

디지털도서관에서 외부서비스 지원을 위한 DLIntra 확장 설계 및 구현^{*}

이광희, 안개일, 전우직
충남대학교 컴퓨터공학과

The Extended Design & Implementation of DLIntra for supporting External Service in the Digital Library

KwangHee Lee, Gaell An, WooJik Chun
Dept. of Computer Engineering, Chungnam National University

요 약

디지털도서관은 디지털로 표현된 음성, 화상, 문서 등 다양한 유형의 정보들이 통신망을 통하여 손쉽게 접근하고 원하는 형태로 표현할 수 있는 새로운 개념의 도서관이다. 우리는 분산 환경에서 도서관 모델을 연구해 왔고 디지털도서관 내부에 분산되어 저장된 데이터(내부데이터)를 검색하고 이용할 수 있는 DLIntra를 구현했다. 본 논문에서는 기존 DLIntra 구조의 수정 없이 외부서비스 제공자가 제공하는 외부데이터를 관리 및 검색할 수 있는 확장된 구조를 제안한다. 이를 위하여 외부데이터를 검색 및 등록하기 위하여 프로토콜 게이트웨이와 외부데이터 등록 에이전트를 설계 및 구현하였다.

1. 서론

디지털도서관이란 기존의 도서관 기능을 전산화하는 단순한 개념이 아니라 디지털로 표현된 음성, 화상, 문서 등 다양한 유형의 정보들을 통신망을 통하여 손쉽게 접근하고 원하는 형태로 표현할 수 있는 새로운 개념의 도서관을 말한다. 우리는 분산 환경에서의 디지털도서관 참조 모델 및 구조 연구[1]를 수행해 왔고 디지털도서관 내부에 분산되어 저장된 데이터를 검색하고 이용할 수 있는 DLIntra(Digital Library Infrastructure)를 구현했다. DLIntra는 분산객체 기반의 구조로 각 분산객체들이 서로 협력함으로써 서비스를 제공한다. DLIntra의 분산객체는 독자성과 자치성의 성격을 갖고 행동하므로 에이전트적이다.

[1]의 연구는 DLIntra 내부에서 유지 및 관리하는 데이터(내부데이터)에 대한 검색서비스에 초점이 맞추어져 있다. UMDL[2], 스탠포드[3], CNRI[4]에서는 디지털도서관에 관한 연구를 수행했다. 스탠포드의 InfoBus 경우, 외부데이터를 직접 사용자에게 전달을 하는데 인하여 DLIntra는 외부데이터를 내부데이터로 변환을 한 후에 사용자에게 전달한다. DLIntra는 외부데이터가 내부데이터로 변환되기 때문에 InfoBus보다 더 많은 서비스 준비 시간이 요구된다. 그러나, 만약 사용자가 이전에 검색된 데이터를 다시 요청한 경우라면 그 데이터는 이미 내부데이터로 DLIntra에 저장되어 있기 때문에 스탠포드의 InfoBus보다 더 빨리 서비스를 제공할 수 있다.

본 논문은 DLIntra 구조를 확장하여 내부데이터뿐만 아니라 외부서비스 제공자(Web Search Engine, ZServer)가 관리하는 데이터(외부데이터)를 관리 및 검색할 수 있도록 설계 및 구현을 기술한다.

2. 본론

2.1 내부데이터를 위한 DLIntra Overview

DLIntra는 LSPG(Library Service Protocol Gateway), UA(User Agent), RA(Retrieval Agent), HS(Handle Server), SA(Storage Agent)로 구성되어 있다. LSPG는 사용자 프로토콜을 DLIntra에서 정의된 LSP(Library Service Protocol)로 변환하고 UA에게 서비스 처리 요구를 한다. UA는 검색 요구를 받으면 문서 식별자를 얻기 위해 RA에게 문서 식별자 검색을 요구하고 문서의 저장된 위치 정보를 얻기 위해 HS에게 문서 식별자에 대응하는 행들 검색을 요구한다. UA는 초록정보를 얻기 위해 SA에게 행들을 넘겨주고 SA로부터 받은 초록정보를 사용자에게 넘겨 줌으로써 초록정보 검색을 종결한다. DLIntra는 분산 에이전트 기반 구조이기 때문에 에이전트를 이용하여 DLIntra에 분산되어 저장된 데이터를 검색할 수 있다(그림 1의 내부 데이터 검색과정 참조) 그러나, 외부에 존재하는 데이터에 대한 서비스 요구를 지원하지 않기 때문에 새로운 에이전트 정의와 구현이 필요하다.

2.2 외부서비스 지원을 위한 DLIntra의 설계

DLIntra에서 외부에 존재하는 데이터에 대해 서비스를 지

^{*} 본 연구는 소프트웨어 연구센터의 디지털도서관 과제의 일부로 수행되고 있다.

원하려면 외부데이터를 내부데이터화해서 DLInfra 에 저장해야 하므로 문서 식별자를 부여하고 저장 및 관리를 위해 문서 핸들(Doc-Handle)을 생성한다. 외부데이터는 빈번한 삽입과 삭제가 일어나므로 임시로 문서 식별자를 할당해야 하며 그 외부데이터가 이미 DLInfra 에 저장되어 있는지 판단하기 위해 외부데이터를 얻은 URL 과 초록정보, 초록정보 길이의 정보를 이용한다. 그러므로 이러한 일을 담당하는 새로운 에이전트의 설계와 구현이 필요하고 기존에 존재하는 에이전트들도 확장이 필요하다. 다음은 DLInfra 에서 외부서비스 지원을 위해 필요한 에이전트를 정의한 것이다.

- RA : 다음과 같이 두 가지 에이전트로 구성된다.
 - > Meta-RA : Working-RA 에 대한 통합된 인터페이스를 제공하고 검색된 문서 식별자에 대한 간단한 랭킹(Ranking)을 수행하는 에이전트
 - > Working-RA . DLInfra 에서 정의된 서비스에 따라 다양하게 존재하며 문서 식별자를 추출하기 위한 실제적인 검색을 수행하는 에이전트이다. 예를 들어 Working RA for Web, Working RA for Z39, Working RA for Internal 등의 Working RA 가 있다
- LAPG(Library Access Protocol Gateway) : 외부서비스를 위해 정의된 에이전트로서 내부 LSP 프로토콜을 외부 프로토콜로 변환하여 외부서비스 서버(웹 검색 엔진, ZServer)에 서비스를 요청하는 에이전트이다.
- IRA(Information Registry Agent) : LAPG 로부터 얻어진 결과를 DLInfra 에 등록하는 에이전트이다.
- SA : 기존의 SA 는 내부데이터를 위한 인터페이스만 정의되어 있었기 때문에 외부 문서 저장 및 관리를 위한 인터페이스가 추가되어야 한다

2.3 외부서비스 지원을 위한 DLInfra 동작 시나리오

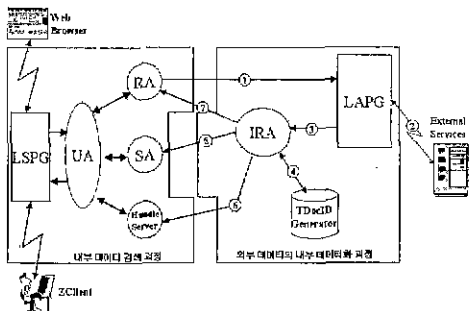


그림 1 외부 문서 등록 과정

<그림 1>은 DLInfra 에게 외부데이터를 등록하는 과정을 나타내고 있다.

- ① UA 로부터 외부서비스 요청을 받은 RA 는 LAPG 에게 외부서비스 검색을 요청한다.
- ② LAPG 는 LSP 를 외부서비스 프로토콜로 변환한 다음 외부서비스 서버(웹 검색 엔진, Z39.50 서버)에 검색을 수행한다.
- ③ LAPG 는 외부서비스 서버로부터 결과를 얻어 구문 검색(Parsing)을 수행하여 필요한 데이터만을 추출하여 IRA 에게

등록을 요구한다.

- ④ IRA 는 외부데이터에 대해 임시 문서 식별자를 부여하기 위해 임시 문서 식별자 생성기에게 문서 식별자 생성을 요구한다. 임시 문서 식별자 생성기는 등록하려는 데이터가 기존에 존재하는 데이터가 아니면 새로운 임시 문서 식별자를 생성하여 넘겨준다.
- ⑤ 외부데이터에 대해 임시 문서 식별자를 얻은 IRA 는 데이터의 저장을 위해 SA 에게 데이터 저장과 문서 핸들 생성을 요구한다 SA 는 외부데이터에 대한 문서 핸들 생성을 위해 자신이 관리하는 데이터베이스에 임시 문서 식별자, 문서 제목, 초록정보, 문서 형식(텍스트, 웹 문서), 프로토콜, URL, 검색 횟수, 생성 일자로 이루어진 레코드를 생성한다. 이중 프로토콜과 URL 필드는 사용자가 전체 문서 검색을 요구할 경우 LAPG 를 호출하기 위해 이용되는 정보이다.
- ⑥ 문서 핸들을 얻은 IRA 는 저장을 위해 핸들 서버에 문서 핸들 저장을 요구한다.
- ⑦ 마지막으로 RA 에게 얻어진 임시 문서 식별자를 넘겨 줌으로서 외부데이터를 등록하는 과정이 끝나게 된다 이후의 문서 검색 시나리오는 내부데이터 검색 시나리오와 같다.

2.4 인터페이스 정의

<그림 2>에 DLInfra 의 에이전트들에 대한 간략한 인터페이스가 잘 나타나 있다. 인터페이스의 정의는 DLInfra 가 분산 환경 개발 도구인 ILU[5]를 사용하여 개발되었으므로 ILU 에서 제공하는 ISL(Interface Specification Language) 형태로 기술되었다. ISL 은 CORBA 의 IDL 일종인 인터페이스 정의 언어이다 <그림 3>, <그림 4>는 LAPG 인터페이스와 IRA 인터페이스를 ISL 로 기술한 것이다.

LSPG	Search() More-Briefs() Get-Full() Present-Briefs() Present-Full()	HS	Request-Handle() Request-HandleAs() Store-Handle()
LAPG	Get-Item-From-External() Get-Full-From-External()	SA	Get-Briefs() Get-Full() Put-Full() Deposit-Briefs()
UA	Search() Get-DocIDs() Get-Full() Put-DocIDs() Put-Briefs() Put-Full()	RA	Search() Get-DocIDs() Put-DocIDs()
		IRA	Put-External-Items()

그림 2 DLInfra 에이전트의 인터페이스

```

INTERFACE LAPG IMPORTS BASE END,

TYPE LAPG = OBJECT COLLECTIBLE
METHODS
ASYNCHRONOUS Get-Item-From-External(Query BASE QUERY, MsgID : SHORT
INTEGER, StartPoint : SHORT INTEGER, ItemNB : SHORT INTEGER, Target
BASE REFERENCE, Server : SHORT INTEGER),
ASYNCHRONOUS Get-Full-From-External(MsgID : SHORT INTEGER, URL :
BASE STRING, Server : SHORT INTEGER, Target : BASE REFERENCE)
END;
    
```

그림 3 LAPG 인터페이스의 ISL 기술

```

INTERFACE IRA IMPORTS BASE END,

TYPE IRA = OBJECT COLLECTIBLE
METHODS
ASYNCHRONOUS Put-Ext-Items(MsgID : SHORT INTEGER, ItemNB : SHORT
INTEGER, TotalNumOfItem : INIEGER, ExtItems : BASE EXTERN-ITEMS, Target :
BASE REFERENCE)
END,
    
```

그림 4 IRA 인터페이스의 ISL 기술

2.5 개발 환경 및 사용자 인터페이스

Sun sparc 2 의 Sun OS 5.4 인 워크스테이션에 1993 년부터 개발 되어온 분산 환경 개발 도구인 Xerox 사의 ILU(Inter-Language Unification, 2.0 alpha12)를 사용하여 분산 객체를 생성하였다.

Z399.50 프로토콜을 사용하는 사용자들은 Z39 50 프로토콜 구현제품(CNIDR 의 Isite[6])에서 제공하는 ZClient 프로그램을 이용하여 손쉽게 DLInfra 를 사용할 수 있다 외부데이터의 저장, 관리, 검색을 위해 Hughes Technology 사의 Mmi SQL 2.0 을 사용하였다.

DLInfra 의 각각의 에이전트들은 세 시스템에 분산되어 구현하였다.

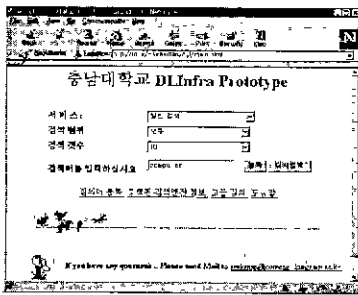


그림 5 사용자 입력 화면

<그림 5>는 웹 브라우저를 이용한 사용자 입력 화면이다. 입력 화면에서 보듯이 사용자는 서비스와 검색 범위를 선택한 다음 검색하고자 하는 질의어를 입력한다. 이런 단순한 질의어 입력 외에도 고급 질의어를 선택하면 불리언 연산자 (Boolean Operator. AND OR, ANDNOT)와 검색어의 검색 위치(제목, 문서 전체, URL)까지 지정할 수 있으므로 좀더 정확한 결과를 얻을 수 있다

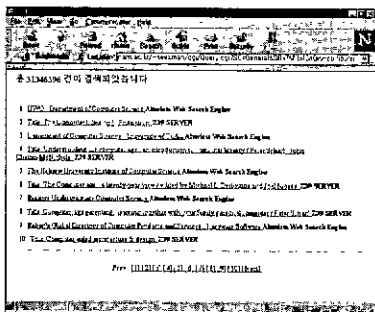


그림 6 초록정보 출력 화면

<그림 6>은 질의어 “computer”에 대한 검색 결과이다. 화면에서 보듯이 전체 검색 건수, 초록정보 그리고 어디서 검색되어 왔는지를 나타내는 위치 설명이 있고 화면 하단에는 인덱스가 나타나 있다. 다음 초록정보를 보고 싶으면 인덱스를 클릭하면 된다. 나열된 초록정보 중 전체 문서를 보고자 한다면 단지 원하는 초록정보를 클릭하면 DLInfra 에서 검색하여 화면에 표시된다.

3. 결론

지금까지 DLInfra 에서 외부서비스 지원을 위해 설계 및 구현에 대해 에이전트의 정의, 동작 시나리오, 인터페이스 정의, 사용자 인터페이스 등을 언급했다. 현재까지 구현된 DLInfra* 는 HTTP, Z39.50 을 프로토콜을 지원하고 내부에 저장되어 있는 데이터 뿐만 아니라 외부에 저장되어 있는 데이터들도 동일한 인터페이스를 통해 검색할 수 있다. DLInfra 에서는 새로운 서비스 지원을 하기 위해 기존의 구조는 변경할 필요가 없고 단지 그 서비스를 제공할 새로운 에이전트만 DLInfra 에 추가하면 된다. 예를 들어 주기적으로 검색 결과를 디지털 메일로 받는 서비스일 때 이 일을 담당할 에이전트 추가 만으로 손쉽게 이루어 질 것이다

지금까지 연구된 DLInfra 에서 성능을 높이기 위해 질의어 히스토리(History) 기능, 검색된 결과 셋의 재사용을 위해 캐쉬 및 저장 관한 연구 및 구현이 요구된다.

4. 참조문헌

- [1] 안개일, 이경희, 전주직, 신동욱, “분산 환경에서의 디지털도서관 참조 모델 및 구조 연구”, 정보과학회 가을학술논문집 1 권, pp. 361-364, 1997.
- [2] William P Birmingham, “An agent-based architecture for digital libraries”, D-Lib Magazine, Jul 1995.
- [3] Andreas Paepche, Steve B. Cousins et al 5 “Towards Interoperability in Digital Libraries: Overview and Selected Highlights of the Stanford Digital Library Project”, IEEE Computer Magazine, May, 1996.
- [4] Robert E, Kahn, William Y. Arms, “Digital Object Architecture Project”, <http://www.cnri.reston.va.us/home/doi.html>
- [5] Bill Janssen, Mike Spreitzer, “ILU Reference Manual”, Xerox Corporation, http://ftp.xerox.com/pub/ilu/2.0a12/manual.html/manual_toc.html, 1998.
- [6] CNIDR, “Isite Information System”, <http://vinca.cnidr.org/software/Isite/guide.html>

* DLInfra 는 <http://rose.comeng.chungnam.ac.kr/~seekman/cgi/main.html>에 접속하면 이용할 수 있다.