

# OAI 기반 Open Digital Library 연구

## A Study on the OAI based Open Digital Library

이 상 기  
Sang-Gi Lee

### 차 례

- |                            |                              |
|----------------------------|------------------------------|
| 1. 서 론                     | 4. Open Digital Library 적용사례 |
| 2. OAI 연구                  | 5. 결론                        |
| 3. Open Digital Library 연구 | • 참고문헌                       |

### 초 록

현재 디지털라이브러리는 상호운용성의 문제, 콘텐츠 확장의 한계, 편리한 관리 환경의 필요성에 직면해 있다. 이를 해결하기 위해선 표준화와 상호운용성이 전제된 시스템이 필요하다. 이런 문제를 근원적으로 해결해 줄 수 있는 차세대 디지털라이브러리 시스템으로 주목 받고 있는 것이 Open Digital Library이다. 여기서는 Open Digital Library의 사상적 기반이 된 OAI 및 OAI-PMH에 대해 조사하였으며, Open Digital Library 프레임워크, 설계 주안점, 적용사례를 연구하였다. 끝으로 표준화된 요소기술을 활용한 Open Digital Library의 발전방향과 확장성을 제시하였다.

### 키 워 드

디지털라이브러리, OAI 프로토콜, XOAI 프로토콜, 자유이용, 상호운용성, 표준화, 컴포넌트화

\* 한국과학기술정보연구원 정보기술지원실 선임연구원  
(Senior Researcher, Information Technology Support Dept., KISTI, sklee@kisti.re.kr)  
• 논문접수일자 : 2004년 8월 31일  
• 게재확정일자 : 2004년 9월 15일

## ABSTRACT

Digital library technology is facing issues such as interoperability, content extension, and the need for a more efficient management environment. A system that is based on standardization and interoperability is required to resolve these issues. Open Digital Library is the system for the next generation that resolves the above problems. We have conducted some research on issues such as OAI and OAI-PMH, which are the basic knowledge behind the Open Digital Library technology and also investigated the Open Digital Library framework, cases of study and implementation of this technology. Finally, we have made some suggestions on the extendability of Open Digital Library technology using the standardized technologies.

## KEYWORDS

Digital Library, Open Digital Library, OAI, OAI-PMH, XOAI-PMH, Open Access, Interoperability, Standardization, Componentization

### 1. 서론

디지털라이브러리가 조직체내 산재하는 디지털 자원의 통합관리나 도서관 자동화 범주를 넘어, 포탈기능을 통한 통합 환경의 부가가치 서비스를 제공할 필요가 있는 모든 영역에 적용되는 범용의 디지털 콘텐츠 관리시스템(CMS)으로 변모하고 있다. 미국, 영국, 호주 등 정보 선진국들은 국가적인 차원에서 연구 및 개발에 박차를 가하고 있다.

디지털라이브러리는 자체의 호호성과 문헌 정보학, 컴퓨터공학, 네트워크 분야와 교집에 있는 관계로 시스템 표준화나 컴포넌트화가 미약하였다. 게다가 라이프 사이클에 따라 조만간 상호운용성의 문제, 콘텐츠 확장의 한계, 편

리한 관리 환경의 필요성에 직면할 것이라고 한다. 현재 이 분야 종사자들의 공통 관심사는 각종 정보자원의 통합, 표준적인 시스템 및 프로토콜 적용, 다양한 서비스 및 적용영역의 확장, 장기보존일 것이다.

이런 위기 속에서 디지털라이브러리 분야도 변화의 바람이 불고 있다. 대표적인 것이 학술 정보의 자유로운 이용(Open Access)을 기술적으로 지원하기 위해 출범한 OAI이다. OAI에서는 디지털라이브러리(Open Archives)의 상호운용성 증진을 위해 간결하면서도 확장성 있는 프로토콜인 OAI PMH를 개발·보급하고 있으며, 몇몇 기관에서는 OAI 기반 디지털라이브러리 구축 솔루션을 무료로 공급(Open Source)하고 있다. 현재 OAI 기반 디지털라이

브리리는 공식적으로 등록기(registry)에 등록 된 것만도 182개(실험적인 것까지 하면 650개)로 매우 빠른 속도로 확산되고 있다.

이런 디지털라이브러리들을 표준 프로토콜로 연계하여 네트워크화 함으로써 상호운용성이 보장되는 거대한 디지털라이브러리(Larger Digital Library) 체제를 구축할 수 있는데 이것이 Open Digital Library이다. 이렇게 됨으로써 콘텐츠에 대한 접근성을 획기적으로 높일 수 있을 뿐만 아니라 디지털라이브러리의 고질적인 문제였던 표준화와 상호운용성 문제를 해결할 수 있다.

본 고에서는 Open Digital Library의 사상적 기반이 된 OAI와 여기서 개발된 표준 프로토콜인 OAI PMH에 대해 살펴보았으며, Open Digital Library 프레임워크, 설계 주안점, 적용사례를 연구하였다. 마지막으로 표준화된 요소기술을 활용한 ODL 발전방향과 확장성을 제시하였다.

## 2. OAI 연구

### 2.1 OAI(Open Archives Initiative) 개요

새로운 학술커뮤니케이션 모델인 Open Access 운동이 활발하며 이를 체계적으로 지원하기 위한 기술들이 개발되고 있다. Open Access를 기술적으로 지원하기 위해 초창기 Open Access 주창자들이 모여 결성한 조직이 OAI(Open Archives Initiative)이다. 이들은

1999년 LANL(Los Alamos National Laboratory)에서 새로운 학술 커뮤니케이션으로의 전환을 위한 기술적인 지원을 목적으로 포럼을 개최했는데 이것이 OAI의 시작이었다.

OAI의 주된 역할은 디지털 콘텐츠의 효율적인 유통과 활용을 목적으로 상호운용성을 갖는 표준을 개발하고 보급하는 것이다(Issues 2004). 이런 목적 하에 개발된 대표적인 프로토콜이 OAI PMH(Open Archives Initiative Protocol for Metadata Harvesting)이다.

OAI PMH(이하 OAI 프로토콜)는 확장성이 뛰어날 뿐만 아니라 매우 간결(simple)하다. 게다가 국제표준인 XML, Dublin Core(unqualified), HTTP에 기반하고 있다.

개발배경은 분산된 각 오픈 저장소(Open Archives)간 상호운용성을 위한 보다 저렴한 요소기술의 필요성 때문이다. OAI 프로토콜은 가볍고 기술 장벽이 낮으며, OAI 저장소(OAI Compliant Archives)간 정보를 쉽게 공유할 수 있는 매우 유용한 프로토콜이다. 기존 Z39.50 프로토콜이 정교한 메타데이터인 MARC을 채택함으로써 비용 부담이 컸던 반면, OAI 프로토콜은 Dublin Core 메타데이터를 사용하여 상대적으로 저비용이다.

2001년에 OAI 프로토콜 v.1.0이 나왔으며 연이어 2002년 6월에는 v.2.0이 발표되었다.

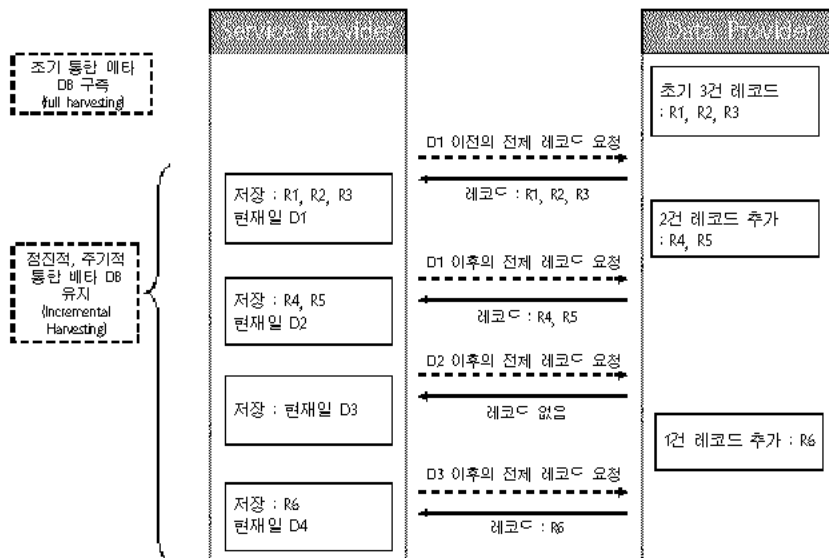
OAI에 기반 한 저장소를 'OAI Compliant Archives' 또는 'Open Archives' 라고 하는데 이것은 OAI 프로토콜을 준수하며, 원격지에서 OAI 프로토콜로 메타데이터에 접근할 수 있는

시스템을 일컫는다. 또한, 'Open Archives' 와 동의어로 통용되는 'repository' 는 전통적인 디지털라이브러리에서는 디지털 객체의 서고(collection)였으나 OAI에서는 네트워크 접근이 가능하고 OAI프로토콜을 지원하는 저장소를 뜻한다(hussein's space 2004).

현재 OAI 프로토콜은 OAI에서 제정한 시스템 간 메타데이터의 수집 및 교환을 지원하는 실질적인 표준(de facto standard)으로 자리매김하고 있다.

OAI에서는 디지털라이브러리의 각 영역별 품질제고 및 집중화를 위해 편의상 두 부분으로 분리했는데 하나는 데이터제공자(Data Provider, DP)이고, 다른 하나는 서비스제공자(Service Provider, SP)이다. 데이터제공자는 정보관리(Information Management)와 서비

스제공자에게 정보를 표준적인 방법으로 노출하는 것이 주 임무이며, 원문(resource) 보유자이기도 하다. 서비스제공자는 데이터제공자로부터 메타데이터 등을 수집하여 부가가치 정보를 서비스하는 곳이다. 데이터제공자가 되기 위한 최소한의 조건은 각 저장소 안의 레코드는 영구적이거나 적어도 영구적인 식별자(identifier)를 가지고 있어야 한다. OAI에서는 레코드가 전 세계적으로 유일하기 위해 식별자 안에 저장소 이름을 포함시킬 것을 강력하게 권고하고 있는데, 형식은 oai + 저장소 아이디 + 레코드 아이디(예 : oai:archive\_id:record\_id)이다. 아울러 모든 레코드는 점진적 수확(Incrementally Harvest)에 필요한 소인(datestamp)을 가지고 있어야 한다. 한편 서비스제공자는 분산된 저장소에 추가되거나 변경



〈그림 1〉 점진적 수확 과정

되는 메타데이터의 점진적 수확이 가능하여야 한다.

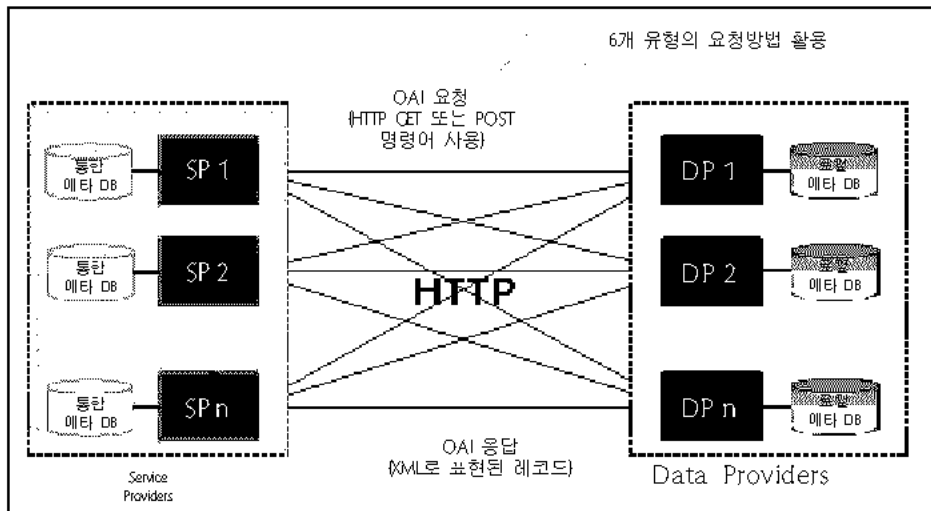
위 <그림 1>은 서비스제공자가 데이터제공자로부터 메타데이터를 수확하는 과정을 도식화한 것이다. 처음 수확자(harvester)는 해당 저장소의 모든 메타데이터를 수집하며, 이후부터는 추가되거나 수정된 것만을 선별적으로 수집한다.

서비스제공자인 OAI 클라이언트는 데이터제

공자인 OAI 서버에게 HTTP GET 또는 POST 방식을 사용하여 6가지 요청(GetRecord, Identify, ListIdentifiers, ListMetadata Formats, ListRecords, ListSets)으로 메타데이터를 요청하게 되며, 요청받은 OAI 서버는 응답 자료를 XML 인스턴스(문서)로 응답한다. 이러한 요청·응답 처리를 규정한 프로토콜이 OAI 프로토콜이다. 6가지 요청을 간략히 설명하면 <표 1>과 같다(The Open Archives

<표 1> OAI 프로토콜 6가지 요청

요청	기능 설명
GetRecord	저장소로부터 하나의 레코드를 수확
ListRecords	한꺼번에 대량의 레코드를 수확
ListIdentifiers	저장소에서 수확 가능한 식별자(identifier)를 검색
ListMetadataFormats	특정 레코드의 메타데이터 포맷 종류를 검색
Identify	저장소의 기본정보(Base_URL 등)를 검색
ListSets	저장소의 셋(set)구조를 검색



<그림 2> OAI 프로토콜 모델

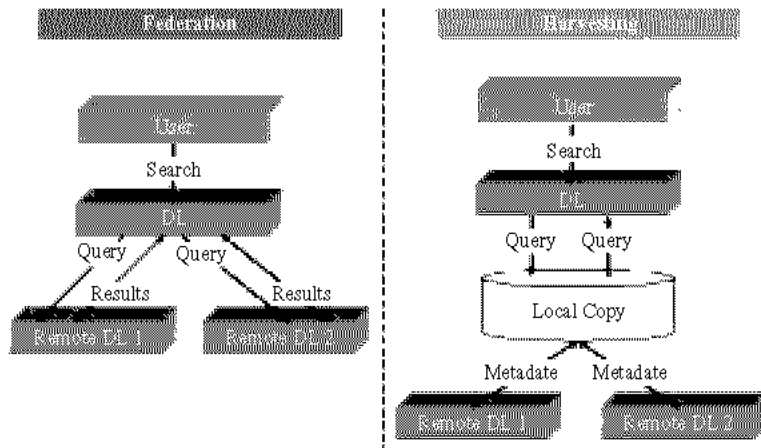
2004).

이들 요청에 응답하는 XML 인스턴스는 3가지 파트로 구성되어 있다. 첫째, 헤드부분(header)으로 공통정보 컨테이너이면서 고유한 식별자(identifier)와 수확일자에 사용되는 소인(datestamp)을 가지고 있다. 둘째, 메타데이터 본문(body)으로 하나의 아이템에 대한 메타데이터 컨테이너로 Dublin Core와 같은 메타데이터를 이용하여 본문 내용을 담고 있다. 셋째, 부가정보(about)를 기술하는 부분으로 참조정보를 가지고 있는 부분이다(이수상 2002).

OAI는 통합 검색을 위해 수확 검색(Harvesting Search) 방식을 사용하고 있으며, 서비스제공자가 데이터제공자로부터 메타데이터를 주기적으로 수확한 후 통합검색을 하게 된다. 이를 연합검색(Federation Search)과 비교하면 다음과 같다. 연합검색은 분산된 저장소에 질의를 보내 응답 결과를 모으고, 조

합한 후 보여주는 형식인데 반해 수확검색은 서비스제공자가 메타데이터를 수확하여 이를 로컬 저장소에 저장(Local Copy)한 후 검색에 이용하는 방식이다. 연합검색이 네트워크나 검색엔진 측면에서 다소 고비용이다. 왜냐하면 각 저장소는 복잡한 검색 질의와 빠른 실시간 응답을 지원해야 하기 때문이다. 반면에 수확검색은 각 저장소가 개별적으로 디지털라이브러리의 요청에 따라 메타데이터를 전송할 수 있어 부담이 적다. 물론 여기서의 비교는 질의 회수, 메타데이터 건수, 네트워크 자원 등을 주 비교대상으로 하였으며 비교 요소에 따라 결과가 달라질 수 있다. 하지만 일반적으로 연합검색이 원격 저장소에 큰 짐으로 간주되는데 비해, 수확검색은 원격 저장소에 요구를 줄일 수 있으며 서비스제공자 자신의 프로세스에 집중할 수 있는 것으로 나타나고 있다.

〈그림 3〉은 연합검색과 수확검색의 작업 흐름



〈그림 3〉 연합검색과 수확검색의 작업흐름(Data Flow)

름도이다.

OAI에서는 기존 프로토콜을 개선하기 위해 OA X 프로젝트를 추진하고 있는데, 여기서는 2가지 목표를 가지고 있다. 첫 번째가 기존 메타데이터 교환뿐 아니라 원문(resource)까지 확장된 기능을 제공하는 것이며, 두 번째가 수확된 메타데이터나 원문의 적재(upload)와 관련된 표준을 제정하여 정보의 일치성(Same Set)을 확보하는 것이다(Tools and technology 2004).

## 2.2 OAI 프로토콜의 확장성

디지털라이브러리 설계에서 재사용성은 항상 관심의 대상이 되어 왔다. 소프트웨어 공학에서 컴포넌트화(componentization)가 힘을 받는 반면 디지털라이브러리에서는 널리 확산되지 못하고 있다. 지금까지 컴퓨터가 전 세계 인터넷 확산의 주역이 되어 왔듯이, 정보 공급자와 사용자 커뮤니티도 날로 증가하고 있다. 정보 공급자는 이용자가 이용할 수 있는 정보의 양적인 면에 공헌하고 있지만, 이용자는 자신이 원하는 정보가 어디에 있는지를 알고 싶

어하며 관련된 정보에 동시에 접근하고 싶어한다. 정보 저장소(Information Repository)에 대한 체계적인 관리는 디지털라이브러리 발전에 공헌하였으며, 이용자의 요구에 부응하기 위해 저장소를 병합(merge)하는 것은 상호용성 등 표준화(standardization)에 기여하였다. 최근 대다수 사람들의 관심은 OAI 프로토콜의 출현으로 상호용성과 관련된 후자에 초점이 맞추어져 있다(Van de Sompel 2004).

디지털라이브러리를 설계하는 전자는 표준화의 관점에서 그다지 큰 관심을 끌지 못하였다. OAI에 기반한 디지털라이브러리 컴포넌트 상호작용 프로토콜을 만들기 위해서는 무엇보다 어떤 것이 확장 가능하고 그렇지 못한지를 정확히 파악하는 것이 중요하다. <표 2>는 OAI 프로토콜을 확장성 측면에서 3가지 유형으로 분류한 것이다. 즉, 확장을 지원하는 것(Supporting), 누락된 것(확장을 위해 추가적인 작업이 필요한 것)(Missing), 확장이 금지된 것(Inhibiting).

### ① Set Organization(Supporting)

OAI 프로토콜은 셋을 사용하여 관련된 분야 별로 수집할 수 있다. 이것은 저장소로부터 레

<표 2> OAI 프로토콜의 확장성

Supporting	Missing	Inhibiting
① Set organization	⑤ Response-level containers	⑦ Harvesting granularity
② GetRecord access	⑥ Submission	⑧ DC requirement
③ Metadata containers		
④ Identification containers		

코드의 서브 셋(subset)이나 선택 범위지정을 통하여 쉽게 활용할 수 있다. 만약 셋이 동적으로 만들어 진다면, 임의의 실시간(run time) 질의에 반응하는 방법을 제공할 수 있다.

#### ② GetRecord Repository Access (Supporting)

OAI 프로토콜 요청기능 중 GetRecord와 ListMetadataFormats은 하나의 특별한 레코드에만 접근할 수 있는 기능이다. 따라서 대량의 검색방법에는 적합하지 않다. 하지만 이들을 결합하여 동적접근방식으로 구성하면 효과적으로 검색할 수 있다.

#### ③ Metadata Containers(Supporting)

OAI 프로토콜은 어떤 메타데이터의 포맷도 그것을 XML로 표현할 수 만 있다면 전송할 수 있다. 따라서 디지털라이브러리에서 작업흐름과 관련된 트랜잭션데이터(transactional data)도 OAI 저장소에 구축 가능할 것이다.

#### ④ Identification Containers(Supporting)

식별자(Identify)는 저장소에 관한 부가적인 정보(about)들이 저장되는 컨테이너를 지원한다. 이것을 확장하면 컨테이너가 지원 가능한 확장된 OAI 프로토콜(XOAI 프로토콜)은 어떤 것들이 있는지 알 수 있다.

#### ⑤ Response Level Containers(Missing)

OAI 레코드는 권리정보 같은 메타정보를 저장하기 위한 XML 컨테이너를 가지고 있다. 그러나 어떤 컨테이너도 응답을 만들 때 무슨 일(act)을 했는지를 체계적으로 전달하지는 못한다. 응답한 전체 레코드 수 등은 이들이 몇 개의

덩어리(chunk)로 분리될 때에는 응답정보에 없어서는 안 될 필수정보로 보인다.

#### ⑥ Submission(Missing)

OAI 프로토콜은 저장소의 노출을 지원하기 위해 설계되었다. 그러나 데이터가 추가(add)되기 위한 어떤 방법도 규정되어 있지 않다. 이것은 저장소에 레코드를 저장하기 위한 표준화된 방법이 없기 때문이다. 따라서 지금 당장 고려해야 할 보안 및 품질과 관련하여 많은 문제점이 있다. 하지만 통제된 단일 디지털라이브러리에서는 이것이 그다지 큰 문제는 아니다. 입력 데이터 처리과정의 도입을 통해 그것이 가능할 것이다.

#### ⑦ Harvesting Granularity(Inhibiting)

OAI 프로토콜은 데이터를 수확하기 위해 소인을 사용한다. 하지만 수확한 레코드의 입도(자세한 정도)가 날짜 수준이며 시간이 빠져있다. 때문에 수확자(harvester)는 새로운 레코드를 누락하지 않기 위해 최소한 하루를 중복적으로 가져오게 된다. 이것은 빠르게 콘텐츠가 증가하거나 빈번하게 변경되는 저장소에서는 문제가 될 수 있다. 타 솔루션에서는 보다 더 정교한 방법(날짜뿐만 아니라 시간까지도 고려된 입도)을 사용하고 있으며 이것이 더 간단하고 효과적인 것으로 주창되고 있다.

#### ⑧ Dublin Core Requirement(Inhibiting)

모든 메타데이터 레코드는 OAI 프로토콜과 이에 해당하는 Dublin Core 메타데이터 셋을 통해 전달된다. 저장소 간의 전체 수준의 상호운용성을 위해서는 문서뿐만 아니라 다양한 것



을 포함하여야 한다. 그런데 그것들을 Dublin Core element에 모두 매칭하는 것은 부적절하다(Hussein Suleman 2004).

위에서 언급된 것과 같이 기존 OAI 프로토콜 중 확장 가능한 것은 확장하고, 한계가 있는 것은 보완하여 이를 디지털라이브러리에 적용하기 위해 출현한 것이 XOAI MPH(이하 XOAI 프로토콜)이다. XOAI 프로토콜은 기본적으로 OAI 프로토콜에 기반하고 있으며 ODL 컴포넌트의 근간을 이루고 있다.

### 3. Open Digital Library 연구

#### 3.1 Open Digital Library 프레임워크

디지털라이브러리는 전통적으로 전산학(computer science), 문헌정보학(Library & Information Science), 그리고 네트워크 기반 정보시스템과의 교점에 있었으며, 이 세 분야의 상이성이 디지털라이브러리를 더욱 혼란스럽게 만들었다. 네트워크 기반 정보시스템은 인터넷 혁명과 함께 지속적으로 디지털라이브러리 구축에 참여하여 왔다. 이제 디지털라이브러리로 정보시스템 분야의 표준화를 기반으로 변화된 모습을 보여줄 때가 되었다.

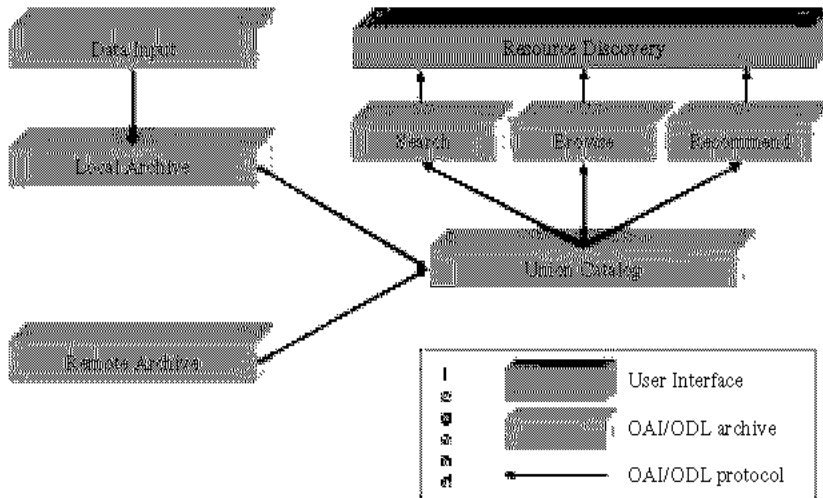
ODL을 하는 이유는 대부분의 디지털라이브러리들이 라이프사이클에 의해 조만간에 다치게 될 문제, 즉 상호운용성, 콘텐츠 확장, 편리한 관리 등과 직결된다. 이 문제를 더욱 심화시키는 것은 데이터 교환과 서비스 협력을 위해서

는 시스템적으로 지원이 가능하여야 한다는 것이다. 이와 같은 상호운용성 요구는 Dublin Core 메타데이터나 OAI 프로토콜 같은 표준의 개발을 통하여 가능하다. 이 같은 표준은 보편적이며 간단함으로 인해 디지털라이브러리 분야에서는 이미 널리 인정받고 있다. 확장성 있는 디지털라이브러리를 구축하기 위해서는 상호운용성과 관련 있는 표준들을 끊임없이 개발하여 컴포넌트 기반의 프레임워크를 구축하여야 한다.

디지털라이브러리는 개념 자체가 모호하다 보니, 단지 접근하기 쉬운 정보 저장소로 이해하려는 경향이 있다. 게다가 이 분야는 표준이나 기술을 한 곳에 모으기가 어렵다. 현존하는 대부분의 디지털라이브러리 시스템은 각 사이트에서 필요에 의해 제각각 개발된 것이다.

OAI 프로토콜은 대규모 디지털라이브러리(Larger Digital Library) 측면에서 보면 컴포넌트들을 서로 연결하는 연결고리라 볼 수 있다. 이들 연결고리에 의해 <그림 4>와 같이 확장된 오픈 저장소(Extended Open Archives) 네트워크 체제가 구축되는데 이것을 ODL이라 한다.

ODL이 가속화된 이유는 모든 서비스가 확장된 오픈 저장소에 의해 수행된다면 각 디지털라이브러리 시스템 내에 자동적으로 컴포넌트화(componentization)와 표준화(standardization)가 이룩될 것이며, 이들 컴포넌트화와 표준화는 디지털라이브러리내의 각 서비스 수준의 상호운용성과 재사용성을 지원 할 것이



〈그림 4〉 Open Digital Library 아키텍처

기 때문이다.

ODL 접근법은 도서관 업무와 매우 유사하다. 즉, 도서관의 각 시스템들은 자신들의 커뮤니티 내에서는 공동이용이 가능하다. 예를 들면 구매담당부서에서는 서적상(book sellers)과 공동이용이 가능하며, 상호대차 부서에서는 다른 도서관의 상호대차 부서와 정보공유가 가능하다. 상호운용성은 위에서 보듯이 기관 전체보다는 각 서비스 계층(level)에서 더 잘 달성될 수 있다.

연구 디지털라이브러리(Research DL)와 제품 디지털라이브러리(Production DL) 간에는 기술적인 측면에서 큰 차이가 있다. 전자는 실험적인 연구에 중점을 두고 있지만 후자는 이용자들이 원하는 실질적인 사항(Real Issues)에 비중을 두고 있다. 이들 두 시스템을 결합하는 것이 그렇게 간단하지는 않지만, 두 시스템이 표준화 되어 있다면 손쉬울 것이다. 이를 해결

해 주는 열쇠가 바로 OAI 프로토콜이다. 디지털라이브러리 서비스를 모델링 하는 것은 인터넷과 같이 간단하게 하기 위해서이다. 간단한 모듈은 신규로 발생되거나 개선되는 프로세스를 처리하는 과정을 매우 간결하게 할 것이다.

ODL의 궁극적인 목적은 무한대로 확장 가능한 시스템을 구축함으로써 더 많은 이용자에게 더 많은 정보에 대한 접근성을 제공하기 위함이다. ODL에 적용된 사상은 간단한 프로토콜, 개방적인 표준, 의미적인 표현, 컴포넌트의 독립성, 느슨한 시스템 연결, 그리고 어디에서든 재사용 가능한 시스템을 설계하는 것이다. 명목상, 이런 원칙은 다음과 같이 규정된다.

- ① 오픈 저장소의 확장성에 사용되는 모든 디지털라이브러리 서비스는 컴포넌트로 캡슐화된다.
- ② 디지털라이브러리 서비스에 접근하는 모든 서비스는 확장된 OAI 인터페이스(XOAI 프

로토콜)를 이용해야 한다.

- ③ OAI 프로토콜은 계속 확장되겠지만 이 프로토콜의 본래 의미에 모순됨이 없어야 한다.
- ④ 모든 디지털라이브러리 서비스는 확장된 OAI 프로토콜을 사용하여 다른 데이터에 접근할 수 있어야 한다.
- ⑤ 디지털라이브러리는 확장된 오픈 저장소 네트워크처럼 건설되어야 한다.

각 개별 사이트의 디지털라이브러리 서비스는 자급자족(Self Contained)의 컴포넌트처럼 구축되지만 다른 것들과의 커뮤니케이션은 확장된 OAI 프로토콜을 사용한다.

디지털라이브러리는 느슨하게 정의된 파라미터와 항상 변하는 요구사항을 지닌 모험적인 소프트웨어 분야이다. 디지털라이브러리에 종사하는 많은 사람들에 의해 최근 적용된 대부분의 솔루션은 OAI 프로토콜이다. 이 프로토콜은 간단한 표준이 필요한 곳에 즉각적인 영향을 미치며, 기존 존재하거나 새로 생기는 많은 저장소에 표준 채택을 선도할 것이다.

OAI 프로토콜은 단순한 개념으로 인해, 이것의 확장성은 시스템 개발자들에게 큰 영향을 줄 수 있는 잠재성을 지닌다. 아울러 디지털라이브러리의 각 컴포넌트에서 저장소로 접근 역할을 수행할 것이며, 이들 컴포넌트들이 독립적인 데이터제공자와 서비스제공자에게 느슨하게 연결된 네트워크 형태로 시스템이 통합될 때 ODL 체제가 구축되는 것이다. 이런 방식은 밀접합된 시스템(Tightly Coupled Systems)에 비해 단순성, 재사용성, 확장성에서 뛰어나다.

종합하면, ODL은 분산되어 있는 오픈 저장소를 표준화된 프로토콜(OAI/XOAI)로 연결함으로써 확장된 디지털라이브러리를 구축하는 것으로, 기본 사상은 OAI의 재사용성과 간편성에 시스템 확장성과 컴포넌트 개념을 추가한 것이다(Hussein Suleman 2001).

### 3.2 Open Digital Library 설계 주요점

ODL 프로젝트의 목적은 데이터 서비스와 같이 그렇게 간단한 방법으로 서비스를 제공하는 디지털라이브러리 아키텍처를 연구하는 것이다. 그렇게 함으로써 디지털라이브러리의 상호운용성, 관리, 개발과 관련된 장애물(bottleneck)을 제거하는 것이며, 결국 새로운 정보시대에 누구나 정보를 공유할 수 있게 하자는 것이다.

ODL은 정보 이용자에게 서비스를 제공하기 위한 확장된 공용의 오픈 저장소 네트워크이다. 네트워크안의 각 노드를 ODL 서비스라 한다. ODL은 소프트웨어공학 중 뛰어난 최신 기법을 사용하여 시스템적인 방법으로 디지털라이브러리를 건설하기 위해 설계된 프레임워크로 어찌 보면 모듈화된 디지털라이브러리를 구축하기 위한 것이다. 또한, 비정상적인 메타데이터 수확(불필요한 메타데이터 수확)과 관련 있는 부분의 더 큰 기능성과 응용성을 지원하기 위해 OAI 프로토콜의 확장에 기반하고 있다.

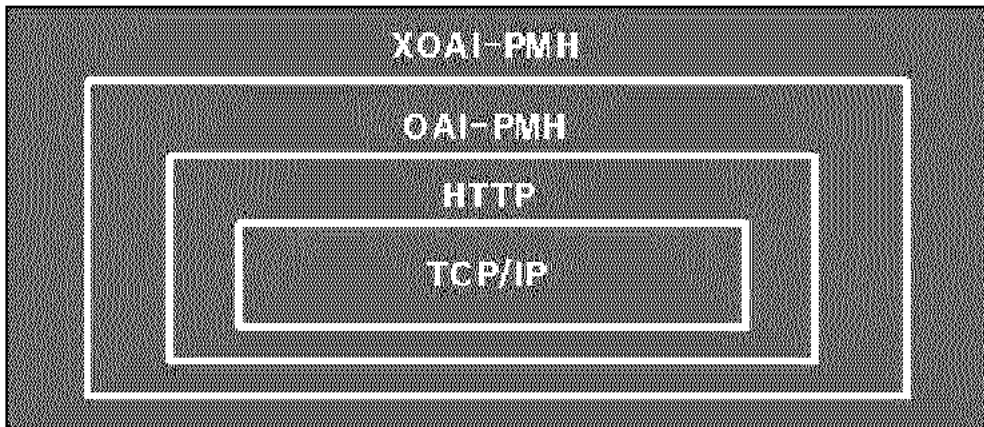
OAI 프로토콜은 HTTP 프로토콜의 많은 설계 방법론을 적용했다. 파라미터는

CGI(Common Gateway Interface)를 통해서 전달되고 에러는 HTTP 프로토콜 방법을 통해 반환된다. 거기에는 미리 정의된 HTTP 에러 메시지가 있으며, 가능한 한 OAI에서는 HTTP 에러 메시지를 사용할 것이다.

〈그림 5〉는 네트워크 프로토콜 간의 계층 관계를 표현한 것이다. 이를 통하여 각 계층 간 합축적인 상속원리를 추론할 수 있다. 즉, 계층적

인 방법은 원래의 표준으로 덤다운(Dumb Down)이 가능하므로 표준을 확장할 수가 있는 것이다. ODL 프로토콜(XOAI)은 OAI 프로토콜로 덤다운이 가능하며, 자동적으로 HTTP 프로토콜로도 가능하다.

다음은 ODL 설계 시 주안점을 요약한 것이다.



〈그림 5〉 네트워크 프로토콜 계층도

① 간결성(simplicity)

인터넷이 강력한 힘을 발휘하는 것은 무엇보다도 간결성이다. 이와 같은 방법으로 건설된 각 컴포넌트는 단일하고 완전하며 독립적인 존재이다. 많은 새로운 프로토콜이 완전한 솔루션으로 개발되지만 그들은 기술의 부족뿐만 아니라 새로운 복잡한 것을 거부하는 고객들의 반대로 대부분 실패한다. OAI는 이런 문제를 극복하기 위해 출범했으며, 주요 임무 중 하나가 바로 장벽이 낮은 상호운용성을 건설하는 것이다.

따라서 ODL은 디지털라이브러리 컴포넌트 커뮤니티의 필요에 부합하기 위해 장벽이 낮은 전통을 계승 발전시켜야 한다.

② 개방성(Openness)

ODL은 오픈 표준(Open Standards)으로 설계되어야 하며, 그 자신이 네트워크 커뮤니티로부터 최대의 이득을 얻기 위해서라도 비평 등을 겸허히 수용하여 프로그램 개발에 반영하여야 한다.

③ 프로토콜의 독립성(Independence of

Protocols)

클라이언트 프로그램으로 원격 메일박스에 접근할 수 있는 매우 유명한 인터넷 프로토콜로 POP과 IMAP가 있다. 이들은 TCP/IP 계층에 기반하고 있으며, 이들은 완벽하게 분리된 프로토콜이기 때문에 다른 것들과 동시에 배치, 개발, 설치를 위한 어떤 요구나 의무도 없다. 또한, 이들 프로토콜들은 다른 것들의 독립된 개발을 허용하는 중요한 개념을 담고 있으며, 서로 다른 목적을 위해 다른 서비스가 각기 독립적으로 개발된다. ODL은 서비스 독립 사상을 채택하여, 다른 것을 지원하기 위한 어떤 타협도 없이 각 서비스의 품질을 최대화하기 위해 모든 서비스는 완전하게 분리되어야 한다.

#### ④ 느슨한 연결(Loose coupling)

느슨한 연결(loose coupling)이란 간단히 말해서, 어느 한 부분이 다른 부분에 지나치게 의존적이 되지 않게 하는 것이다. 예를 들자면 전역변수를 많이 사용하지 말자는 것도 이 개념에 해당된다. 전역변수에 의존하면 서로 의존적인 코드가 많아져서 어떤 부분을 바꿀 때 여러 부분을 동시에 수정해야 하는 상황이 발생하기 때문이다. 시스템 개발을 프로그램 개발에서 독립적인 컴포넌트의 구성으로 바꾸기 위해서는, 느슨한 연결과 잘 정의된 프로토콜에 기반한 모듈 개발을 통하여 컴포넌트 단계로 발전해야 한다. 이런 방식은 이미 잘 정의된 프로토콜을 통하여 커뮤니케이션하는 인터넷상의 유명한 서비스에서 증명되었다. 이처럼 컴포넌

트화는 ODL의 본질이다.

#### ⑤ 계층(Layers)

혼합된 네트워크 프로토콜로부터 잘 이해할 수 있는 시스템으로 진화한 컴퓨터 네트워킹이 TCP/IP이다. TCP/IP를 통하여 계층 구조가 큰 장점이 있다는 것은 이미 검증되었다. ODL은 프로토콜과 함께 계층적인 접근을 통하여 OAI 또는 다른 ODL 프로토콜의 확장을 통하여 개발한다. 이런 방식은 다른 목적을 위해 HTTP에 대한 기본 개념의 재정의 없이 HTTP와 함께 프로토콜을 설계하는 것이 가능하다. ODL 컴포넌트는 OAI 프로토콜을 확장하는데 이와 같은 사상을 적용할 수 있다

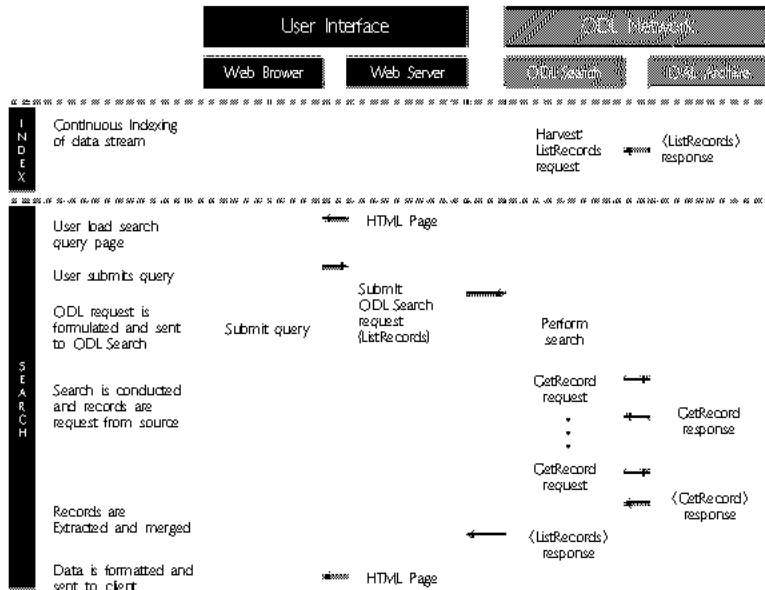
#### ⑥ 재사용성(Reuse)

재사용성은 소프트웨어와 설계 방법론에서 매우 중요하다. ODL 컴포넌트는 가능하면 이전의 컴포넌트를 재사용하여 설계한다.

#### ⑦ 목적과 직교성(Orthogonality with a Purpose)

직교성(Orthogonality)은 무선 인터넷 기술에서 신호가 서로 직교해 간섭을 최소화하는 것이다. 학문적인 관점에서 직교성은 완벽한 시스템을 개발하는데 도움을 준다고 한다. 그러나 이것이 항상 효율적인 것만은 아니다. 몇몇의 경우는 실용적인 측면에서 불필요하다. ODL 설계는 이와 같은 직교성을 따르겠지만 완벽한 시스템 설계보다는 실제 이용자 관점에 집중하여 설계한다.

### 3.3 Open Digital Library 컴포넌트 사례



〈그림 6〉 ODL-SEARCH 컴포넌트 처리절차

ODL 시스템에 대한 이해를 돕기 위해 〈그림 6〉는 실제로 구현된 컴포넌트인 ODL Search 처리 절차를 도식화한 것이다. 이용자 인터페이스 계층은 클라이언트의 웹 브라우저, 웹 서버, 스크립트(HTML 폼에 있는 데이터를 ODL 네트워크에 제출하는 기능)로 이루어져 있다.

ODL Search 컴포넌트는 데이터색인과 검색 두 가지 기능을 갖고 있다. 전자는 ODL Search 컴포넌트가 수확자를 이용하여 데이터 제공자로부터 정보를 수집하는 것이다. 예를 들면 새로운 것이나 갱신된 것을 수확하기 위해 특정 기간을 주고 주기적으로 수집하는 ListRecords 요청이다. 후자는 검색기능으로 검색을 하기 위해 이용자가 HTML 폼에 질의어를 입력하여 제출하는 것이다. 이 질의어는 웹 서버에 전달되며, 그러면 이것을 수행하기

위해 스크립터(핸들러)를 기동하게 된다. 스크립트는 파라미터를 추출해 내고 ODL Search ListRecords를 활성화하며, 추출된 것을 ODL Search 컴포넌트에 제출하게 된다. 요청 받은 컴포넌트는 내부 색인을 이용하여 검색을 수행하고 원천 OAI 저장소로부터 각 매타데이터 레코드를 순차적으로 수집한 후 병합하여 단일의 ListRecords 반응과 같이 스크립터에 되돌려 준다. 스크립트는 이것을 포맷한 후 검색 결과를 HTML 폼으로 이용자에게 전달함으로써 프로세스가 종료된다. 이것은 검색부터 결과를 얻는 과정까지를 표준화하고 컴포넌트화한 대표 사례이다.

위의 ODL SEARCH 컴포넌트에서 사용된 XOAI 프로토콜 중 ListRecords와 확장되기 전 OAI 프로토콜인 ListRecords를 비교하면

다음과 같은 차이가 있다. 먼저 전자는 검색어를 프로토콜에 포함시킴으로써 검색에 기반한 실질적인 메타데이터 수집인 반면에, 후자는 특정 기간에 속한 데이터를 일괄 수집함으로써 경우에 따라서는 불필요한 정보까지도 수집되는 방식이다. 게다가 검색 기능은 각 사이트에서 알아서 처리하도록 하고 있다. 이와 같은 방식은 장단점이 있게 마련이다. 하지만 검색은 필수기능이고 궁극적으로는 검색을 통해 고객들에게 원하는 정보를 제공하는 것이 목적이고 보면, 표준화된 일괄 방식을 제공하는 것도 매우 중요하다고 보여진다.

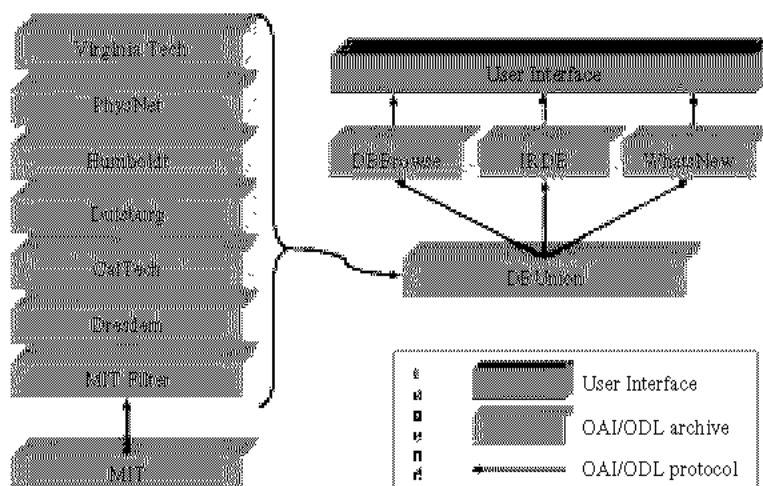
#### 4. Open Digital Library 적용사례

##### 4.1 NDLTD ODL

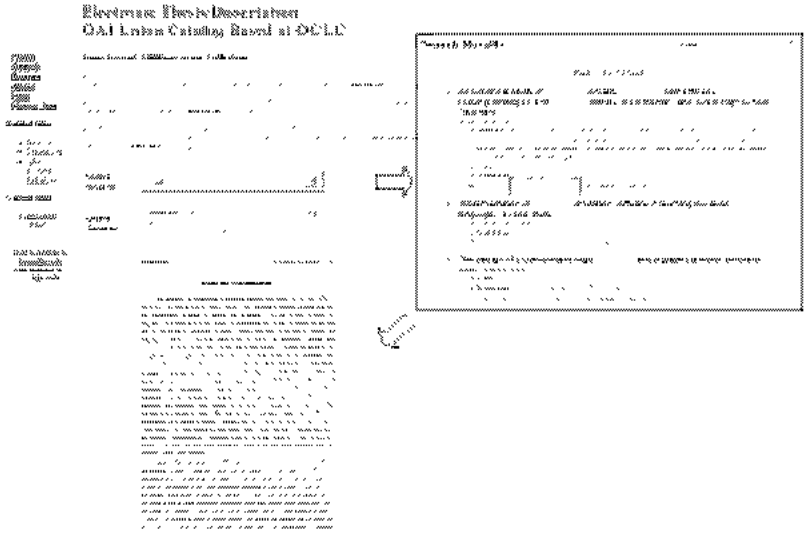
버지니아텍 대학에서는 개발된 ODL 컴포

넌트(ODL Union, ODL Filter, ODL Search, ODL Browse, ODL Recent)를 사용하여 기존 NDLTD의 일부 시스템을 교체하였다. 각 컴포넌트들은 개별적으로 구현되었으며, 이들이 ODL 스펙에 부합하는지를 검증하기 위해 Repository Explorer(시험 도구)를 이용하여 테스트 하였다. 아울러 기존 NDLTD에 의해 관리되던 메타데이터의 연합 저장소 범주를 넘어 새로운 디지털라이브러리 서비스를 실현하기 위해 ODL 네트워크를 구축하였다 (Suleman 2001). 또한, 재사용성을 테스트하기 위해 몇몇 컴포넌트는 대학 내에 있는 CSTC(Computer Science Teaching Center)의 개발서버에 적용했다.

〈그림 7〉은 NDLTD ODL 아키텍처이며, 여기에는 OAI 프로토콜 인터페이스를 통해 메타데이터를 수확하는 7개의 저장소가 있다. 그중 하나는 다른 저장소와는 상이하기 때문에 필터



〈그림 7〉 NDLTD ODL 아키텍처



〈그림 8〉 NDLTD ODL 시스템 사용자 인터페이스

(filter)를 사용하였으며, 7개 저장소 데이터들은 통합검색 등 부가가치 서비스를 위해 로컬 중앙저장소로 수확 된다. 그리고 이들 자료에 대한 검색, 브라우저, 최신자료를 서비스하기 위해 3개 층의 응용 서비스(three high level services)가 제공된다. 검색은 데이터 색인과 질의어에 합당한 자료를 응답하기 위해 OAI 인터페이스(OAI like interface) 기능을 사용하며, 브라우저 기능은 데이터를 정렬하고 통제된 어휘에 의한 아이템 접근을 위하여 OAI 인터페이스와 약간 다른 형태의 인터페이스를 제공하고 있다. 최신정보(recent)는 최근 아이템을 정렬한 후 그들 중 임의의 정보를 보여 주는 방식을 채택하고 있다. 마지막으로 이용자 인터페이스(UI)는 HTML 폼의 데이터를 ODL 요청으로 변환하기 위해 최소한의 스크립트를 사용하였

으며, XSL 스타일시트를 이용하여 사람들이 가독하기 힘든 XML 문서를 HTML 형태로 변환하였다.

현재 NDLTD에는 188개 대학(7개 협회포함), 28개 기관·단체 등 총 216개의 회원사가 참여하고 있는 Open Digital Library이다.

〈그림 8〉은 NDLTD ODL 시스템(<http://purl.org/net/etdunion>)의 사용자 인터페이스(UI)이다. 여기에 적용된 기능은 OAI 프로토콜 확장에 잘 부합하기 때문에 향후 확장된 프레임 워크 설계에 재사용 가능하다.

저장소 내에 불필요한 기능을 줄이는 것이나 속도를 증가시키는 기술들은 이미 검증되었다. 보다 더 큰 속도 향상은 FastCGI나 Speedy CGI 등을 통해 달성 가능할 것이며, 메타데이터 증가에 따른 저장매체 증가는 메타데



이더 저장소 간 저장매체 공유를 통한 주문형 (on demand) 방식으로 해결 가능할 것이다.

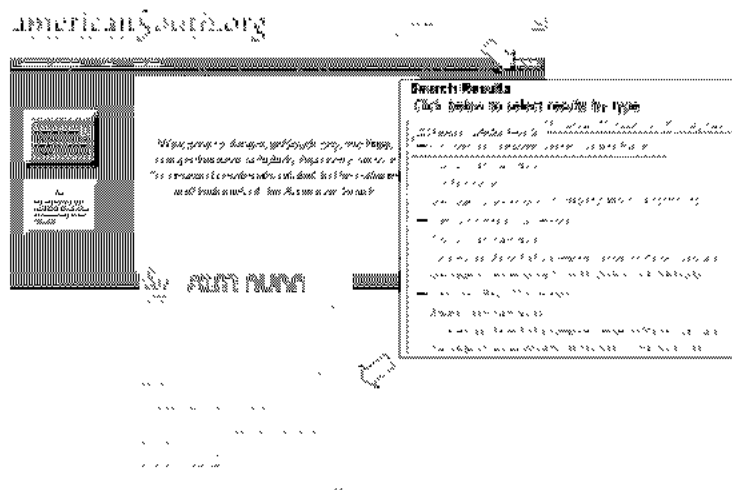
#### 4.2 americanSouth.org

americanSouth.org는 2001년 Mellon Metadata Harvesting Initiative(MMHI)의 7 가지 프로젝트 중 하나로 Emory 대학에서 수행한 프로젝트이다. 주요 목적은 연구자원에 대한 메타데이터의 색인과 특정 분야(남미지역의 인문, 사회, 자연 과학, 역사학)에 대한 가치 있는 자원 탐색 매커니즘을 지원함으로써 디지털 자원에 대한 접근성을 제고하는 것이다.

이를 위해 도서관과 박물관으로부터 해당기관에 대한 정보를 수집하고 있다. 아울러 통합, 색인, 검색을 위해 참여기관으로부터 주기적으

로 메타데이터를 수확하여 중앙집중형 메타데이터를 구축하고 있으며, 대량의 정보자원에 대한 메타데이터의 질적 개선을 위해 MetaCombine 프로젝트를 별도로 수행하고 있다. 현재 americanSouth.org에는 36개의 기관이 참여하고 있으며, 이 기관들은 기본적으로 메타데이터를 가지고 있으며 OAI 프로토콜에 기반한 데이터제공자 역할을 수행하고 있다.

americanSouth.org에서는 OAI 프로토콜 기반의 메타데이터 수집 및 통합검색과는 별도로 관련 분야에 대한 Weblink 서비스를 하고 있다. Weblink는 프로토콜에 의해 수집을 하는 것이 아니고 해당 사이트에 대한 단순한 링크 (Related Link)만을 제공한다. 비록 프로젝트는 종료(2004년 1월)되었지만 계속적으로 남미 지역의 인문, 사회, 자연 과학, 역사학에 대한



<그림 9> americanSouth.org 검색결과 화면

메타데이터를 수집·서비스 하면서 Weblink 서비스를 발전시킬 계획을 가지고 있다 (americanSouth.org 2004).

다음은 American South.org의 검색결과 화면으로, 홍콩대학 등의 콘텐츠를 검색하여 전자원문을 획득하는 과정을 보여주고 있다.

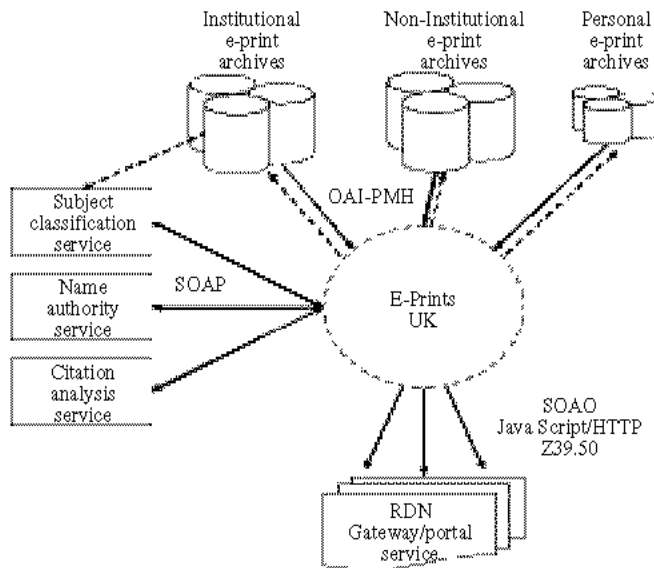
### 4.3 ePrints UK

ePrints UK는 ODL 개념보다는 보다 더 확장된 포괄적(comprehensive)이고 통합적인(integrated) 디지털라이브러리 사례로 국가적인 지원 하에 현재 프로젝트가 진행 중에 있다.

영국 JISC에서는 고등교육과 평생교육의 콘텐츠에 대한 접근성 제고를 위해 5개 클러스터(clusters), 14개 프로젝트로 구성된

FAIR(Focus on Access to Institutional Resources) 프로그램을 운영하고 있는데, ePrints UK는 FAIR 프로젝트 중 하나로 저장소에 대한 접근성 제고와 수집 콘텐츠에 대한 품질 제고가 목적인 서비스제공자 프로젝트이다.

ePrints UK 프로젝트는 King's College London의 RDNC(Resource Discovery Network Centre)와 Bath 대학의 UKOLN(UK Office for Library and Information Networking)이 주도하고 있다. 본 프로젝트는 기관, 훈련소, 그리고 개인 데이터제공자로부터 메타데이터를 모아 서비스제공자 역할을 하는 것이다. 서비스는 RDN 허브나 다른 인프라를 활용하고 있다. 기관 자산(Institutional Assets)인 조직 내에서 생산된 정보 자원은 매우 이용하기 어려웠으며, 특히



〈그림 10〉 ePrints UK 아키텍처

그 조직 밖의 이용자에게는 더욱 심하였다. FAIR 프로그램에서는 기관 자산을 외부에 오픈하기 위해 메타데이터를 수확함으로써 조직 간 정보 공유가 가능한 형태인 OAI 프로토콜을 채택하였다.

ePrints UK는 웹 서비스 기술을 활용하여 허브 사이트로부터 메타데이터를 수확하는 기능을 대폭 강화하였다. 아울러 웹 서비스 프로토콜(SOAP)을 사용하여 자동 주제 분류(Automatic Subject Classification), 표목(Authority Headings), 그리고 OpenURL을 이용한 인용색인 분석을 연구 중에 있다.

웹 서비스 기술을 활용한 강화된 수확 기능은 데이터제공자의 메타데이터 품질 제고에도 매우 필요하다. 데이터제공자는 가능하면 통제된 어휘를 사용하려 할 것이며, 서비스제공자가 이것을 잘 활용할 수 있도록 정확한 메타데이터를 제작하길 원할 것이다. 지금까지의 연구결과는 서비스제공자가 품질제고를 위해 메타데이터를 수집하는데 까지 연구가 진행되었으며, 보완된 정보를 데이터제공자 메타데이터 레코드에게 되돌려주는 기능은 아직 연구 중에 있다. 다음은 ePrints UK 아키텍처를 간략히 도식화한 것이다.

ePrints UK는 국가적인 사업으로 정부가 자금을 지원하고 있으며 최종 목표는 전 국민을 대상으로 누구나 자유롭게 교육 및 학술 관련 정보를 이용할 수 있는 체제를 구축하는 것이다. 아울러 이전에 OAI 저장소가 양적·질적인 측면에서 이루지 못했던 연구를 달성하는 것이

다. 이런 노력을 통하여 정보전문가와 연구자들이 점차 다양한 OAI 프로그램과 저장소가 있다는 것을 알게 됨으로써, 기존 학술 커뮤니케이션과는 아주 다른 것을 창조하게 될 것으로 전망하고 있다(Ruth Martin 2004).

참고적으로 JISC(The Joint Information Systems Committee)는 영국 내 대학교육(HE)과 성인교육(FE)에 대한 재정기관을 대표하는 정보화 정책자문 위원회로, 대학 및 성인교육 기관이 필요로 하는 정보시스템 및 기술을 지원하는 역할을 하고 있다. RDNC의 RDN은 다양한 주제(Subject)별로 양질의 인터넷 정보를 모아 놓은 곳으로 학습, 교수, 연구 커뮤니티의 인터넷 자원을 위한 무료국가관문시스템(Free National Gateway)이다. 이 서비스는 주제기반 관문(hubs) 시리즈를 통하여 8만여 이상의 자원과 연계하고 있다. RDN은 우선적으로는 영국 내 대학교육과 성인교육의 인터넷 사용을 위한 것이지만 궁극적으로는 전 국민이 무료로 이용하기 위한 것이다. RDN은 참여기관의 주제 분야별 전문가를 통하여 인터넷 자원을 모은다. RDN은 70여개의 교육 및 연구기관이 참여하는 공동 연구 성과이며 JISC의 지원을 받아 수행되는 주제별 관문 활동의 토대를 구축하는 것이다.

## 5. 결론

지금까지 Open Digital Library의 프레임워크와 적용사례 등에 대해 알아보았으며, 아울

러 이들의 사상적 기반이 된 OAI에 대해 연구하였다. OAI에 기반 한 디지털라이브러리(오픈 저장소)는 급속한 팽창기에 있으며, 이런 추세는 학술정보의 자유이용(Open Access) 패러다임과 국가적인 지원 체제에 따라 더욱 가속화될 것이 분명하다.

과거 자신들의 필요에 의해 제각각 구축된 디지털라이브러리는 곧 한계에 봉착하게 될 것이며, 끊임없는 요구에 직면할 것이다. 이제 전통적인 디지털라이브러리 시스템들도 표준과 상호운용성에 기반한 간결한 시스템으로 거듭날 때가 되었다.

OAI에 기반 한 디지털라이브러리는 어느 정도 필요성과 우수성이 검증된 반면에 ODL은 이제 막 태동 단계에 있다. 물론 이들의 기저가 같고 ODL이 OAI 프로토콜의 미비점을 보완하고 더욱 확장하며 컴포넌트화하는 것이고 보면, 둘은 경쟁관계이기 보단 상호 보완적인 관계라 볼 수 있다. 이런 맥락에서 ODL의 근간 프로토콜인 XOAI를 최근 OAI 표준화 위원회에 제출하여 함께 가고자 하는 노력은 의미하는 바가 크다 하겠다.

현재 ODL에서 다루고 있는 분야는 아주 기본적인 기능 위주로 설계되고 구현되고 있다. 차세대 디지털라이브러리로 자리매김하기 위해 선 한걸 개선되고 발전되어야 할 것이다.

ODL의 발전과 확장성을 위해 고려되어야 할 것으로는 다음과 같은 것이 있다. 첫째, ODL은 프로토콜에 기반 한 표준화된 시스템이다. 따라서 차세대 웹 서비스 표준 프로토콜인 SOAP과

어떤 형태로든 연계되어야 한다. 둘째, 현재 메타데이터 위주로 시스템이 설계되고 개발되고 있지만 궁극적으로 전자원문 획득이기 때문에 이 분야 국제표준 프로토콜인 OpenURL적용을 적극 고려하여야 한다. 셋째, OAI뿐만 아니라 NON OAI에 대한 고려도 있어야 한다. 왜냐하면 아직까진 NON OAI가 상당수를 차지하고 있기 때문이다. 점차 감소하고는 있지만 OAI와 공존할 수밖에 없는 상황을 인정해야 하기 때문이다. 넷째, 저장소별로 제각기 구축된 메타데이터를 통합하다보면 분류 체계 등에 문제가 있을 수 있다. 이를 위한 대안으로 자동 문서분류를 위한 기계학습 모델인 SVM의 적용을 고려해 볼만하다.

NDLTD ODL에서 보듯 실험실 수준이었던 ODL을 제품 디지털라이브러리까지 발전시킨 것은 매우 고무적인 현상이다. 네트워크화 된 거대한 학술정보 자유이용 체제 구축과 디지털라이브러리 표준화 차원에서 ODL에 대한 연구와 관심이 필요한 시점이다.

## 참고문헌

- 김정아. 2002. 『학술커뮤니케이션의 변화와 E print 아카이브에 관한 연구』, 석사학위논문, 충남대학교 대학원, 문헌정보학과.
- 이수상. 2003. 메타데이터 상호운용성 보장을 위한 요소기술. 『한국도서관정보학회지』, 34(1): 91-109.

- 이수상, 2002. 전자도서관의 최근동향. 『데이 터베이스연구』, 18(3).
- americanSouth.org. [cited 2004.9]. [http://www.arl.org/scomm/open\\_access/](http://www.arl.org/scomm/open_access/).
- Edward A. Fox, Hussein Suleman. "Designing Protocols in Support of Digital Library Componentization" [cited 2004.8]. [http://oai.dlib.vt.edu/odl/pubs/ecdl\\_2002\\_odl\\_paper\\_revised.pdf](http://oai.dlib.vt.edu/odl/pubs/ecdl_2002_odl_paper_revised.pdf).
- Hussein Suleman, Edward A. Fox, 2001. "A Framework for Building Open Digital Libraries" *D Lib Magazine*, 7(12). <http://www.dlib.org/dlib/december01/suleman/12suleman.html>.
- Hussein Suleman, Edward A. Fox. "Beyond Harvesting: Digital Library Components as OAI Extensions" [cited 2004.8]. [http://oai.dlib.vt.edu/odl/pubs/cstr\\_2002\\_odl\\_1.pdf](http://oai.dlib.vt.edu/odl/pubs/cstr_2002_odl_1.pdf).
- hussein's space. [cited 2004.8]. <http://www.husseinsspace.com/research.htm>.
- Integration of Non OAI Resources for Federated Searching in DLIST. [cited 2004.8]. <http://www.dlib.org/dlib/july04/coleman/07coleman.html>.
- Issues in Scholarly Communication. [cited 2004.8]. [http://www.arl.org/scomm/open\\_access/](http://www.arl.org/scomm/open_access/).
- Open Digital Libraries. [cited 2004.8]. <http://oai.dlib.vt.edu/odl/>.
- OpenURL Syntax Description. [cited 2004.8]. [http://www.exlibris-group.com/sfx\\_openurl\\_syntax.htm](http://www.exlibris-group.com/sfx_openurl_syntax.htm).
- Ruth Martin, "ePrints UK: Developing a national e prints archive" [cited 2004.8]. <http://www.ariadne.ac.uk/issue35/martin/>.
- Stephen Pinfield, 2003. "Open Archives and UK Institutions An Overview" *D Lib Magazine*, 9(3). <http://www.dlib.org/dlib/march03/pinfield/03pinfield.html>.
- The Open Archives Initiative Protocol for Metadata Harvesting. [cited 2004.8]. <http://www.openarchives.org/OAI/openarchivesprotocol.html>.
- Tools and technology for Open Repositories. [cited 2004.8]. <http://www.itor.org/>.
- Van de Sompel, Lagoze. "The Open Archives Initiative Protocol for Metadata Harvesting" [cited 2004.8]. <http://www.openarchives.org/>.