

# 학술정보 유통에 있어 OAI 프로토콜의 적용에 관한 연구

## A Study on the Practical Implementation of a OAI Protocol for Scholarly Communication

이 수 상(Lee Soo-Sang)\*

### <목 차>

- |   |               |
|---|---------------|
| I. 서론                                     | 1. 시스템의 구성 모델 |
| II. OAI 프로토콜의 이해                          | 2. SP의 구현 내용  |
| 1. 등장배경                                   | 3. DP의 구현 내용  |
| 2. 기본개념                                   | 4. 차이점 분석     |
| 3. SP/DP의 구성요건                            | IV. 결론        |
| III. OAI 프로토콜의 적용 사례분석 :<br>KERIS의 생성유통체계 |               |

### 초 록

OAI는 디지털도서관의 상호운용성을 위해 개발된 프로토콜이자 철학적 기반이다. 이는 메타데이터의 수확을 위한 프로토콜로서, 분산환경의 디지털도서관의 효율적인 접근과 정보접근 역량을 제고시켜 학술정보의 효율적인 유통을 보장하기 위하여 개발되었다. 따라서 본 논문은 OAI 프로토콜의 등장배경, SP/DP 모델, 관련된 기술 및 구성요건을 살펴보고, 국내에서 처음으로 시도된 KERIS의 적용사례와 OAI에서 권장하는 기본요건과의 차이점을 비교분석하고 있다.

주제어 : OAI 프로토콜, 디지털도서관, 학술유통, SP/DP 모델

### Abstract

The OAI(Open Archives Initiative) is an evolving protocol and philosophy regarding interoperability for digital libraries. This protocol has its roots in an effort to enhance access to digital libraries as a means of increasing the availability of scholarship communication. Thus, this research outlines the history of the OAI protocol, the basic model of SP(service provider)/DP(data provider), and the prerequisites to implement the OAI protocol. Viewed in this light, this paper also describes the experiences of KERIS project of SP/DP model and lists the differences between prerequisites and KERIS case.

Key Words : OAI Protocol, digital library, scholarly communication, SP/DP model

\* TG인포넷 정보기술연구소 부장(sslee@tginfonet.com)

· 접수일 : 2004. 5. 28 · 최초심사일 : 2004. 6. 11 · 최종심사일 : 2004. 6. 11

## I. 서 론

1991년 미국 LANL(Los Alamos National Laboratory)에서 연구원으로 근무하던 징스파르그(Paul Ginsparg)에 의해 고에너지 물리학분야의 디지털 학술논문(eprints)을 관리하고 서비스하는 아카이브(archive)인 arXiv<sup>1)</sup>이 개발되었다. 이 arXiv의 탄생은 이후 디지털도서관 분야에 중요한 영향을 미친 사건이었다. CogPrints(심리학), NCSTRL(전산학), RePEc(경제학) 등과 같은 여러 영역의 주제형 아카이브(subject archive) 시스템들<sup>2)</sup>이 개발되고, MIT의 DSpace, 캘리포니아 대학의 eScholarship, 노팅엄 대학의 ePrints 시스템 등과 같은 기관형 저장소(institutional repository)들<sup>3)</sup>이 등장하는 배경이 되기 때문이다.

arXiv와 유사한 디지털 학술논문시스템(e-print archive)<sup>4)</sup>들은 각각의 영역에서 제각기 다른 인터페이스를 갖춘 단독시스템으로 서비스하는 수준에 머물지 않고, 공통의 이용자 인터페이스를 통한 통합검색(cross-archive search) 기능을 모색하기에 이르렀다. 결국 1999년 후반부터 징스파르그와 동료들은 OAI 프로토콜이라는 핵심기술을 이용한 새로운 통합검색 방안을 모색하게 된다.

분산시스템인 아카이브나 저장소로부터 통합서비스가 가능한 정보환경을 구축하는 일은 이제 더 이상 이상적인 모델이라 말할 수 없다. 학술자원을 포함하는 각종 정보자원에 대한 접근장벽이 최소화되고 있으며, 특히 디지털 정보처리 및 관리 기술, 인터넷을 통한 정보접근 기술을 기반으로 하여 개방접근이 가능한 여러 유형의 디지털도서관 서비스가 현실화되고 있기 때문이다.<sup>5)</sup> 또한 분산환경의 디지털도서관들은 새로운 학술정보 유통(Scholarly

---

1) <http://arxiv.org/>

2) CogPrints([cogprints.ecs.soton.ac.uk](http://cogprints.ecs.soton.ac.uk)), NCSTRL([www.ncstrl.org](http://www.ncstrl.org)), RePEc([repec.org](http://repec.org)).

3) DSpace([www.dspace.org](http://www.dspace.org)), eScholarship([escholarship.cdlib.org](http://escholarship.cdlib.org)), 노팅엄 ePrints([eprints.nottingham.ac.uk](http://eprints.nottingham.ac.uk)).

4) 디지털 학술논문시스템은 'e-print archive'를 번역한 용어이다. 이프린트(e-print)는 디지털 형식의 학술자원(scholarly resources)을 의미한다. 아카이브(archive 또는 repository)는 개방형의 디지털 아카이브(open archive)로서, 내용적으로는 개방접근이 가능한 학술자원을 수집, 처리, 보존하여 이용자에게 검색서비스를 제공하는 시스템이며, 개념적으로는 특수용도의 디지털도서관이 된다. 물론 아카이브 또는 레퍼지토리는 보다 소규모인 경우에 부르는 명칭이며, 일반적인 디지털도서관보다 관리 및 운영 체계가 미비한 시스템이라 할 수 있다. 초창기 아카이브 서비스의 시작은 일반적인 정보콘텐츠를 체계적으로 수집하고 관리하는 디지털도서관을 지향하였다기 보다는 관심분야에 국한된 학술정보자원의 관리 및 유통과 같은 풀뿌리(root-gross) 정보서비스를 위하여 관심있는 한 두명에 의해 개발되고, 관리되었다. 그래도 디지털도서관의 한 유형으로 볼 수 있기에 그냥 디지털 도서관이라고 불러도 무방하다.

5) Carl Lagoze, Herbert Van de Sompel, "The Open Archives Initiatives : Building a Low-barrier Interoperability Framework," *Joint Conference on Digital Libraries*(Roanoke, VA), 2001. ([www.openarchives.org/documents/jcdl2001-oai.pdf](http://www.openarchives.org/documents/jcdl2001-oai.pdf)) [인용 2004. 5. 2].

Communication)을 가능케 하는 통합형 디지털도서관(Networked Digital Libraries) 체제를 구축하고, 여기에 전세계의 연구자들을 연결하는 보다 혁신적인 학술정보 공동체를 형성할 수 있다.<sup>6)</sup>

본 논문은 국제적인 도서관 정보환경 구축을 위한, 학술정보 유통을 위한 통합형 디지털도서관의 핵심역할을 수행하는, 또한 디지털도서관과 관련된 여러 가지 주요한 개념 및 철학의 근간이 되는 OAI 프로토콜의 등장배경, 기술 및 구성요건과 국내에서 처음으로 시도한 적용 사례의 특징들과 비교검토하고, 향후 전개될 발전방향도 간단하게 전망하는 내용으로 구성될 것이다.

## II. OAI 프로토콜의 이해

OAI(Open Archives Initiative) 프로토콜은 메타데이터의 수확(harvesting)<sup>7)</sup>을 위한 프로토콜로서 공식적으로 사용하는 명칭은 OAI-PMH(Protocol for Metadata Harvesting)이라 부른다. OAI라는 용어가 의미하는 바를 정리하면서 그 개념을 살펴보자. OAI의 오픈(Open)은 프로토콜에 개방되어 있다는 것과 다양한 아카이브(archives)의 정보콘텐츠가 적절한 방법에 따라 접근 또는 공유가 가능하다는 의미를 나타낸다. 즉, 특정한 프로토콜에 대한 아카이브의 '개방성(openness)'을 의미한다. 아카이브(Archives)는 정보콘텐츠 특히 디지털 콘텐츠의 저장소라는 의미에서는 레파지토리(repository)로 사용되기도 한다. 넓은 의미에서 아카이브나 레파지토리는 단위 디지털도서관 그 자체이며, 하나의 유형이라 할 수 있다. 아카이브는 arXiv와 같은 주제형 아카이브일 수 있으며, 디지털 학술자원을 생산하는 대학이나 연구소 등과 같은 기관의 지적자산을 수집, 보존, 서비스하는 기관형 저장소 등 정보자원의 디지털 장서(digital collections)를 수집, 관리, 배포, 보존, 서비스 등을 제공하는 정보시스템이다. 포괄하는 정보콘텐츠의 주제범위에 따라, 또는 운영 주체나 체계적인 관리방식에 따라 서로 다르게 부르는 것뿐이다.

6) 이수상, "디지털 도서관의 개방 접근에 관한 연구," 한국도서관·정보학회지, 제34권, 제3호(2003), pp.93-110. 이 논문에서는 이러한 개방과 통합을 위한 각종 디지털도서관 서비스에 관한 논의들을 일목요연하게 정리하고 있다.

7) 수확이라는 용어가 낯설 수도 있다. 하비스팅(harvesting)에 적절하게 대응되는 우리말이라 생각한 것이다. OAI의 메타데이터 하비스팅을 의미적으로 분석하면, 아카이브의 메타데이터를 특정한 스케줄에 따라 수집하는 행위로 설명할 수 있다. 즉, 아카이브에 수집대상인 정보자원(아이템)이 생성되어 있으면, 수확기(하비스터)가 논밭의 곡식을 수확하듯이 정보자원의 메타데이터를 수집한다는 의미와 상통하므로 수확이라는 용어가 틀린 것은 아닌 듯 하다.

그러므로 개방 아카이브(Open Archives)는 개방형 디지털도서관을 의미하며, OAI 프로토콜 기능을 준수하고 표준 프로토콜을 통하여 메타데이터에의 접근을 허용한다는 의미이다. 더불어 아카이브의 콘텐츠에 대하여 비용이나 접근의 제한없이 개방접근이 가능하다는 의미도 포함한다. 물론 개방접근이라 해서 저작권법의 적용을 받지 않는다는 것은 아니다. 한편, 이니셔티브(Initiative)는 목표와 사명을 갖춘 프로젝트 또는 전문적인 활동을 주도한다는 의미라고 하겠다.

정리하면, OAI라는 용어 그 자체는 디지털도서관의 콘텐츠를 보다 효율적으로 유통하는데 유용한 각종 상호운용성 표준(interoperability standards)을 개발하고 보급하는 역할을 수행하는 국제적인 조직<sup>8)</sup>을 말한다. OAI가 주관하고 있는 대표적인 아카이브 시스템의 상호운용성 표준이 바로 OAI 프로토콜이다. 그러나 경우에 따라서는 OAI가 조직보다는 프로토콜의 의미를 나타내는 경우로 불리기도 한다.

초기 OAI 프로토콜이 개발될 당시는 학술논문의 인쇄전자료(pre-prints)나 인쇄후자료(post-prints)를 저장하는 아카이브들을 연결하는 메카니즘으로 출발하였다. 최근 들어서는 이러한 학술정보 유통영역뿐만 아니라 박물관의 문화유산<sup>9)</sup>, 강의콘텐츠<sup>10)</sup>, 학위논문<sup>11)</sup>, 언어자료<sup>12)</sup> 등으로 그 영역이 확대되고 있으며, 범용의 정보콘텐츠 유통표준으로서 자리매김하고 있다.

## 1. 등장배경

OAI 프로토콜의 등장배경을 주요 사건의 전개과정과 프로토콜 기술의 발전과정으로 나누어 살펴보자. 먼저, OAI 프로토콜(OAI 조직 포함)의 탄생에 영향을 미친 주요한 역사적 사건을 기술하면 다음과 같다.<sup>13)</sup>

- 
- 8) 사무국과 추진 및 기술위원회 조직 등으로 구성되어 있으며, DLF(Digital Library Federation)와 NSF(National Science Foundation), 그리고 CNI(Coalition for Networked Information)에서 운영채원을 지원하고 있다. 활동조직에는 국제적인 관계자들이 참여하고 있으며, 그 중에서 주로 유럽과 북미의 기관 및 단체, 전산전문가와 사서들이 참여하고 있다. 공식 홈페이지 : <http://www.openarchives.org>.
  - 9) UIUC(The Univ. of Illinois at Urbana-Champaign) OAI Metadata Harvesting Project, <http://oai.grainger.uiuc.edu/>.
  - 10) 미국의 NSDL([www.nsdl.org](http://www.nsdl.org)), 캐나다의 CAREO([www.careo.org](http://www.careo.org)) 등 강의용 콘텐츠를 저장소(Learning Objects Repository)에 구축하고, 제공하는 서비스.
  - 11) NDLTD(Networked Digital Library of Theses and Dissertations), <http://www.ndltd.org/>.
  - 12) OLAC(Open Language Archives Community), [www.language-archives.org](http://www.language-archives.org).
  - 13) 시간별 사건은 OAI 홈페이지([www.openarchives.org](http://www.openarchives.org))에 게시된 자료와 쏘펠(Herbert Van de Sompel)과 라고즈(Carl Lagoze)의 논문인 "The Santa Fe Convention of the Open Archives Initiative," *D-lib Magazine*, Vol.6, No.2(2000), 그리고 슈버(Peter Suber)가 정리한 "Open-Access Timeline"([www.earlham.edu/~peters/fos/timeline.html](http://www.earlham.edu/~peters/fos/timeline.html)) 등을 기반으로 정리하였다.

- 1999. 10. (Santa Fe), OAI 1차회의 : 복수의 아카이브들(arXiv, CogPrints, RePEc, NCSTRL, 각종 기관형 저장소 등)을 대상으로 하는 UPS(Universal Preprint Service) 즉, 통합서비스 방법을 모색하기 위해 모인 회의. 메타데이터 수확 프로토콜의 전신인 산타페규약(Santa Fe Convention)을 발표. 기존의 분산검색 방식(federated search)을 지양하고, 메타데이터 수확을 통한 새로운 방식(harvesting search)을 제안.
- 2000. 6. (San Antonio), OAI 2차회의 : 1차회의에서 합의한 사항을 승인하고, 새로운 내용을 추가하는 작업을 수행.
- 2000. 9. (Cornell대학), OAI 기술위원회 1차회의 : 메타데이터 수확에 관한 기술적인 합의와 각종 조직 및 정책적 쟁점사항들을 집중적으로 논의한 회의. 그 결과 당시 코넬(Cornell)대학의 교수로 재직 중인 쯘펠(Herbert Van de Sompel)을 중심으로 OAI 사무국을 개설.
- 2000. 9. (Lisbon), OAI 1차 유럽회의 : 기술위원회 회의 직후에 개최된 유럽회의로서, OAI 조직에 유럽의 새로운 참가자들이 가입.
- 2000. 11. (Geneva), OAI 기술위원회 2차회의 : 유럽의 과학자들과 사서들이 새로운 학술정보유통 시스템에 OAI 기술표준을 수용하기 위한 방안을 논의한 워크샵.
- 2001. 1. (Washington DC) 공개회의(미국) : 새로운 OAI 프로토콜 사양(버전 1.0)을 발표. 많은 알파 테스트(alpha-test)와 기술회의를 거친 후 쯘펠과 라고즈 주도하에 개발.
- 2001. 2. (Berlin) 공개회의(유럽) : 유럽에서 수행된 새로운 OAI 사양 발표.
- 2001. 7. (Cornell대학) : 1.0 버전의 오류수정과 새로운 기능을 추가하여 쯘펠과 라고즈의 주도하에 1.1 버전을 개발.
- 2002. 6. (Cornell대학) : 쯘펠과 라고즈가 주도하였으며, OAI 기술위원회 위원인 넬슨(Michael Nelson), 와너(Simeon Warner)가 함께 참여하여 개발한 보다 안전한 2.0 버전을 발표.

기술적 관점에서 보면, OAI 프로토콜은 수많은 아카이브들을 대상으로 통합서비스를 제공하는 방법을 모색하기 위한 1999년 10월 산타페 회의에서 논의된 결과물이다. 이처럼 아카이브들의 통합검색과 정보공유 등과 같은 상호운용성을 보장하기 위한 방안을 모색하기 위하여, 당시 회의의 참가자들은 미리 준비한 UPS 프로토타입을 집중적으로 검토하였다. UPS 프로토타입은 전산학 분야 아카이브인 NCSTRL에서 사용한 통합검색 기법인 'Dienst' 프로토콜을 토대로 쯘펠, 크리첼(Thomas Krichel), 넬슨등이 수정하여 제시한 것이었다. UPS 프로토타입은 서로 다른 복수의 아카이브들로부터 메타데이터를 수집하고, 이것을 공통의 포맷으로 변환하여 검색과 연계 서비스를 제공하도록 되어 있다.

쯘펠과 그의 동료들은 UPS 검토회의에서 다음과 같은 5가지 사항을 제안하였다<sup>14)</sup>: 1)수확

서비스와 메타데이터 제공자(SP/DP)의 구분, 2)분산검색의 대안으로서 수확된 메타데이터를 대상으로 하는 검색방안, 3)SP/DP 모델의 제반특성을 설명하는 프레임워크, 4)아카이브들에 대한 고유한 식별체계, 5)공통의 메타데이터 포맷등이 바로 그것이다. 이를 토대로 회의의 참석자들은 UPS 기반의 메타데이터 수확 모델을 채택하기로 합의하였으며, 그래서 이것을 ‘산타페규약’이라 불렀다. 얼마 지나지 않아 당시 참석자들을 중심으로 OAI 조직을 결성하고, 프로토콜 명칭도 ‘OAI-PMH’라 개정하게 된다.

이후 쌰펠과 라고즈를 중심으로 하여 얼마간의 알파 테스트를 거친 이후 2001년 1월, OAI 프로토콜은 1.0 버전이 발표된다. 1.0 버전의 주요 특징은 6가지 명령어 세트와 기본 DC (unqualified Dublin Core)를 표준포맷으로 채택한 점이다. 1년 이상의 관찰기간을 거친 이후 2001년 6월에 이르러 XML 표준을 채택한 1.1 버전이 발표되고, 2002년 6월에는 버전 2.0이 발표된다. 2.0 버전에는 1.1 버전과는 달리 메타데이터 교환을 위한 XML 스키마를 채택하고, 애매한 용어나 개념들을 정리하며, 프로토콜과 HTTP 표준간 역할을 명확히 구분하는 등과 같은 몇 가지 개선사항이 추가되었다. 버전 2.0은 더 이상 실험용의 버전이 아니라 실제의 응용 영역에서 안정적으로 활용할 수 있는 버전으로 평가받고 있으며, 디지털도서관의 상호 운용성을 위한 명실상부한 기술적 기반으로서 역할을 인정받게 되었다.

## 2. 기본개념

### 1) SP/DP 모델

OAI 프로토콜의 운용은 SP/DP라는 아주 단순한 모델로서 설명된다. SP(Service Provider)는 DP들로부터 수확한 메타데이터를 기반으로 검색, 브라우징, 연계 및 원문제공 등과 같은 부가가치 서비스를 제공하는 기능을 수행한다. DP(Data Provider)는 자체적으로 각종 디지털장서를 수집, 보유하고 있는 시스템이며, SP의 수확요청에 대응하여 적합한 메타데이터를 제공하는 역할을 수행한다. 이러한 SP/DP 모델은 각각의 역할 또는 기능에 따라 구분되는 개념이지, 그것이 바로 시스템인 것은 아니다.

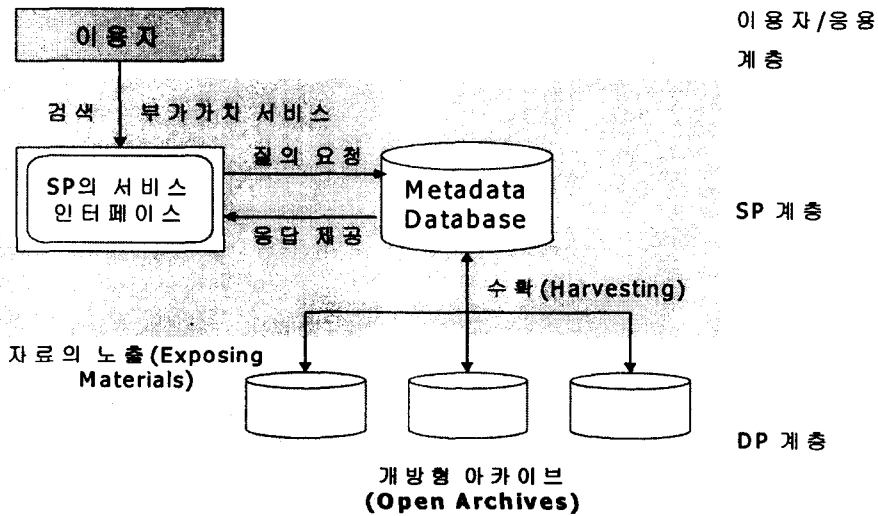
시스템적으로 구분한다면, SP의 역할과 기능을 수행하는 시스템은 통합형 디지털도서관 체제가 된다. DP의 역할과 기능을 수행하는 시스템은 SP의 수확요청에 대응하는 기능인 DP 인터페이스를 갖추고 있는 기관 또는 주제별로 운영되는 단위 디지털도서관이다.<sup>14)</sup> 그래서

14) Herbert Van de Sompel, et al., "The UPS Prototype : An Experimental End-User Service across E-Print Archives," *D-Lib Magazine*, Vol.6, No.2(2000).

(<http://www.dlib.org/dlib/february00/vandesompel-ups/02vandesompel-ups.html>). [인용 2004. 5. 5].

15) 응용 영역의 입장에서 보면, DP는 도서관, 박물관 등에서 운영하는 아카이브 또는 레퍼지토리로써

통합형 디지털도서관 체제라 할 경우, 이는 DP 인터페이스를 갖춘 단위 디지털도서관들을 네트워크로 연결하여, 메타데이터들을 수집하고, 이용자에게 각종 부가가치 서비스를 제공하는 시스템이 된다.<sup>16)</sup> 물론 단위 디지털도서관 자체가 SP 기능을 구비하여, 통합형 디지털도서관으로 구성할 수 있다. 통합형 디지털도서관 체제를 구성하는 SP/DP 모델을 도식화하면 <그림 1>과 같다.



<그림 1> 통합형 디지털도서관의 SP/DP 모델

## 2) OAI 프로토콜

<그림 2>와 같이 OAI 프로토콜은 6가지 명령어에 의하여 SP와 DP간 커뮤니케이션을 수행한다. 6가지 명령어를 구분하면, 'Identifier', 'ListSets', 'ListMetadataFormats'의 3가지 명령어는 아카이브에 대한 정보를 확인하는 것이며, 실질적으로 메타데이터를 수확하는 명령어는 'ListIdentifiers', 'GetRecord', 'ListRecords'의 3가지이다.

개방형 디지털도서관이다.

16) 대표적인 통합형 디지털도서관 체제의 사례는 NDLTD([www.ndltd.org](http://www.ndltd.org)), OAIster([oaister.umdl.umich.edu](http://oaister.umdl.umich.edu)), Arc([arc.cs.odu.edu](http://arc.cs.odu.edu)), KERIS dcollection([www.dcollection.net](http://www.dcollection.net)) 등이다.





|   |                        |
|---|------------------------|
| <pre>&lt;?xml version="1.0" encoding="UTF-8"&gt; &lt;record&gt; &lt;header&gt; &lt;identifier&gt;oai:arXiv:quant-ph/9901001&lt;/identifier&gt; &lt;timestamp&gt;1999-01-01&lt;/timestamp&gt; &lt;/header&gt;</pre>  | <p>Header<br/>컨테이너</p> |
| <pre>&lt;metadata&gt; &lt;dc xmlns="http://purl.org/dc/elements/1.1/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://purl.org/dc/elements/1.1/ http://www.openarchives.org/OAI/1.1/dc.xsd"&gt; &lt;title&gt;Quantum slow motion&lt;/title&gt; &lt;creator&gt;Hug, M.&lt;/creator&gt; &lt;creator&gt;Milburn, G. J.&lt;/creator&gt; &lt;description&gt;We simulate the center of mass motion of cold atoms in a standing, amplitude modulated, laser field as an example of a system that has a classical mixed phase-space.&lt;/description&gt; &lt;date&gt;1999-01-01&lt;/date&gt; &lt;type&gt;e-print&lt;/type&gt; &lt;identifier&gt;http://arXiv.org/abs/quant-ph/9901001&lt;/identifier&gt; &lt;/dc&gt; &lt;/metadata&gt;</pre> | <p>Body<br/>컨테이너</p>   |
| <pre>&lt;about&gt; &lt;ea xmlns="http://www.arXiv.org/eprints-about" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.arXiv.org/eprints-about http://www.arXiv.org/eprints-about.xsd"&gt; &lt;archive&gt;The Los Alamos arXiv&lt;/archive&gt; &lt;usage&gt;Metadata may be used without restrictions as long as the OAI identifier remains attached to it.&lt;/usage&gt; &lt;/ea&gt; &lt;/about&gt; &lt;/record&gt;</pre>   | <p>About<br/>컨테이너</p>  |

<그림 3> 응답 레코드의 사례

### 3) 전송용 메타데이터

OAI 프로토콜에서 메타데이터의 문제는 매우 중요한 사안이므로 산타페규약에서부터 논의되었다. 당시에는 RePEc에서 사용된 ReDIF 포맷을 공통 메타데이터 포맷으로 선정하였다. 이를 OAMS(Open Archives Metadata Set)라 하였으며, 많은 아카이브는 OAMS로 변환하는 매핑절차에 따라 자신들의 메타데이터 요소들을 OAMS로 변환하도록 하였다. OAMS는 단순한 구조의 시맨틱스로 정의되었으며, 메타데이터를 표현하고 전송하는 공통의 선택으로 XML을 사용하는 등의 특징을 가지고 있다.

이러한 OAMS의 기본요건은 다음의 2가지로 정리할 수 있다. 개별 아카이브들은 자신의 응용영역에 적합한 풍부한 요소들로 구성된 메타데이터를 사용하고, OAI 프로토콜에 따라 메타데이터를 전송하기 위한 최소한의 부분집합을 제공할 수 있어야 한다. 아카이브들은 서로 다른 메타데이터 표준을 사용하더라도, 전송 및 공유를 위한 공통 포맷을 지원하여야 한

다는 의미이다.

이와 같은 공통포맷의 필요성은 당연한 것이었다. 만일 아카이브들마다 다양한 메타데이터에 대한 변환작업이 요구된다면, 메타데이터를 수집하여 사용하는 SP의 입장에서 보면 대단히 부담스러운 일이 된다. 결국 DP와 SP간 공통의 메타데이터 포맷을 사용하는 것은 공통 메타데이터 포맷으로의 변환만 지원하면 되므로, 이들간의 커뮤니케이션이 단순화되고 SP의 구현도 편리해지는 잇점이 있다. OAI에서는 1.0 프로토콜을 발표하면서, 공통의 메타데이터 표준으로 기본 DC를 선택하였다. 기본 DC 포맷은 신택스와 시맨틱스가 단순하고, 모든 영역을 포괄하는 사실상의 표준으로 인정받고 있는 터라, 적절한 선택이었다고 볼 수 있다.<sup>17)</sup>

OAI 프로토콜에서 교환 및 전송용 메타데이터에 요구되는 일반원칙을 정리하면 다음과 같다. 첫째, DP시스템에서는 해당되는 응용 영역에 적합한 메타데이터 표준을 사용하면 된다. 반드시 기본 DC 포맷을 사용할 필요가 없다는 것이다. 메타데이터 표준에는 제한이 없으며, 응용영역에 따라 협의만 되면 무방하다는 것이다. 둘째, 메타데이터의 교환 및 전송은 XML 데이터로 구조화되어야 한다. 전송 및 교환을 위한 레코드의 신택스는 XML을 사용한다는 것이다. 셋째, XML 레코드에 포함된 각 메타데이터 포맷에 대한 정보(스키마 정보)는 반드시 네임스페이스로 명시되어야 한다.

#### 4) OAI 식별체계

식별체계에 대한 요구 자체는 매우 단순하지만, 구현에는 여러 가지 고려사항들을 검토해야 한다. 요구사항의 핵심은 DP의 메타데이터나 원문콘텐츠에 쉽게 접근할 수 있도록 표준화된 방식의 식별체계(Identifier format)를 사용하자는 내용이다. SP의 입장에서 보면, 식별체계는 수집된 레코드의 DP를 구분하는 식별자(DP 식별자)와 해당 DP에서의 레코드 아이টে에 대한 식별자(레코드 식별자)의 2가지 영역으로 구분되어진다. OAI에서는 이를 OAI 식별자(oai-identifier)라고 부른다. 그렇게 되어야지 SP는 DP 아카이브의 아이টে에 대한 메타데이터를 추출하고, SP의 통합서비스 이용자는 원문 아이টে에 접근할 수 있게 된다. OAI 식별자의 신택스는 다음과 같이 URN(Uniform Resource Name)과 유사한 형식을 갖추고 있다.

$$\text{oai-identifier} = \text{scheme} \text{ ":" } \text{namespace-identifier} \text{ ":" } \text{local-identifier}$$

여기서 스킴(scheme)은 OAI 프로토콜을 준수한다는 의미로서, 'oai'라는 문자로 표시된다. 네임스페이스 식별자(namespace-identifier)는 OAI 네임스페이스에서 구분이 가능한 아카이

17) 당시 OAI는 기본 DC 포맷, RFC 1807 메타데이터 포맷, XML 형식의 MARC 21 포맷등 3가지 XML 스키마를 발표하게 된다. OAI 프로토콜을 사용하기 위해서는 이 3가지 포맷을 모두 지원해야 한다는 것은 아니며, 최소한 기본 DC 포맷만은 지원해야 한다.

브(DP) 이름이다. 로컬식별자(local-identifier)는 해당 아카이브 내에서 식별이 가능한 레코드의 식별자를 말한다. 예를 들어, 'oai:arXiv:hep-th01'인 경우 'oai'는 OAI 사이트라는 문자표시이며, 'arXiv'는 아카이브에 대한 고유명, 그리고 'hep-th01'는 해당 arXiv에서의 레코드 제어번호와 같은 식별자이다. 이처럼 아카이브의 이름이나 레코드를 식별하는 표준체계에 대해서는 프로토콜 자체에서 명시되고 있지 않지만, 그 형식만은 제시하고 있다. SP에서는 네임스페이스 식별자를 등록, 관리하는 기능을 갖추고 있어야 한다. OAI 식별자는 1.0 버전 프로토콜에서부터 소개되었으며, 이후 크게 변하지 않고 있다.<sup>18)</sup>

### 3. SP/DP의 구성요건

OAI 사무국은 OAI 프로토콜의 기본적인 기능 및 구현 사양에 대한 지침을 권장하고 있다. 여기서는 이러한 사양지침에 대한 상세한 설명보다는 OAI 프로토콜에 의해 구현되는 SP와 DP의 구성요건 즉, 기본적인 기술요건과 핵심적인 기능 구성에 대한 지침부분을 검토하기로 한다. 이 지침문서는 오픈아카이브포럼(Open Archives Forum)에서 제시한 튜토리얼을 참조하였다.<sup>19)</sup>

- SP의 기본요건

| 기본요건       | 내 용  |
|------------|--|
| 인터넷 서버     | 인터넷에 연결되어 있는 서버이어야 한다.                       |
| 데이터베이스 시스템 | 관계형 또는 XML 기반의 데이터베이스 시스템을 운영하여야 한다.         |
| 프로그래밍 환경   | 웹서버에 HTTP 요청이 가능하여야 하며, XML 파서를 포함하고 있어야 한다. |

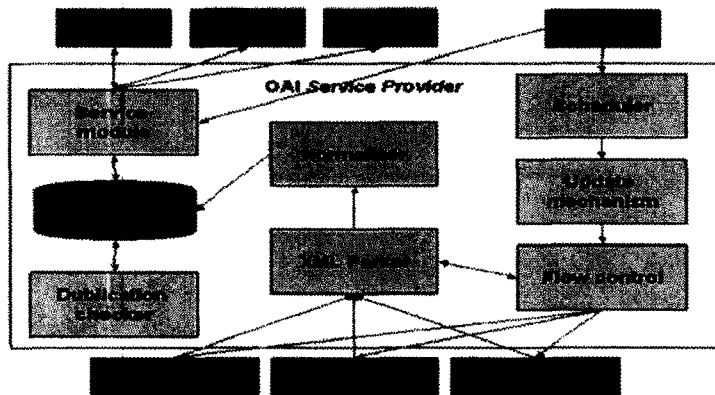
18) OAI 문서, "Specification and XML Schema for the OAI Identifier Format," [www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm](http://www.openarchives.org/OAI/2.0/guidelines-oai-identifier.htm). [인용 2004. 5. 5].

19) 오픈아카이브포럼 튜토리얼 자료, "OAI for Beginners - the Open Archives Forum Online Tutorial," <http://www.oaforum.org/tutorial/index.php>. [인용 2004. 5. 5].

• SP의 핵심기능

| 핵심기능                             | 내 용  |
|----------------------------------|--|
| 아카이브 관리기<br>(Archive Management) | 수집대상 아카이브를 선택하는 기능. 아카이브들은 수동 또는 자동으로 삭제가능                               |
| 요청기<br>(Request Component)       | 정해진 HTTP 요청을 생성하여, OAI 아카이브(DP)로 전송하는 기능                                 |
| 스케줄러(Scheduler)                  | 수동 또는 자동으로 관련된 아카이브를 규칙적으로 작동시키는 기능                                      |
| 흐름제어(Flow Control)               | 대량의 결과를 일정한 크기로 나누어 흐름제어 토큰(resumption token)을 사용하여 전송하는 기능              |
| 갱신기능<br>(Update Mechanism)       | 과거에 수확한 메타데이터를 새롭게 수확한 메타데이터로 갱신하는 기능                                    |
| XML 파서<br>(XML Parser)           | XML 스키마를 사용하여 저장소로부터 수집한 응답을 분석, XML로 인코딩되어 있는 메타데이터를 내부 데이터 구조로 변환하는 기능 |
| 정규화 처리기<br>(Normaliser)          | 날짜, 저자, 언어코드 등 서로 다른 메타데이터 포맷으로 표시된 데이터를 동일한 구조로 변환하는 기능                 |
| 데이터베이스(Database)                 | 정규화 처리기의 결과를 데이터베이스에 저장하는 기능. 경우에 따라서는 XML DB로 저장 가능                     |
| 중복검사기<br>(Duplication Checker)   | 서로 다른 DP에서 수집한 동일 레코드를 병합하는 기능   |
| 서비스 모듈<br>(Service Module)       | 수집된 레코드를 대상으로 이용자에게 제공되는 실제적인 서비스 기능                                     |

이러한 기본요건과 핵심기능을 토대로 OAI에서는 <그림 4>와 같은 SP의 구성도를 제안하고 있다.



<그림 4> OAI 권장 SP 구성도

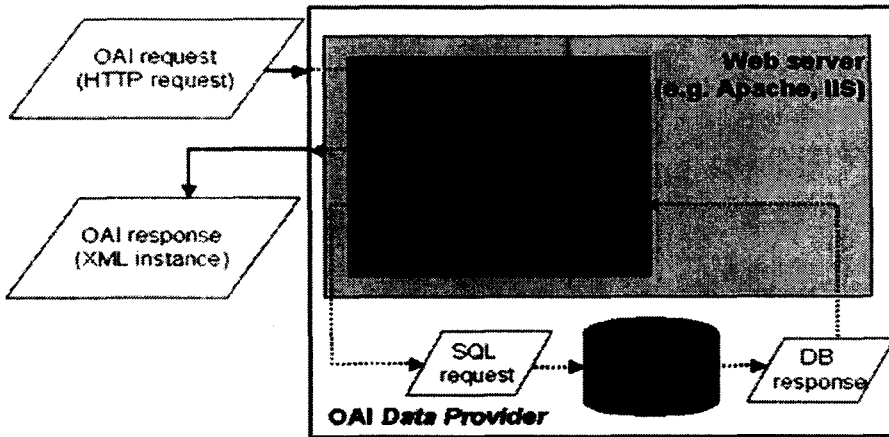
• DP의 기본요건

| 기본요건                             | 내 용  |
|----------------------------------|--|
| 메타데이터                            | 자원 즉 아이템(items)에 관한 메타데이터는 데이터베이스(예를 들어, SQL DB와 같은)에 저장되어야 하며, 각 아이템은 식별자(unique identifier)를 가져야 한다. |
| 웹서버                              | 인터넷(예를 들어, Apache, IIS 등)으로 접속이 가능한 서버이어야 한다.  |
| 프로그래밍 인터페이스(API)                 | Perl, PHP, Java-Servlet 등으로 구현되며, 웹서버에 설치되고, DB(또는 파일시스템) 접근이 가능하여야 한다.                                |
| 아카이브 식별자                         | URL기반으로 구성되어야 한다.  |
| 아이템 식별자                          | 아이템 단위로 고유한 식별자가 부여되어야 한다.   |
| 메타데이터 포맷                         | 한 종류 이상의 포맷을 지원하여야 한다. 최소한 기본 DC 포맷을 지원하여야 한다.   |
| 시간표시자(datestamps)                | 메타데이터의 시간표시자는 최근 수정일을 기준으로 생성되어야 한다.   |
| 논리적 세트 구조(logical set hierarchy) | 필요시 사용가능 하여야 한다.   |
| 흐름제어                             | 필요시 흐름제어 토큰에 의해 구현되어야 한다.  |

• DP의 핵심기능

| 핵심기능  | 내 용  |
|---|--|
| 요청분석기 (Argument Parser)                                     | OAI 요청 명령어를 분석하는 기능                            |
| 오류생성기 (Error Generator)                                     | XML형식의 오류 메시지를 생성하는 기능                         |
| DB탐색 및 메타데이터 추출기(Database Query /Local Metadata Extraction) | 요청한 포맷으로 DB에서 메타데이터를 검색하는 기능                   |
| 응답 XML 생성기(XML Generator/Response Creation)                 | 메타데이터 정보를 담은 응답 XML을 생성하는 기능                   |
| 흐름제어  | 대규모 데이터인 경우, 흐름제어 토큰을 사용하여 전송 데이터의 흐름을 제어하는 기능 |

OAI에서 권장하는 DP의 구성도는 <그림 5>와 같다.



<그림 5> OAI 권장 DP 구성도

### III. OAI 프로토콜 적용 사례분석 : KERIS의 생성유통체계

여기서는 OAI 프로토콜을 적용한 사례분석을 통해 그것에 구현되어진 응용모델, SP와 DP의 요건 및 기능모듈의 특징 등을 살펴보기로 한다. 적용사례는 국내에서는 처음으로 시도된 KERIS(한국교육학술정보원)의 지식자원의 '생성유통체계' 구축 프로젝트이다. 2003년부터 시작된 본 사업은 대학기관(이화여대, 부산대, 충남대, 성균관대)과 교육유관기관(KERIS, 교육인적자원부, 한국교육개발원, 한국직업능력개발원)이 참여한 시범사업이었으며, 국가적 차원의 학술연구정보의 생성 및 유통 체계를 시범적으로 구축하는 것을 목표로 삼고 있다. 이후 2004년부터는 기능개선과 참여기관의 확대보급을 위한 사업을 진행하고 있다.

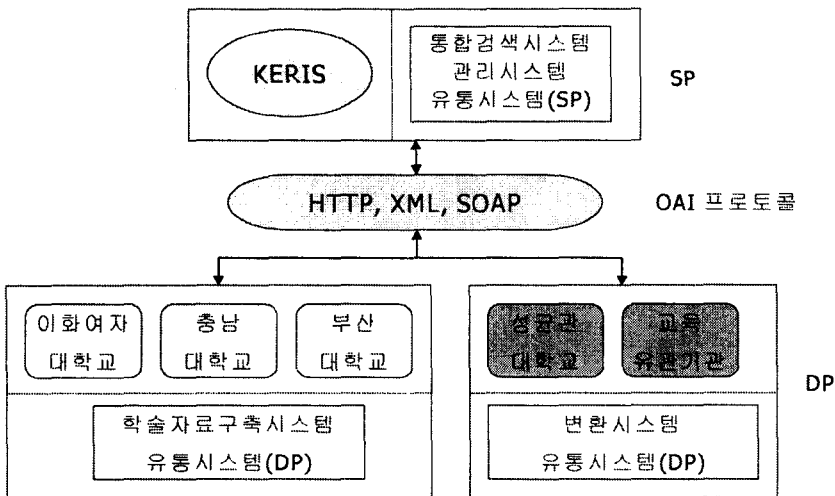
사업의 개요 및 구체적인 목적에 대한 상세한 정보는 사업홈페이지<sup>20)</sup>의 여러 문서에 잘 나타나 있으므로 여기서는 구체적인 언급을 생략하고, 본 논고의 핵심안건인 OAI 프로토콜이 어떻게 구현되었는지, 앞서 언급한 SP/DP의 기본요건 및 핵심기능이 어떻게 구현되었으며, OAI 권장사항과 어떠한 차이가 나는지를 항목별로 나누어 살펴보기로 하겠다.

20) <http://www.dcollection.net/project>.

## 1. 시스템의 구성 모델

‘생성유통체계’(일명 KERIS dCollection 시스템)로 명명된 KERIS 사례의 구성 모델은 <그림 6>과 같이 OAI에서 제시하는 전형적인 SP/DP 모델을 준수하고 있다. SP는 KERIS가 되고, DP는 대학 및 교육유관기관들로 구성된 참여기관들이다. DP 참여기관에서는 각종 학술자원(학위논문, 학술지논문, 연구보고서 등)을 자체적으로 생성(온라인 제출 또는 self-archiving)하여 서비스하는, SP는 생성된 자원의 메타데이터를 OAI 프로토콜을 이용한 유통 기능으로 수집(또는 수확)하여 통합메타DB를 구성하며, 일반이용자에게 통합검색 서비스를 제공하는 모델이다. 이른바 학술정보자원의 생성, 유통, 서비스를 위한 통합 플랫폼인 것이다.

OAI 프로토콜은 HTTP, XML, SOAP 표준을 사용하여 구현하고 있다. HTTP는 SP에서 DP로 명령어를 요청할 때 사용하는 표준이며, XML은 DP에서 SP로 응답하는 레코드의 포맷이다. SOAP(Simple Object Access Protocol)은 SP와 DP간의 명령어나 응답과 같은 데이터를 전송할 때 사용하는 표준이다.



<그림 6> KERIS 생성유통 체계 모델

이 모델에서 DP를 구성하고 있는 단위 모듈은 학술자료구축시스템, 변환시스템, 유통시스템(DP)이고, SP를 구성하고 있는 단위 모듈은 유통시스템(SP), 관리시스템, 통합검색시스템이다. 여기서 DP는 2가지 유형으로 구분된다. KERIS에서 개발하여 보급한 ‘학술자료구축시스템’을 사용하여 학술자료를 구축하고 유통하는 유형과 기관에서 기존에 운영하고 있는 시스템을 통해 학술자료를 구축하고 유통하는 경우이다. 이들 시스템 각각에 대하여 간략히 설

명하면 다음과 같다.

- 학술자료구축시스템 : 생성유통 체계에 참여하는 대학에 설치되는 시스템으로서, 디컬렉션(dCollection)이라 부르기도 한다. 이 시스템을 사용하는 경우 참여대학에서는 온라인대출 기능을 통해 학술자료를 수집하고 데이터베이스를 구축하며, 유통시스템(DP) 인터페이스를 통해 SP의 메타데이터 수집요청에 대응한다.
- 변환시스템 : 참여기관에서 학술자료시스템과 유사한 기능을 갖춘 시스템을 기 보유한 경우, 기존 포맷의 메타데이터를 유통시스템(DP)가 처리할 수 있는 형식으로 변환하는 프로그램이다.
- 유통시스템(DP) : 참여기관 시스템의 전방(front-end)에 설치되는 인터페이스이며, OAI 프로토콜에 따라 유통시스템(SP)의 수집요청에 응답하여 메타데이터를 제공하는 프로그램이다.
- 유통시스템(SP) : KERIS에 설치되는 인터페이스이며, 각 참여기관의 유통시스템(DP)에 메타데이터를 요청하는 명령어를 수행하고 통합메타DB를 구축하는 프로그램이다.
- 관리시스템 : 유통시스템(SP)로부터 수집된 통합메타데이터를 수정, 삭제하며, 관리하고, 각종 통계데이터를 산출하는 프로그램이다.
- 통합검색시스템 : 각 참여기관에서 수집한 통합메타DB를 대상으로 통합검색 기능 및 부가가치 서비스를 제공하는 프로그램이다.

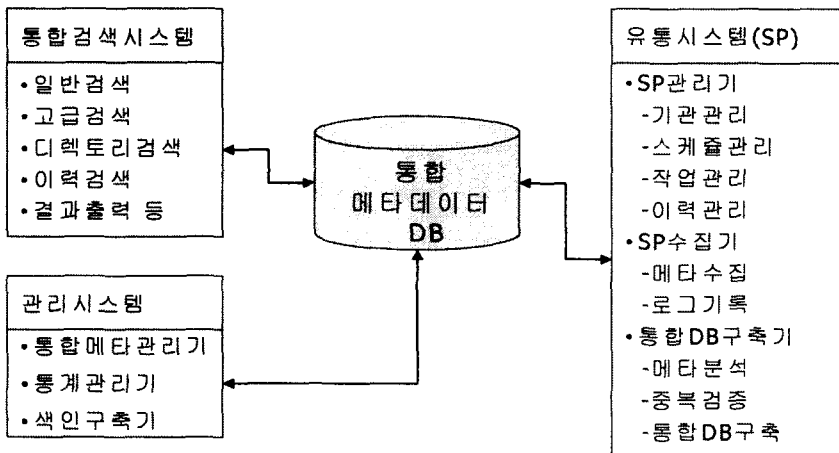
## 2. SP의 구현 내용

### (1) SP의 핵심기능 구성

KERIS 생성유통 체계에서의 SP의 기능 구성도는 <그림 7>과 같으며, 그것의 주요 특징을 정리하면 다음과 같다. SP는 프로토콜을 처리하는 유통시스템(SP), 통합메타DB 관리 기능, 통합검색 기능으로 나누어진다. 여기서 유통시스템(SP)과 관리시스템은 통합메타데이터를 수집하여 DB화하는 시스템이며, 통합검색시스템은 통합메타DB를 대상으로 각종 검색서비스를 제공한다. 통합검색시스템은 일반적인 검색기능과 유사하므로 세부적인 설명은 생략하고, OAI 프로토콜에 따라 메타데이터를 수집하고 관리하는 역할을 수행하는 유통시스템(SP)과 관리시스템의 주요 기능들을 살펴보기로 한다.



- SP관리기 : 기관관리, 스케줄관리, 작업관리, 이력관리의 단위 모듈로 구성된 관리기능이다.
  - 기관관리 : 참여기관의 리스트 및 상세정보 조회, 등록, 삭제 등의 관리기능을 수행한다. 관리정보는 기관명, 기관등록명, 기관코드, 기관URL, 운영상태 등으로 구성된다.
  - 스케줄관리 : 스케줄 리스트 및 상세정보 조회, 운영 및 중단 선택, 삭제 등의 관리기능을 수행한다. 스케줄에 포함되는 정보는 기관명, 기관등록명, 기관코드, 기관URL, 수집시작일시, 수집범위(from), 수집범위(until), 수집유형, 갱신주기, 처리상태, 운영상태, 스케줄 ID 등으로 구성된다.
  - 작업관리 : 작업관리 리스트 및 상세정보 조회, 선택 등의 관리기능을 수행한다.
  - 이력관리 : 참여기관의 작업이력 리스트 및 상세정보 조회, 삭제 등의 기능을 수행한다.
- SP수집기 : 메타수집과 로그기록 기능으로 구성된다. 메타수집 기능은 요청한 메타데이터를 유통시스템(DP)로부터 흐름제어 토큰값이 없을 때(null)까지 수집한다. 수집작업을 완료한 후 성공과 실패에 따라 처리한다.
- 통합DB구축기 : 메타분석, 중복검증, 통합DB구축 기능으로 구성되며, 통합메타DB를 구축하는 기능을 수행한다. 메타분석 기능은 수집된 메타데이터를 분석하는 기능을 수행하며, 중복검증 기능은 동일한 식별자(ID)를 가지는 아이템인지를 확인하여 처리한다. 통합DB구축 기능은 메타데이터를 아이템 DB에 저장한다.
- 통합메타관리기 : 구축된 통합메타DB에 대한 관리기능으로 통합메타DB의 검색, 리스트출력, 상세보기, 수정 및 추가 등의 작업을 수행한다.
- 통계관리기 : 통합메타DB의 구축통계, 이용통계 등의 작업을 수행한다.
- 색인구축기 : 통합검색에 필요한 각종 색인을 생성하고 관리하는 기능을 수행한다.



<그림 7> KERIS 생성유통 체계의 SP 기능구성도

## (2) SP의 기본요건 적용

KERIS 생성유통 체계에서의 SP는 OAI에서 권장하는 SP의 기본요건을 충실히 준수하고 있으며, 각각의 구체적인 적용 방식을 요약하여 정리하면 다음과 같다.

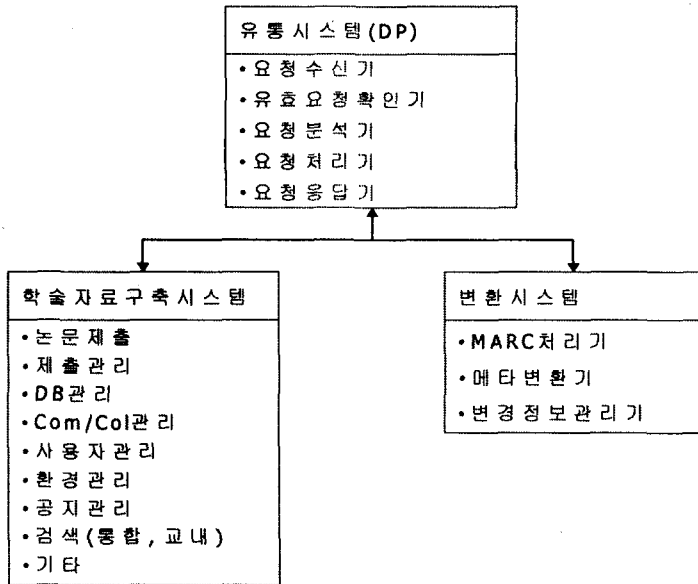
- 인터넷 서버 : KERIS의 시스템 운영환경을 고려하여 인터넷 서버는 WEBTOB 환경에서 운영하고 있다.
- 데이터베이스 시스템 : DBMS는 오라클(Oracle 9i) 환경에서 DB서버를 구성하고 있다.
- 프로그래밍 환경 : 각종 관리기능은 JEUS기반의 EJB나 JSP 언어를 사용하였고, 유통 시스템(SP)는 Java와 Applet 언어를 사용하고 있다.

## 3. DP의 구현 내용

### (1) DP의 핵심기능 구성

KERIS 생성유통 체계에서의 DP의 기능구성도는 <그림 8>과 같으며, DP의 프로토콜 처리기능인 유통시스템(DP)와 학술자료구축시스템, 그리고 변환시스템으로 구성된다. 학술자료 구축시스템과 변환시스템은 앞서 그 내용을 소개하였기에, 여기서는 OAI에서 권장하는 DP의 기능들을 처리하고 있는 유통시스템(DP)를 구성하는 주요 모듈을 중심으로 살펴보기로 한다.

- 요청수신기 : SP에서 전송한 요청SOAP을 분해하여 요청내용을 파싱한다.
- 유효요청확인기 : 유효한 SP로부터 요청된 것인지를 확인한다.
- 요청분석기 : 요청내용을 분석(명령어 유형과 파라미터값)하고 정상적인 요청이면 요청 처리기를 호출한다.
- 요청처리기 : 6가지 명령어 요청에 해당되는 작업을 DB나 설정파일 등을 이용하여 수행하고 XML 형식으로 적합한 응답을 생성한다.
- 요청응답기 : 생성된 응답을 SOAP으로 바인딩하여 SP로 전송한다.



<그림 8> KERIS 생성유통 체계의 DP 기능구성도

## (2) DP의 기본요건 적용

AI에서 권장하는 DP의 기본요건도 대체로 충실히 준수하고 있으며, 각각의 구체적인 적용 방식을 정리하면 다음과 같다.

- 메타데이터 : 아이템에 관한 메타데이터는 현재 DBMS(Oracle 8i) 이상에서는 동작되도록 구성하고 있다.
- 웹서버 : 아파치(Apache) 기반의 웹서버에서 운영된다.
- 프로그래밍 인터페이스 : 유통시스템(DP)과 변환시스템은 주로 Java 언어를 사용하고, 학술자료구축시스템은 톰캣(Tomcat) 기반의 JSP 언어를 사용하고 있다.
- 아카이브 식별자 : URL 기반의 식별자(기본URL)를 사용하고 있다.
- 아이템 식별자 : 12자리의 순차적 제어번호(로컬 ID)를 사용하고 있다.
- 메타데이터 포맷 : 응용영역에 필요하다고 판단되는 SOMS<sup>21)</sup>라고 하는 메타데이터 표준을 자체적으로 개발하여 사용하였으며, 기본DC 포맷은 지원하지 않고 있다.

21) SOMS(Scholarly Object Metadata Set)는 KERIS의 생성유통체계 환경에서 다양한 학술자료들의 서지적 특성을 기술할 수 있는 범용의 표준 메타데이터로서, MIT의 DSpace에서 정의한 메타데이터, DCMI의 DC 메타데이터 요소들과 용어 등을 참조하여 개발한 것이다. 상세한 정보는 KERIS 사업홈페이지의 관련 페이지([www.dcollection.net/project/s2\\_metadata\\_standard.htm](http://www.dcollection.net/project/s2_metadata_standard.htm))를 참조하면 된다.

- 시간표시자 : 최근에 수확된 시간을 표시하는 시간표시자는 사용하고 있으며, '초'단위로 표시하는 방법을 사용하고 있다.
- 논리적 세트 구조 : 현재로서는 논리적 세트 구조를 적용하지 않고 있다.
- 흐름제어 : 적용하고 있으며, SP의 흐름제어 토큰값에 따라 처리하고 있다.

#### 4. 차이점 분석

KERIS 생성유통 체계의 SP/DP의 구현 내용을 OAI에서 권장하는 핵심기능과 대비하여 정리하고, 그 차이를 살펴보기로 한다. 먼저, SP/DP의 핵심기능의 구현 내용에 대한 대비표는 <표 1>과 <표 2>와 같다.

<표 1> SP의 핵심기능의 구현내용 비교

| OAI권장<br>핵심기능 | KERIS 핵심기능 및 내용   |  |
|---------------|-------------------|--|
|               | 기능모듈              | 구현 내용  |
| 아카이브 관리기      | SP관리기<br>-기관관리    | SP의 수집대상이 되는 DP기관에 대한 각종 정보를 등록, 관리  |
| 요청기           | SP관리기<br>-작업관리    | 작업쓰레드를 사용하여 작업을 요청. 동시에 복수 DP에서 수집작업 수행 가능   |
| 스케줄러          | SP관리기<br>-스케줄관리   | -일괄구축 : 한번의 수작업으로 처리<br>-자동구축 : 스케줄(단위:일)에 따라 메타데이터를 자동으로 수집   |
| 흐름제어          | SP관리기<br>-메타수집    | 캐싱처리는 하지 않으며, DP에서 한번에 전송할 레코드 수를 설정하여 처리  |
| 갱신기능          | SP관리기             | -별도의 자동갱신 기능 제공 없음<br>-데이터동기화 기능을 제공하여 관리자가 지정한 스케줄에 따라 참여기관의 모든 데이터의 식별자를 수집하여 통합메타DB와 비교하여 데이터의 무결성 보장 |
| XML 파서        | 통합DB구축기<br>-메타분석  | -파싱과 검증작업을 수행<br>-검증이 완료된 데이터는 DB에 저장  |
| 정규화 처리기       | 관리시스템<br>-통합메타관리기 | -별도의 정규화처리 기능을 제공하지 않음<br>-통합메타관리기에서 메타데이터의 수정, 삭제, 추가 등의 기능을 제공하여 처리                                    |
| 데이터베이스        | 통합DB구축기           | DBMS를 사용하여 데이터를 저장   |
| 중복검사기         | 통합DB구축기<br>-중복검사  | 동일한 식별자(ID)를 가지는 아이템이 존재하는지를 검사  |
| 서비스 모듈        | 통합검색시스템           | -일반검색, 고급검색, 디렉토리검색<br>-이력검색, 원문보기 등   |

<표 2> DP의 핵심기능의 구현내용 비교

| OAI 권장<br>핵심기능      | KERIS 핵심기능 및 내용                           |   |
|---------------------|---|---|
|                     | 기능 모듈                                     | 구현 내용   |
| 요청분석기               | 유통시스템(DP)<br>-요청수신기<br>-유효요청확인기<br>-요청분석기 | 요청내용을 수신하여, 유효한 요청인지를 확인하고, 요청 명령어의 종류와 파라미터값을 분석     |
| 오류생성기               | 유통시스템(DP)                                 | 미리 정의된 오류메시지 코드에서 각 기능에서 요청한 오류를 선택하여 응답 구성           |
| DB탐색 및<br>메타데이터 추출기 | 유통시스템(DP)<br>-요청처리기                       | 아이템 DB에서 해당되는 메타데이터를 추출                               |
| 응답 XML 생성기          | 유통시스템(DP)<br>-요청처리기<br>-요청응답기             | 요청처리기에서 응답을 생성하고, 요청응답기를 통해 응답을 전송                    |
| 흐름제어                | 유통시스템(DP)                                 | 요청분석 과정에서 흐름제어 토큰값이 있는 경우, 토큰값을 분석하여 다음 작업을 요청처리기에 전달 |

위 표는 핵심기능의 처리내용을 대상으로 OAI의 권장사항과 KERIS의 구현내용을 비교한 것이다. 표에서처럼 기본적으로 이들간에는 커다란 차이가 없음을 알 수 있다. SP/DP의 기본요건에 해당되는 내용도 포함하여 몇가지 드러나는 차이점을 정리하면 다음과 같다.

- OAI의 권장사항보다 KERIS 사례의 SP/DP 시스템은 핵심기능이 다소 세분화되어 있음을 알 수 있다. 물론 OAI의 권장사항에서 제시되지 않은 기능들이 KERIS 사례에서는 구현되어 있기도 하다.
- OAI의 권장사항보다 KERIS 사례가 단순하게 구현되거나, 아예 구현되지 않은 기능도 일부 존재한다. 특히 SP의 갱신기능, 정규화 처리기, 중복점사기 등과 같은 기능에서는 OAI의 권장사항에 미치지 못하고 있다. 정규화 처리기능에도 차이가 많다. SP/DP 모두 SOMS라는 표준 메타데이터를 사용하고 있어 별도의 정규화 과정이 필요없었기 때문이다. 대신에 수집된 메타데이터의 질적 수준을 높이기 위하여 SP에서 메타데이터에 대하여 보정작업을 수행할 수 있도록 하는 편집기능을 제공하고 있다. 이 부분은 KERIS 생성유통 체계의 참여기관이 확대되고, 국내의 다른 DP들을 연결할 경우에 대비하여 고려해야할 사항들이라 하겠다.
- DP의 기능구현에는 커다란 차이가 없음을 알 수 있다. SP의 요청내용을 확인하고, 응답 메타데이터를 추출하고, 필요시 오류메시지를 확인하여 제공하는 수준의 기능이기에는 차이가 있을 수가 없다고 하겠다.

- KERIS 사례에서는 DP 기본요건의 하나인 논리적인 세트 구조는 적용하지 않고 있다.
- SP/DP의 데이터교환을 위하여 SOAP이라는 표준 프로토콜을 사용하고 있다. SOAP의 적용은 OAI 프로토콜에서는 아직 제안하지 않은 방식이며, KERIS에서는 전송 및 메시지 처리를 위하여 실험적인 차원에서 적용하였다.

## IV. 결 론

OAI 프로토콜은 기능적인 관점에서 보면 디지털 정보자원을 보유하고 있는 각종 디지털 도서관 시스템(archive, repository 등)으로부터 XML 형식의 메타데이터를 수확하는데 필요한 메카니즘이지만, 개념적인 관점에서 보면 개방형 디지털도서관들이 보다 효율적인 환경에서 정보자원의 생성과 유통작업을 수행하고, 정보공유체제를 형성하는 기반기술의 역할을 수행한다.

본 논문에서는 이러한 OAI 프로토콜의 등장배경과 기술요건을 검토하고, 국내에서 시도한 KERIS 사례의 특징과 OAI 권장사양의 차이를 비교검토하는 작업을 하였다. KERIS의 생성 유통체계 구축 프로젝트는 국내 처음으로 OAI 프로토콜을 적용한 사례이었던데, 경험부족의 한계에 대한 지적을 받을 여지가 충분히 있다. 그러나 그러한 요소기술 경험의 부족, 과도한 과업의 범위, 그리고 개발기간의 촉박함이라는 여건의 어려움에도 불구하고, 외국의 관련 사례들을 철저히 조사분석하여 참조하였고, 보다 안정적이고 실용적이라 평가하고 있는 2.0 버전의 프로토콜을 사용하였기에, 나름대로 의미가 있는 결과를 얻은 프로젝트로 평가할 수 있다. 그러기에 OAI에서 권장하는 SP/DP의 핵심기능과 기본요건과 그다지 큰 차이가 없이 구현될 수 있었던 것이다.

현재까지는 OAI 프로토콜이 분산환경의 디지털도서관에서 메타데이터를 수확하는 차원에서 정의되어 적용되고 있지만, 향후에는 수확대상의 메타데이터도 다양해지고, 원문콘텐츠까지 수확하도록 발전하게 될 것으로 판단된다. 즉, 서지정보의 메타데이터뿐만 아니라 저작권이나 파일의 구조정보, 원문의 보존정보 등과 같은 다양한 유형의 메타데이터가 유통되며, 원문콘텐츠도 프로토콜에 의해 교환될 전망이다.

OAI 프로토콜은 학술정보유통 영역뿐만 아니라 교육/강의 콘텐츠, 박물관 문화유산 등에 응용되는 디지털도서관뿐만 아니라 이들을 네트워크로 연결하는 통합형 디지털도서관 체제를 구축하는데 매우 중요한 기술표준임은 명백하다. 그러므로 OAI 프로토콜을 기반으로 하는 정보자원의 통합형 디지털도서관 체제는 협력기반의 정보공유를 위한 실질적인 모델이라

할 수 있다.

본 논문에서는 주로 관련된 기술적인 측면만을 다루었지만, 정보공유 모델로서의 통합형 디지털도서관 서비스 체제는 정보정책적인 관점에서 제안된 디지털 장서개발의 '소유는 특성화하고 접근은 보편화하는' 전략적 원칙<sup>22)</sup>에 부합하는 적실한 방법론으로서, 또한 개발도상국 연구자들에게 비용의 장벽이 없이 국제적인 규모의 지식공동체(global knowledge community)에 접근하는 기회를 제공하는 새로운 학술정보의 유통환경<sup>23)</sup>이 될 것이다.

### <참고문헌은 각주로 대신함>

---

22) 이재환, 디지털 시대의 도서관정보정책, 한울아카데미, 2003, 105쪽.

23) Leslie Chan, Barbara Kirsop, "Open Archiving Opportunities for Developing Countries: towards equitable distribution of global knowledge", *ariadne*, 30, 2001. [인용 2004. 5. 10].