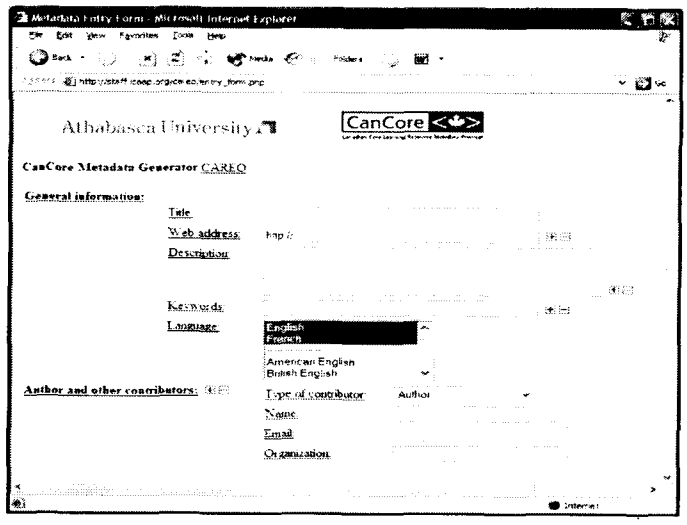
응용 프로파일에서 정의한 메타데이터 스키마를 생성/수정/삭제하는 기능을 수행한다. 사용하고자 하는 스키마의 유형(RDF, RDF Schema, XML DTD/Schema)에 따라 주요 세부기능이 달라질 수 있다. 핵심적인 기능은 응용 프로파일로 설계된 메타데이터 세트에 대한 스키마파일을 생성하고 관리하는 기능이라 할 수 있다. <그림 7>은 RDF 스키마를 편집하는 RDF Editor의 사례이다.



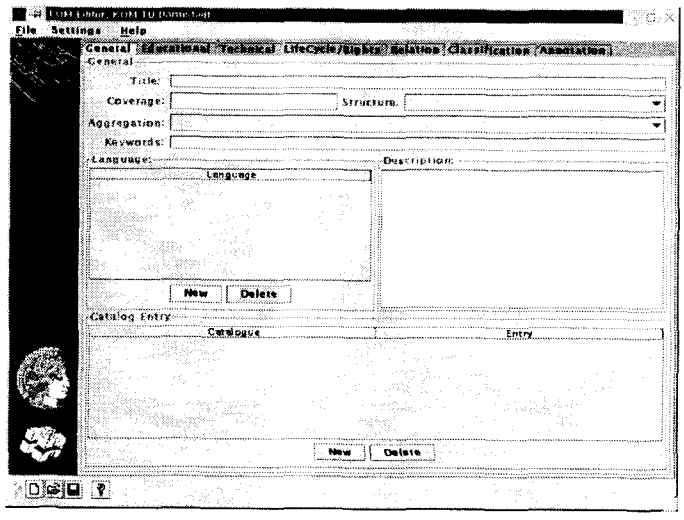
〈그림 7〉 RDF Editor 사례 (출처 : http://www.cs.rpi.edu/~puninj/rdfeditor/)

2. 메타데이터 레코드 입력기(metadata record editor)

메타데이터 레코드의 편집화면을 제공하여, 실제의 데이터를 입력/수정/삭제/검증 등 기능을 수행한다. 메타데이터 스키마 편집기는 스키마 설계를 위한 관리자용이라면, 메타데이터 입력자의 입장에서 보는 메타테이터 편집기이다. 핵심기능은 데이터 입력기능, 입력도움말 기능, 검증기능, 입력된 데이터의 XML 변환 기능, 저장기능, 데이터 검색기능 등을 들 수 있다. 아래 <그림 8>과 <그림 9> 각각은 메타데이터 레코드 입력기의 사례들이다.



〈그림 8〉 Web기반의 CanCore Metadata Editor
(출처: http://staff.icaap.org/careo/entry_form.php)



〈그림 9〉 LOM Editor 사례 (출처: http://www.multibook.de/lom/en/workEN.html)

3. 메타데이터 관리기(metadata manager)

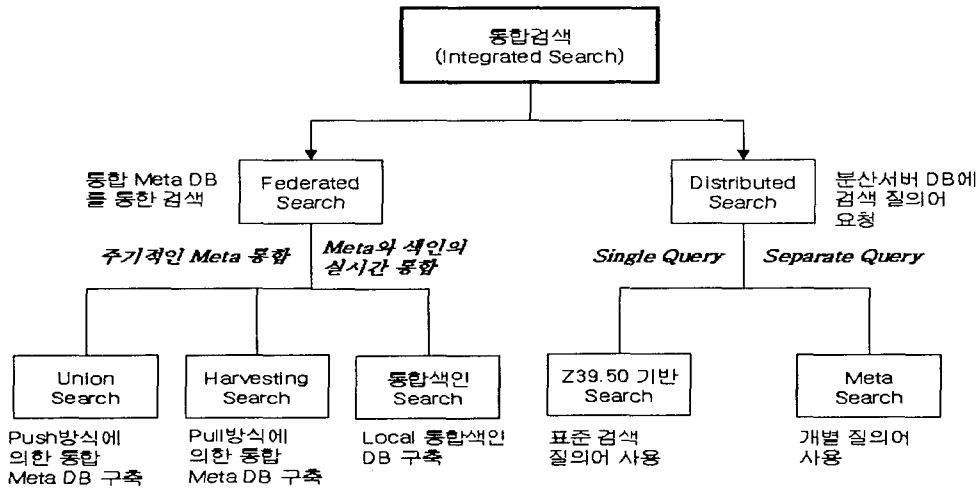
메타데이터의 외부 반출 및 외부 메타데이터의 반입을 위한 교환/처리 기능, 메타데이터 스키마 설계자에 의해 정의된 메타데이터 스키마에 따른 XML, 관리정보, 검색 DB의 물리적인 메타데이터 저장구조를 생성, 관리하는 기능을 갖추고 있다.

V. 자원탐색(Resource Discovery)

여기서의 자원탐색은 지리적으로 분산되어 있는 각각의 전자도서관(아카이브)들을 통합환경에서 검색(search)하고 해당 자원에 대한 편리한 접근(access)을 말한다. 즉, 자원탐색은 통합검색(integrated search)과 자원접근(resource access)이라는 2가지 주요한 정보활동으로 구성된다. 통합검색은 하나의 단위시스템 내의 정보를 탐색하는 기능이 아니라, 분산환경의 전자도서관들을 대상으로 하는, 텍스트 자원뿐만 아니라 박물관, 도서관, 전자책, 멀티미디어, 교육매체들에 대한 메타데이터 정보를 통합환경에서 검색하는 것을 의미한다. 자원접근은 통합환경에서 검색된 결과정보를 기반으로 적합한 자원(appropriate copy)을 찾아서 접근하는 하이퍼텍스트 연계(hypertext link)기능을 말한다.

1. 통합검색

일반적으로 통합검색 기능은 검색대상 시스템의 데이터나 시스템 환경에 따라 여러 가지 방식으로 구현이 가능하다. 여러 전자도서관 사례에서 사용하고 있는, 대표적인 통합검색 방안의 유형을 정리하면 <그림 10>과 같다.



〈그림 10〉 자원탐색을 위한 통합검색의 5가지 방법

통합검색의 각 방안의 유형은 통합메타 DB를 통한 검색인가 아닌가에 따라 크게 연합검색(federated search)과 분산검색(distributed Search)으로 구분하였다. 연합검색은 통합

메타 DB의 구축 방법에 따라 종합검색(union search)과 수확검색(harvesting search), 그리고 통합색인 검색의 3가지로 나누어진다. 종합검색은 MARC기반의 종합목록 시스템에서 사용하는 방법으로 참여기관과 통합기관 모두 표준형식의 메타데이터(MARC)를 사용하는 점, 참여기관에서 통합기관으로 메타데이터를 송부하는 푸시(push)방법을 사용한다는 점이 주요 특징이다. 이와 달리 수확검색은 참여기관(data provider)의 메타데이터의 형식에 관계없이 표준형식의 메타데이터(단순 DC)로 통합하며, 통합기관에서 참여기관의 메타데이터를 수확하는 풀(pull)방식을 사용한다. 통합색인 검색은 메타데이터 수준이 아니라 색인 수준에서 데이터를 통합하는 특수한 형식이라 할 수 있다.

교차검색(cross search)라고 부를 수도 있는 분산검색은 검색질의어의 구성방식에 따라 Z검색과 메타검색으로 구분할 수 있다. Z검색은 Z39.50이라는 표준형식의 질의어를 대상 시스템에 보내어 검색하고, 결과를 얻어오는 방식이다. 메타검색은 대상 시스템의 검색엔진에서 사용하는 검색방식으로 검색 질의어를 보내어 검색하는 방식이다.

각각 방법의 특징을 요약정리하면 <표 1>와 같다.

〈표 1〉 5가지 통합검색 방식 비교

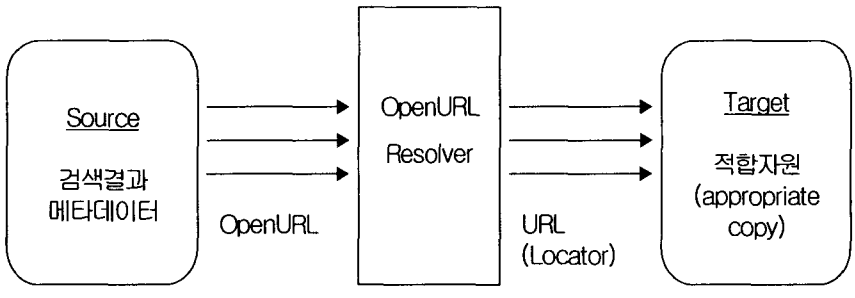
방 식	특 징
종합검색	· 중앙집중식 통합 Meta DB(Union DB)를 구축 · 참여기관은 주기적/즉시 표준목록으로 데이터를 송부(Push 방식) · 수집된 결과는 필터링(중복제거, 정렬 등) 작업 수행 · 사례 : OCLC, KERIS 등 종합목록
수확검색	· 검색 서비스 서버에 통합 Meta DB(Local Copy DB)를 구축 · 데몬 형식의 Harvester에 의해 메타데이터를 수확(Pull 방식) · OAI-MH(OAI Metadata Harvesting) 프로토콜 준수 · 수집된 결과는 정규화 및 필터링 작업 수행 · 사례 : arXiv e-Print Archive, NCSTRL, NDLTD, NASA Technical Reports Server 등 OAI-MH 프로토콜을 수용하는 Archives
통합색인 검색	· 각 DB의 Meta 및 색인의 통합 DB를 구축 · 통합 DB 구축시 필터링(중복제거, 정렬 등) 수행 · 사례 : 연세대, 이화여대 등 전자도서관
Z검색	· 각 서버에 단일의 표준질의어로 동시 검색요청 · 수집된 결과에 대한 편집작업 수행 · 사례 : Z39.50 Protocol을 수용하는 서버
메타검색	· 각 데이터 서버에 해당되는 개별 질의어로 검색요청 · Z39.50기반 검색의 대안으로 사용 · 사례 : NCSTRL 초기 시스템 등

2. 자원접근

통합검색의 결과, 이용자는 검색된 자원에 대한 메타데이터를 얻게 된다. 자원 자체(원

문형식이든, 인쇄형식이든)를 직접 획득하기 위해서는 또 하나의 과정이 필요하다. 이것이 바로 자원접근의 문제이다. 가장 쉬운 접근방법은 식별자를 통하는 경우이다. 검색결과의 메타데이터에 식별자가 포함되어 있더라도, 해당 자원으로 접근하지 못하는 경우가 있다. 식별자가 DOI와 같은 고정된(persistent) 식별자이거나, 자원을 소장하고 있는 시스템의 고유한 식별자(unique-id)라면 접근에 문제가 없다(접근권한에 관한 문제는 논의에서 제외함). 그러나 메타데이터에 포함된 식별자가 URL과 같이 변화 가능성이 있거나, 식별자 요소를 아예 포함하지 않는 경우도 있다. 결국 식별자의 유형이 상관없이, 이용자는 검색결과를 바탕으로 자신이 원하는 자원으로 접근할 수 있는 일반적인 매커니즘이 있어야 한다.

검색결과에서 얻어진 식별자나 메타데이터는 이용자가 접근하고자 하는 자원에 대한 인용(citation)정보가 된다. 이러한 인용정보 메타데이터를 URL의 형식으로 변환하는 방법을 제시한 것이 바로 OpenURL 프레임웍이다. OpenURL은 DOI와 같은 고정된 식별자이거나, 대상자원을 유일하게 구분할 수 있는 메타데이터 요소들의 집합으로 구성된다. 따라서 검색결과의 메타데이터는 OpenURL로 재구성되어 원하는 정보자원에 접근하는 실질적인 주소인 URL로 변환하게 되는데, 이것은 OpenURL Resolver라는 OpenURL 해석기의 도움으로 가능하다. OpenURL을 기반으로 하는 적합자원의 접근 개념도는 <그림 11>과 같다.



〈그림 11〉 OpenURL을 통한 자원접근의 개념도

그림에서처럼 이용자가 원하는 적합자원에 접근하는 과정에는 OpenURL과 관련하여 다음의 3가지 핵심적인 처리기능이 필요하다.

① OpenURL 생성기능 - 정보원(source)이 되는 검색결과로부터 OpenURL 데이터를 생성한다.

② OpenURL 전송기능 - HTTP 프로토콜을 이용하여 OpenURL 데이터를 OpenURL 해석기에 전송한다.

③ OpenURL 해석기능 - OpenURL 해석기에 입력된 OpenURL 데이터를 토대로 적합한 대상자원(target)의 URL인 주소(Locator)를 찾아준다.

OpenURL 해석기에 의해 얻어진 주소는 이용자의 입장에서 판단한 가장 적합한 자원(단수 또는 복수)을 접근할 수 있는 주소를 말한다. 대상자원(target)은 전자저널, 학술DB, full-text기반의 원문, 특정한 도서관의 OPAC, 상호대차 신청 사이트 등이 해당된다. 따라서 대상자원의 적합성 여부는 소속기관에 따른 자원의 소유상태, 이용자의 인증, 선호도 등에 따라 해석기가 판단하게 된다. 그렇게 판단된 주소는 직접주소(direct locator)와 간접주소(indirect locator)의 2가지 유형으로 구분된다. 직접주소는 해당 자원으로 직접 접근할 수 있는 URL이며, 간접주소는 인쇄자원에 대한 위치정보(특정 도서관의 청구기호), 상호대차 신청화면 등과 같이 해당 자원에 접근할 수 있는 서비스로 안내한다.

Ⅵ. 포탈 서비스(Portal Service)

1. 기본 서비스

전자도서관의 포탈 서비스는 이용자 친화적인 인터페이스를 통해 전자도서관의 다양한 부가가치 디지털 서비스(e-service)를 제공하고, 또한 개인화 서비스(My Library)를 제공하는 역할을 수행한다. 앞서 언급된 내용들은 전자도서관의 관리자적 관점에서 고려되는 각종 개념 및 기술이라면, 전자도서관 포탈이라는 형식으로 제공되는 서비스는 이용자의 관점에서 인식하게 되는 도서관이 된다. 일반적인 비즈니스 영역의 포탈과 마찬가지로 전자도서관 포탈은 다음과 같은 기본적인 특징 및 서비스를 제공하게 된다.

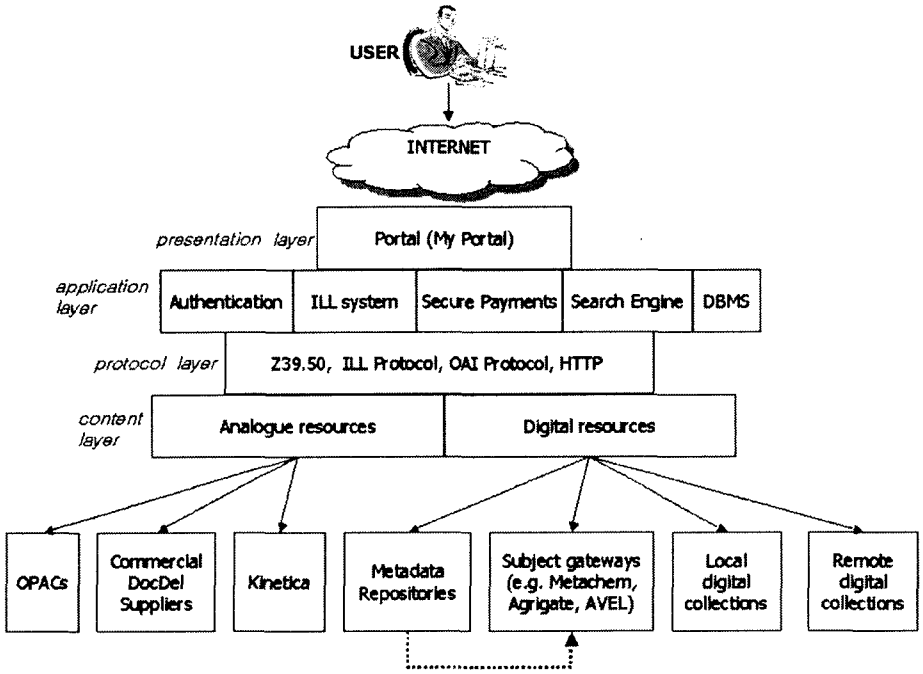
첫째, 단일의 정보 창구(gateway) 역할 수행한다. 기능적으로는 단일 로그인/접근 기능, 레가시(legacy) 및 분산서버 시스템의 통합, Non-stop(24x7) 서비스 등이 해당된다. 형태로는 가상포탈(virtual portal), 게이트웨이 포탈(gateway portal), 하이브리드 포탈(hybrid portal) 등으로 나타난다. 둘째, 다양한 정보자원의 통합 서비스를 제공한다. 온라인 자원뿐만 아니라 아날로그 자료들을 통합한 하이브리드형의 정보자원, 통합정보검색, 브라우징 및 자원연동(reference linking) 기능, SDI와 정보추천 등 각종 부가가치 정보제공 서비스가 포함된다. 셋째, 고객 중심의 서비스를 제공한다. 개인화된 인터페이스, push/pull 방식의 정보전달, 커뮤니티 역할, 브랜드형의 도서관서비스 등이 포함된다. 넷째, e-경영 기술을 접목하여 부가가치를 높인다. 전자상거래(B2B, B2C), 기업정보포털(EIP), 고객관계관리(CRM) 등의 기술이 접목될 수 있다.

정리하면, 도서관 포탈의 핵심기능은 첫째, 가장 중요한 기능으로서 도서관 내외의 다양한 정보원들에 대한 통합검색(탐색 및 접근)을 제공할 수 있어야 한다. 둘째, 상호대차, 원문정보제공, 대출정보 조회, SDI 등 각종 부가가치 서비스를 제공하여야 한다. 셋째, 개인의 기호에 맞도록 맞춤서비스를 제공할 수 있어야 한다. 마지막으로, 유관 시스템과 연

동하는 기능을 제공할 수 있어야 한다.

2. 포탈 아키텍처

<그림 12>는 이러한 특성을 갖춘 전자도서관 포탈의 일반적인 아키텍처이다. 이 아키텍처는 다음과 같이 4가지 계층으로 구성되어 있다.[12]



〈그림 12〉 전자도서관 포탈 아키텍처

① 표현 계층(presentation layer) : 이용자들이 눈으로 확인하는 웹 기반의 인터페이스로서 통합검색 및 디지털 서비스를 제공한다. 검색서비스와 디지털자원제공 서비스, 그리고 개인화 서비스 등이 포함된다.

② 응용 계층(application layer) : 서로 다른 유형의 응용 소프트웨어로 구성된다. 인증시스템, 통합검색엔진, 저작권시스템, 개인화 기능, 채팅, 이메일 기능, VOD 기능 등이 응용 소프트웨어 사례이다.

③ 프로토콜 계층(protocol layer) : 응용 소프트웨어의 클라이언트 측과 서버 측을 연

12) Edward Lim, "The Last Book: The Delivery of Future Content". *alia 2000 proceedings*. <http://www.alia.org.au/conferences/alia2000/proceedings/edward.lim.html>.

결하는 표준 커뮤니케이션 방법을 구축하는 역할을 수행한다. 따라서 다양한 프로토콜이 적용된다.

④ 콘텐츠 계층(content layer) : 이용자가 접근할 수 있는 모든 정보자원을 포함한다. 아날로그 형식의 도서뿐만 아니라, 인용 및 서지정보, 전자저널 및 전자책과 같은 원문정보, 전문 게이트웨이, 메타데이터 저장소, 웹사이트 등이 해당된다.

Ⅶ. 결 론

지금까지 '메타데이터 상호운용성(metadata interoperability)'을 보장하기 위한 5가지 요소기술에 대한 개념, 특성, 하위기술, 표준화 및 기술적 동향 등을 심도있게 살펴보았다. 응용영역에 적합한 메타데이터 스키마 선정을 위한 응용 프로파일, 선정된 프로파일에 대한 데이터 모델링, 메타데이터 편집기 구현, 분산환경의 자원탐색, 도서관 포탈 서비스로 구분된 이 5가지 핵심요소는 개방형 전자도서관을 구축하기 위한 필요충분조건이라 할 수 있다. 이 요소기술들이 제대로 반영되어야 하며, 또한 얼마나 잘 준수하였는가 하는 점이 바로 전자도서관을 평가하는 기준이 될 수 있을 정도이기 때문이다.

이러한 모든 논의를 관통하는 하나의 중심개념을 도출한다면, 그것은 바로 현재의 전자도서관은 웹이라는 개방된 공간을 중심으로 분산된 다양한 전자도서관 콘텐츠들을 연결하고 있다는 점이다. 결국 이러한 개방성/연결성을 보장하기 위해, 메타데이터의 상호운용성 기술이 무엇보다도 중요해 졌으며, RDF라는 메타데이터 표현기반, XML이라는 처리언어, 통합검색, 포탈서비스 등과 관련하여 표준화된 요소기술과 개념들을 효율적이고 안정적으로 구현하도록 요구되고 있다.

추측컨데, 향후의 전자도서관은 시맨틱웹을 근간으로 하는 새로운 형태의 거대한 정보공간을 구성하는 중심역할을 수행할 것으로 보인다. 현재와 같은 웹기반의 전자도서관은 콘텐츠의 위치적 연동 및 개방을 추구한 것이라면, 시맨틱웹상의 전자도서관은 콘텐츠의 의미적 연동까지도 보장하게 되며, 보다 인간 친화적인 정보혁명으로서 자리매김할 것으로 보인다.

결국 미래의 전자도서관은 현재까지 논의된 개방성과 호환성을 위한 각종 표준화 기술들을 기반으로 설계, 구현될 것이다. 이러한 배경은 모든 정보자원들을 시맨틱웹과 같은 하나의 거대한 검색공간을 구축할 수 있으며, 각종 부가가치 서비스를 제공할 수 있어, 디지털 사회의 정보인프라로서 역할을 수행할 것이다.

〈참고문헌은 각주로 대신함〉