2002년 한국정보처리학회 춘계학술발표논문집 제9권 제1호

Metadata Harvesting을 위한 service provider의 설계 및 구현

이종필*, 지용인*, 이현숙*, 이만호*
*충남대학교 컴퓨터과학과
e-mail : (jplee, jiyongin, hylee, mhlee)@cs.cnu.ac.kr

A Design and Implementation of service provider for Metadata Harvesting

jong-phil Lee*, Yong-in Ji*, Hyun-sook Lee*, Mann-ho Lee*
*Dept. of Computer Science, Chungnam National University

요 약

OAI는 간단한 프로토콜을 정의함으로써 디지털도서관 사이의 상호이용의 문제점을 해결하기 위해 제시된 프로토콜이다. OAI를 통해 디지털도서관사이의 상호이용을 가능하게 하기 위해, 디지털도서관이 가지고 있는 컨텐츠에 대한 메타데이터를 제공하기 위한 data provider와 이를 수집하여 유용한 서비스를 제공하기 위한 service provider라는 두개의 프레임웍이 필요하다. 본 논문에서는 OAI protocol을 따르는 많은 data provider들이 가지고 있는 정보들을 수집하고, 수집된 정보를 통해 새로운 서비스를 제공하는 service provider의 기능을 설계 및 구현하였다.

1. 서론

기존의 많은 디지털도서관 시스템들은 각기 다른 접근방법을 제공하며, 정보를 제공하는 형태 또한 매우 다양하여 디지털도서관 사이의 상호이용에 많은 어려움이 있었다. OAI는 이와 같은 디지털도서관 사이의 상호이용의 어려움을 해결하기 위한 방법으로 메타데이터를 수집하기 위한 간단한 프로토콜을 정의하였다. 정의된 OAI protocol을 통해 디지털도서관이 가지고 있는 컨텐츠에 대한 메타데이터를 제공하고, OAI protocol에 맞는 request를 보내어 디지털도서관이 가지고 있는 메타데이터를 수집함으로써 디지털도서관 상호간의 정보교환이 가능하게 되었다. 또한 OAI는 OAI protocol에 맞는 request를 받아들여 request에 상응하는 메타데이터를 제공하기 위한 data provider와 여러 data provider들에게 OAI protocol request를 보내어 data provider가 제공하는 메타데이터를 수집하고 이를 이용하여 사용자들에게 검색 서비스를 제공하는 service provider라는 두 개의 프레임웍을 정의하였다.

본 논문에서는 OAI가 정의한 두 개의 프레임웍중에서 service provider를 설계하고 구현하였다.

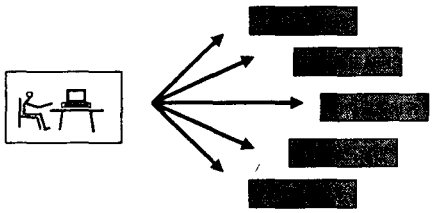
본 논문의 구성은 다음과 같다. 2 장에서는 OAI에 대한 개념을 다루고, 3 장에서는 data provider에 OAI protocol에 맞는 request를 보내어 디지털도서관들의 정보를 수집하고 이를 이용해 유용한 서비스를 제공하는 service provider에 대해서 설명한다. 그리고 4 장에서는 결론 및 향후 연구방향에 대해서 살펴본다.

2. OAI(Open Archive Initiative)

2.1 OAI의 정의 및 개념

기존의 디지털도서관 시스템들은 각 시스템마다 각기 다른 접근방법을 제공하며, [그림 1]과 같이 사용자는 디지털도서관이 제공하는 서로 다른

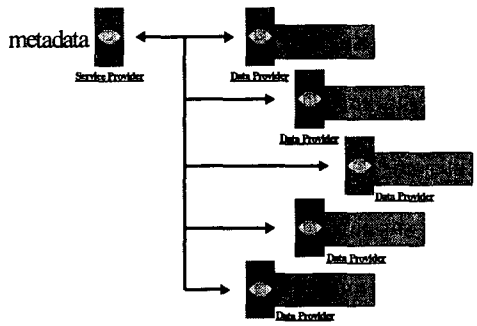
접근방법을 통해 디지털도서관에 접근하게 된다. 이는 매우 불편하며, 또한 디지털도서관 사이의 상호이용을 어렵게 한다.



[그림 1] 기존의 디지털도서관 접근방법

OAI는 이러한 디지털도서관 사이의 상호이용의 문제를 해결하기 위한 방법으로 메타데이터를 수집(Harvesting)하기 위한 간단한 프로토콜을 정의하여 디지털도서관 사이의 상호이용의 문제점을 해결하고자 하였다.

OAI는 디지털도서관이 가지고 있는 컨텐츠에 대한 메타데이터를 제공하기 위한 data provider라는 프레임웍과 여러 개의 data provider들이 제공하는 메타데이터를 수집하고 이를 이용해 유용한 서비스를 제공하는 service provider라는 두 개의 프레임웍을 정의하였으며, 이 두개의 프레임웍을 통해 디지털도서관 사이의 상호이용을 가능하게 하였다. 아래 [그림 2]는 OAI의 두 프레임웍인 data provider와 service provider를 나타내며, OAI protocol을 통해 디지털도서관에 접근하는 모습을 보여준다.



[그림 2] OAI protocol을 이용한 디지털도서관 접근 방법

즉, service provider는 OAI protocol에 맞는 request를 data provider에 보내며, data provider는 이 request를 분석하여 해당하는 정보(메타데이터)를 반환하게 된다. 이 반환된 정보를 service provider가 수집하여 유용한 서비스를 제공함으로써 디지털도서관 상호간의 정보교환이 가능하게 된다.

2.2 OAI의 구성요소

2.2.1 Repository

Repository는 OAI protocol request를 받아들일 수 있는 서버를 말하며, 이 request는 HTTP protocol 안에 포함되어져 네트웍을 통해 전송되어 진다.

2.2.2 Record

Record는 XML-encoded byte stream 형태로 제공되며, 한 record안에는 하나의 메타데이터 형태만을 제공한다. OAI는 하나의 컨텐츠(item)에 대해서 여러 가지 메타데이터 형태들을 제공하는 것을 허용한다. 따라서 하나의 컨텐츠에 대한 여러 개의 record가 존재할 수 있게 된다.

① header : 모든 record들에게 공통적으로 포함되어져 있고, 메타데이터를 수집하기 위해 필요한 정보인 unique identifier와 datestamp 정보를 포함하고 있다. unique identifier는 repository안에 있는 한 컨텐츠에 대한 metadata를 추출하기 위한 key이며, 메타데이터의 형태를 나타내는 metadataPrefix와 결합하여 한 item에 대해서 지정된 metadata format을 가진 record를 요청하기 위해 사용된다. 일반적으로 다음과 같이 표기한다.

oai:archive-identifier:record-identifier

arXiv라는 archive identifier를 가진 data provider안에 있는 quant-ph/9901001 이라는 record identifier를 갖는 unique identifier는 다음과 같이 표현한다.

oai:arXiv:quant-ph/9901001

Datestamp는 item이 생성, 삭제된 날짜 또는 item의 내용이 변경된 가장 최근의 날짜 정보를 가지고 있으며, 지정된 날짜 사이의 정보를 수집할 때 이용될 수 있다. '1999-01-01'과 같이 표현한다.

② metadata : OAI는 한 item에 대해서 여러 개의 메타데이터 형태들을 제공하는 것을 허용한다. 즉, 한 item에 대한 여러 개의 record가 존재할 수 있다. 한 record안에는 한 개의 메타데이터 형태만을 제공한다. 주로 사용되는 메타데이터 형태는 Dublin Core, rfc1807, oams이며 각 repository마다 메타데이터 형태를 정의하여 제공할 수 있다.

③ about : Record안에 있는 metadata part에 관한 data를 유지하기 위한 optional container이다. Record의 metadata 부분에 관한 저작권과 사용에 대한 제한 사항과 기간 등의 정보를 포함하고 있다.

2.2.3 Set

Record들에 대해서 선택적으로 harvesting을 가능하게 하기 위해 repository안에 있는 item들을 그룹화하기 위한 optional construct이다. 각 repository는 item들의 계층적 구조를 정의할 수 있으며 각 계층은 여러 개의 top-level node들을 가질 수 있다. 계층에서 각 node는 하나의 set이 되며 다음 3개의 구성요소로 구성되어 있다.

□ setTag : a non-space separated string of alphanumeric characters

□ setSpec : Root element로부터 actual node에 이르는 path상에 있는 각 node의 setTag들의 리스트로 포함하며 colon[:]에 의해 구별되어 진다.

□ setName : set의 목적을 표현하는 문자열이다. 즉, 어떠한 set인지를 설명한다.

Repository안에 있는 각 item들은 한 개 이상의 set에 포함될 수 있으며, 또한 어떠한 set에도 set에도 포함되지 않을 수 있다. 따라서 repository안에 있는 모든 set에 있는 정보를 수집한다 하더라도 repository안에 있는 모든 item들에 대한 record들이 검색되어지는 것은 아니다. Set의 의미와 구성에 대한 것은 OAI protocol에 정의되어있지 않으며 각 repository 나름대로 정의할 수 있다. 또한 Service provider에서는 수집해온 data들을 분류하기 위해 set을 정의할 수 있다.

2.3 OAI protocol

OAI protocol은 data provider의 정보를 요구하는 supporting protocol request와 data provider가 가지고 있는 item들에 대한 메타데이터를 요구하는 harvesting protocol request로 나뉘어 진다. 또한 각 protocol request는 3 개의 verb들로 구성되어 있으며 그 의미는 다음과 같다.

2.3.1 Supporting protocol request

① Identify : administrative, identity, community-specific information등을 포함하는 Repository에 관한 정보를 검색한다.

② ListMetadataFormats : Repository가 제공하는 메타데이터 형태들에 대한 정보를 검색한다.

③ ListSets : repository안에 구성해 놓은 set의 구조에 대한 정보를 검색한다.

2.3.2 Harvesting protocol request

① GetRecord : identifier와 metadataPrefix argument를 함께 사용하여 unique identifier로 identifier를 가지며 metadataPrefix가 지정한 메타데이터 형태를 갖는 record를 검색한다.

② ListIdentifiers : Repository로부터 수집될 수 있는 record들의 identifier들을 검색한다. set argument와 함께 사용하여 set에 있는 record들의 identifier들을 검색하는데 사용될 수도 있다.

③ ListRecords : metadataPrefix argument와 함께 사용하여 Repository로부터 metadataPrefix 형태의 메타데이터를 갖는 record들을 검색하는데 사용된다. until, from, set argument를 함께 사용하여 선택적으로 harvesting 할 수 있다.

2.4 Flow control

Service provider가 data provider에 ListRecords와 ListIdentifiers와 같은 request를 보내면, data provider가 검색된 전체 리스트를 한꺼번에 보내는 것보다 부분적으로 분할하여 보내는 것이 네트워크상의 속도나 bandwidth를 고려할 때 보다 효과적이다. Data provider가 검색된 전체 리스트 중에서 그 일부를 반환할 경우 incomplete list가 XML 문서 형태로 반환되며 그 안에 resumptionToken이 포함된다. 그 예는 다음과 같다.

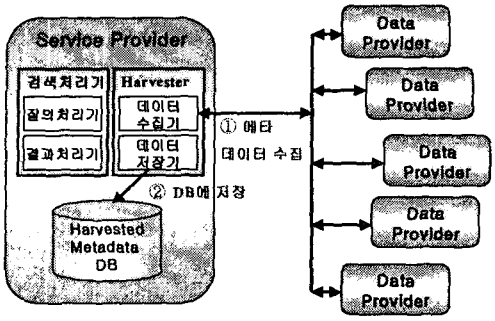
<resumptionToken>xxx45abttyz</resumptionToken>

Service provider는 반환된 문서 안에 resumptionToken이 있는지를 확인하여 있으면 다음 request에 이를 포함시킨다. 이 때 data provider는 resumptionToken을 확인하여 나머지 리스트중의 일부를 incomplete list로 만들어서 service provider에 반환한다. Data provider가 반환하는 XML 문서 안에 resumptionToken이 포함되어져 있지 않을 때까지 위의 과정을 반복하며, 이때 service provider는 incomplete list들을 결합하여 complete list를 구성하게 된다.

3. Service provider

3.1 Service provider의 설계

Service provider는 OAI protocol을 따르는 data provider로부터 메타데이터를 수집하는 부분과 수집된 메타데이터 정보를 이용하여 사용자에게 유용한 서비스를 제공하는 부분으로 나뉘어 진다. [그림 5]는 메타데이터를 수집하기 위한 service provider의 구조를 나타낸다.



[그림 5] 메타데이터 수집을 위한 service provider의 구조

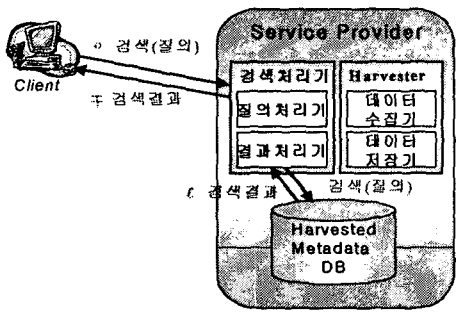
디지털도서관이 제공하는 메타데이터 정보를 수집하기 위해 service provider는 OAI protocol을 따르는 data provider에 OAI protocol에 맞는 질의어를 만들어 request를 보내게 된다. 2.3 에서 설명한 6 개의 verb를 이용하여 질의어를 만들며 그 한 예는 다음과 같다.

http://an.oa.org/OAI-script?
verb=ListRecords&from=1998-01-15
&set=physics:hep&metadataPrefix=oai_rfc1807

위 request는 physics:hep set안에 존재하는 item들 중에서 1998년 1월 15일 이후에 생성, 변경된 item들을 rfc1807 메타데이터 형태의 메타데이터를 포함하는 record들을 반환할 것을 data provider에게 요청하는 것이다.

service provider의 요청에 따라 data provider가 보내주는 정보는 XML문서로 작성되어 있다. service provider가 수집된 정보를 통해 유용한 서비스를 제공하기 위해 수집된 XML 문서를 XML DOM Parser를 통해 파싱하여 필요한 정보를 추출한 후 데이터베이스에 저장하여야 한다. 이때 추출되는 정보는 record의 header정보, 메타데이터 format 정보, 그리고 메타데이터 정보가 된다.

[그림 6]은 수집된 메타데이터 정보를 이용해 사용자에게 서비스를 제공하기 위한 service provider의 구조를 나타낸다.



[그림 6] 검색 서비스를 위한 service provider의 구조

Service provider가 검색 서비스를 제공하기 위해, 수집된 메타데이터 정보를 색인하여 검색 서비스에 이용하게 될 dictionary 파일과 inverted 파일을 생성해야 한다. Dictionary 파일은 term의 정보, term이 inverted 파일에 나타나는 위치정보, 그리고 term이 나오는 문서들의 수를 나타내는 정보를 가지는 tuple들의 리스트로 구성되어 있다. 또한 inverted 파일은 term의 정보, term이 나오는 문서 ID정보, 그리고 각 문서에서 term이 나오는 수를 나타내는 정보를 가지는 tuple들의 리스트로 구성되어 있다.

질의어가 들어오면 term 단위로 나누고, 해당 term이 dictionary 파일에 있는지를 검색한다. 찾는 term이 dictionary 파일에 있으면 dictionary 파일로부터 term의 위치 정보 및 해당 term이 나오는 문서의 수 정보를 얻고, 이를 통해 inverted 파일에 접근하여 term이 나오는 문서 ID들의 list를 얻어온다. 이 문서 ID들의 list를 통해 데이터베이스를 검색하여 문서 ID에 해당하는 정보를 얻은 후 적절히 변환하여 사용자에게 제공하게 된다.

4. 결론 및 향후 연구방향

OAI는 기존 디지털도서관들이 갖는 단점인, 접근 방법의 다양성, 제공되는 정보 형태의 다형성의 문제점을 해결하여 디지털도서관들 사이의 상호이용을 가능하도록 하였다. 본 연구에서는 OAI를 따르는 여러 디지털도서관들로부터 필요한 정보를 수집하고, 수집된 정보를 통해 검색 서비스를 제공하는 service provider를 구현하였다.

Data provider는 item들의 계층적 구조를 정의하는 set을 갖는다. Data provider들로부터 메타데이터를 수집하는 service provider가 data provider들이 구성한 set의 정보를 포용할 수 있는 set을 정의하고 이를 효율적으로 관리한다면 보다 나은 유용한 서비스를 제공할 수 있을 것이다. 앞으로 이러한 service provider의 set에 대해서 연구하는 것도 좋은 연구 방향이 되리라 생각한다.

5. 참고 문헌

[강지훈 99] 강지훈, 맹성현, 이만호, "분산 디지털도서관 시스템에서 XML을 이용한 가상문서의 표현처리," 제2회 디지털도서관 컨퍼런스, 서울, 1999년 11월.

[XML 98] W3C, Extensible Markup Language (XML) 1.0, Recommendation, Feb. 1998. (http://www.w3.org/TR/REC-xml).

[DOM 98] W3C, Document Object Model (DOM) Level 1, Recommendation, Oct. 1998. (http://www.w3.org/TR/REC-DOM-Level-1).

[OAI 01] OAI community, Open Archives Initiative(OAI). Protocol Version 1.1 of 2001-07-02. Document Version 2001-06-20

(http://www.openarchives.org/OAI/openarchivesprotocol.htm)