□ 기술해설 □

전자 도서관 표준 프로토콜 개요 및 구현 방안

충남대학교 안개일*·전우직**

● 목 차 ●

1. 서 론
2. 전자 도서관 표준 프로토콜
 2.1 Z39.50 표준의 역사
 2.2 Z39.50 표준의 범위
 2.3 Z39.50의 정보검색 서비스
3. 기존의 전자 도서관 구현 사례
 3.1 Zprise 시스템
 3.2 Willow 시스템
 3.3 Isite 시스템
 3.4 MIRAGE 시스템
4. 웹상에서의 Z39.50 프로토콜 구현
 4.1 세 가지 구현 모델 분석
 4.2 자바 클래스 라이브러리 방식에 기반한 구현
5. 결 론

1. 서 론

현대인들은 시시각각으로 불어나는 엄청난 정보의 홍수 속에서 살아가고 있다. 이러한 환경속에서 사용자가 원하는 정보를 정확하고, 빠르고 그리고 손쉽게 제공받을 수 있도록 서비스를 증대화시키는 것은 매우 중요한 문제이다.

이러한 문제는 컴퓨터기기와 통신기술 및 저장 장치 등 제반기술의 획기적인 발전에 따라서, 사용자의 요구를 점차적으로 만족시켜가고 있다. 먼저, 스캐너, 문자인식기 등의 발전으로 많은 정보가 쉽게 전자로 표현되고 있으며, 이러한 전자정보는 대량의 정보를 조작하고, 관리할 수 있는 데이타베이스 시스템 및 정보 검색 시스템에 저장될 수 있다. 또한, 세계 최대의 통신망인 인터넷이 구성되어 수백만 명의 사람들이 이를 통해 전자 정보를 이용하고 있다.

따라서 이러한 제반기술을 통하여, 방대한 양의 정보를 체계적으로 관리하고 이러한 정보를 정확하고 쉽고 편리하게 접근하기 위한 한 방안으로 전자 도서관의 연구가 활발히 진행되고 있다[10]. 미국을 중심으로 유럽을 포함한 많은 나라에서 전자 도서관에 대한 연구를 수행하여, 현재는 시스템 원형(prototype)을 구축한 상태이다[6][8][11][12][13]. 국내에서는 충남대학교(MIRAGE 시스템 구축)[1][3]와 숙명여대(Isite를 이용한 전자 도서관 시스템 구축)[2] 등에서 이에 대한 연구를 하고 있으나, 전자 도서관 본래의 모든 기능을 수행하는 시스템은 아직 개발하지 못하고 있다.

이러한 전자 도서관을 개발하는데 있어서 많은 핵심 기술들이 필요한데, 그 중의 하나가 바로 Z39.50이라는 전자 도서관 표준 프로토콜이다. Z39.50은 OSI 참조 모델의 응용 계층(Application Layer)에 해당되는 프로토콜로써, 클라이언트는 서버에게 정보를 요청하고 이 요청에 따라 서버는 클라이언트에게 정보를 제공하는 프로시저와 정보 표현 형식을 정의하여 이기종간의 상호 운용성을 보장하는 프로토콜이다.

본 고에서는 전자 도서관 표준프로토콜의 기본적인 기능을 소개하며, 기존의 전자 도서관 구현사례에 대하여 살펴본다. 또한 웹 상에서 직접 전자 도서관을 효율적으로 접근할 수 있

*학생회원
**정 회 원

도록 하기 위하여, 가능한 모델들을 제시하여, 이를 기반으로 바람직한 구현 모델을 제안하고자 한다.

2. 전자 도서관 표준 프로토콜

전자 도서관 표준 프로토콜은 정보 검색이 목적인 클라이언트/서버 모델에 기반한 프로토콜로써 Z39.50 프로토콜[5]이라고 한다. 이 프로토콜에서는 클라이언트가 서버에 의해 제공된 데이타베이스를 검색하고, 검색된 결과를 추출할 수 있는 기본적인 기능과 자원 관리, 도움 기능, 결과 정렬 등의 서비스 기능을 제공하기 위한 프로시저와 구조를 정의하고 있다.

2.1 Z39.50 표준의 역사

Z39.50 표준은 문헌 정보를 검색하기 위하여 1984년에 처음으로 등장하였다. 1988년에 ANSI(American National Standard Institute)와 NISO(National Information Standards Organization)에 의하여 Z39.50이라는 정식 표준(Z39.50-1988)이 최초로 탄생하였다. 그 후 1989년부터는 NISO를 대신하여 국회 도서관에서 관장하는 Z39.50 관리 기관(Z39.50 Maintenance Agency)이 새로이 설립되었다. 1990년에는 제조업자, 밴더, 대학교 등이 주축이 되어 ZIG(Z39.50 Implementors Group)라는 그룹을 결성되었고, 실질적인 영향력을 행사하고 있다. 1992년에 Z39.50 관리 기관이 국제표준에서 정의된 ISO 10162/10163, Search and Retrieve와 호환성을 갖기 위하여 Z39.50-1988 표준을 개선하여 Z39.50-1992 표준을 생성하였다. 1995년 7월에 버전 3인 Z39.50-1995 표준이 Z39.50 관리기관이 가장 최근에 제안한 표준이다.

2.2 Z39.50 표준의 범위

Z39.50은 이기종들간의 상호 운용성을 보장해 주며 OSI 참조 모델의 응용 계층(Application Layer)에 해당되는 프로토콜이다. Z39.50에 정의된 프로시저와 형식에 의하여, 클라이언트는 서버에게 정보를 요청하고 이 요청에 따라 서버는 클라이언트에게 정보를 제공한다.

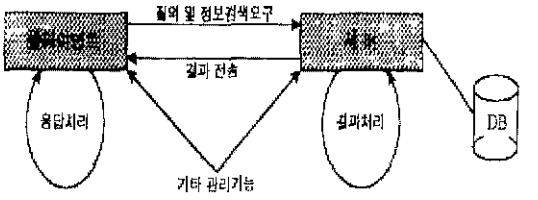

그림 1 Z39.50 프로토콜의 제공기능

다음은 Z39.50표준이 지원하는 기능이다(그림 1 참조).

- 질의 기능 : 클라이언트가 서버에게 원하는 정보를 질의할 수 있게하는 기능으로 검색에 필요한 질의 및 다양한 종류의 파라메타를 전송하여 검색방식, 응답의 범위와 표현방식, 질의결과처리방식 등을 지정한다.
- 정보검색 기능 : 정보 요구에 따라 서버에서 검색된 결과를 전체 또는 일부를 검색하는 기능을 지원한다.
- 응답처리 기능 : 검색된 결과는 사용자에 의해 이름이 지정되어 보관되고 보관된 결과를 다음 질의에 이용할 수 있는 기능을 지원하며 이 결과에 대하여 종합, 순차배열, 상세검색등을 지원한다.
- 결과처리 기능 : 클라이언트의 검색요구를 데이타베이스를 검색하여 결과세트를 만들어 내며, 결과검색 요구시에는 결과세트를 검색하여 클라이언트에게 제공하는 기능을 갖고 있다.
- 결과전송 기능 : 클라이언트가 요구한 검색요구 및 검색결과를 요구하는 정보검색요구에 응답하는 기능을 제공한다. 요구한 정보검색 결과를 서버는 여러 개의 메시지로 나누어 전송할 수 있는 기능도 제공한다.
- 기타 관리기능 : 위에서 설명한 기본기능 외에도 서버의 상태, 사용자접근제어, 문서교환형식 및 사용문자의 협상 등 다양한 관리 기능을 지원한다.

2.3 Z39.50의 정보검색 서비스

Z39.50은 메세지 요구(request)와 응답(response)으로 이루어진 확인 서비스(confirmed

service), 요구만 있고 응답은 없는 비확인 서비스(non-confirmed service) 및 확인 서비스와 비확인 서비스가 모두 가능한 제한적인 확인 서비스(conditionally-confirm service)를 정의하고 있다. 이러한 서비스들은 Z39.50표준에서 정의하고 있는 11개의 정보 검색 서비스 기능에서 사용된다(그림 2 참조).

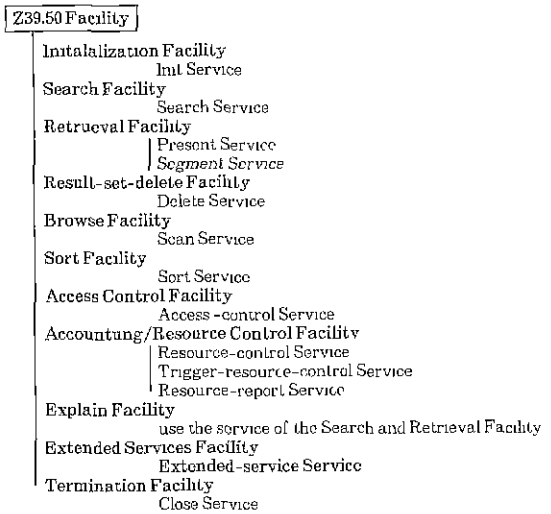

그림 2 Z39.50 프로토콜의 기능

다음은 11개의 정보 검색 서비스 기능들에 대한 설명이다.

1) 접속 기능(Initialization Facility)

접속 서비스(Init Service)가 정의 되어 있으며, 접속 서비스는 서버와 접속하기 위하여 클라이언트가 요구하는 확인 서비스이다. 이때 클라이언트와 서버는 파라메타를 사용하여 이 세션에서 지원할 서비스(검색, 정보검색, 삭제, 동시검색 등)를 설정하며, 또한 전송메시지크기 등을 협상한다.

2) 질의 기능(Search Facility)

질의 서비스가 정의되어 있으며, 질의 서비스는 서버에 있는 데이타베이스에 질의를 하여 검색된 결과를 클라이언트에서 전송받을 수 있도록 클라이언트가 요구하는 확인 서비스이다. 이때 클라이언트와 서버는 파라메타를 사용하여 질의의 형태, 데이타베이스 이름, 결과세트 이름, 검색된 레코드 수 등의 정보를 서로 주고 받는다.

3) 정보검색 기능(Retrieval Facility)

전송(Present)서비스와 세그먼트(Segment) 서비스가 정의되어 있다.

전송 서비스는 검색된 결과세트로부터 원하는 레코드를 전송받기 위하여 서버에게 클라이언트가 요구하는 확인 서비스이다. 이때 클라이언트와 서버는 파라메타를 사용하여 결과세트번호, 클라이언트가 요구한 레코드의 수, 결과세트 시작위치, 응답 레코드 등의 정보를 서로 주고 받는다.

세그먼트 서비스는 검색된 결과세트로부터 클라이언트가 원하는 레코드들을 서버가 단 한번에 제공하고자 할 때 요구하는 비확인 서비스이다. 이때 서버는 세그먼트 레코드 및 응답 레코드의 수 등의 정보를 가지고 있다. 클라이언트가 정보검색 서비스를 사용하여 레코드들을 요구할 때 서버는 세그먼트 서비스로 레코드를 전송하며 마지막 레코드는 정보검색 서비스를 사용하여 제공한다(그림 3 참조). Z39.50 표준에서는 세그먼트 수준 1과 세그먼트 수준 2가 정의되어 있다. 세그먼트 수준 1은 서버는 레코드를 여러 개의 세그먼트로 분할하여 한번에 전송하는 기능이 있으며, 세그먼트 수준 2는 만약 레코드가 한 세그먼트 안에 들어 가지 않을 때에, 그 레코드를 여러 개의 조각으로 분할하여 보내는 기능이 있다.

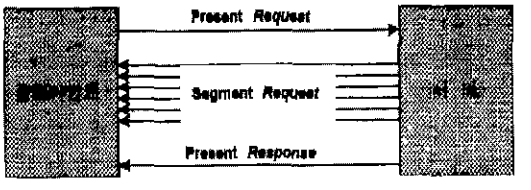

그림 3 세그멘트 서비스를 이용한 정보검색

4) 결과세트삭제기능(Result-set-delete Facility)

삭제(Delete) 서비스가 정의되어 있으며, 삭제 서비스는 검색된 결과세트를 삭제하기 위해 클라이언트가 요구하는 확인 서비스이다. 이때

클라이언트와 서버는 파라메타를 사용하여 삭제할 결과세트번호, 전체 또는 일부의 결과세트를 삭제하는지 여부 및 삭제결과 상태를 나타내는 상태정보 등을 서로 주고 받는다.

5) 브라우저 기능(Browse Facility)

브라우저 기능으로는 스캔(Scan) 서비스가 정의되어 있으며, 이 서비스는 사용자가 데이타베이스를 편리하게 검색할 수 있도록 질의할 수 있는 용어(term)를 검색할 수 있게 클라이언트가 요구하는 확인 서비스이다. 이때 클라이언트와 서버는 파라메타를 사용하여 데이타베이스 이름, 용어 리스트, 전송될 항목 및 항목의 수 등의 정보를 서로 주고 받는다.

6) 정렬 기능(Sort Facility)

정렬 기능으로 정렬 서비스가 정의되어 있다. 이 서비스는 사용자가 결과세트를 정렬할 수 있도록 클라이언트가 요구하는 확인 서비스이다. 이때 클라이언트와 서버는 파라메타를 사용하여 입력 결과 세트, 정렬된 결과 세트, 정렬 상태 등의 정보를 서로 주고 받는다.

7) 접근제어 기능(Access Control Facility)

접근제어 서비스가 정의되어 있으며, 이 서비스는 보안을 가진 서비스를 사용자가 요구할 때 제어하기 위하여 클라이언트가 요구하는 확인 서비스이다. 이때 클라이언트와 서버는 사용자의 신분을 확인할 수 있는 정보를 통해 접근제어 서비스를 수행한다.

8) 자원 관리 기능(Accounting/Resource Control Facility)

자원관리 기능에는 자원제어(Resource-control) 서비스, 트리거자원제어(Trigger-resource-control) 서비스, 자원리포트(Resource-report) 서비스가 정의되어 있다.

자원제어 서비스는 서버의 상태(에, 서버의 자원 부족 또는 수행중인 결과 세트 상태)를 클라이언트에게 알려 주며 현재 수행중인 서비스를 계속할지 또는 중지할지를 요구하기 위하여 서버가 요구하는 확임 및 비확인 서비스이다. 이때 서버와 클라이언트는 자원리포트, 연산중지여부, 응답여부요구, 연산을 계속 수행할지여부 등의 정보를 서로 주고 받으며 서비스를 수행한다.

트리거자원제어 서비스는 클라이언트가 서버에서 수행중인 연산을 중지시킬 수 있는 액션과 서버의 상태를 요구할 수 있는 액션, 그리고 서버의 상태를 요구하면서 서버와 협상할 수 있는 액션 등 3가지 액션을 제공하기 위하여 클라이언트가 요구하는 비확인 서비스이다. 두번째 및 세번째 액션의 결과로 서버는 자원제어 서비스를 수행하여, 다시 서버가 클라이언트에게 확인또는 비확인 서비스를 할수 있게 한다. 클라이언트의 액션종류(연산중지, 자원리포트, 자원제어)에 따라서 만약 액션이 연산중지라면 중지할 검색 연산의 결과세트정보를 명세해야 하며, 만약 자원리포트 및 자원제어라면 자원리포트 형식 정보를 명세해야 한다.

자원리포트 서비스는 클라이언트가 서버에서 수행중인 연산의 상태를 알기 위하여 클라이언트가 요구하는 확인 서비스이다. 이때 클라이언토와 서버는 파라메타를 사용하여 연산식별자, 자원리포트형식, 자원리포토 등의 정보를 서로 주고 받으며 서비스를 수행한다.

9) 설명 기능(Explain Facility)

설명 서비스가 정의되어 있으며, 이 서비스는 클라이언트가 현재 연결 중인 서버에 대한 정보(서버의 이름, 데이타베이스 이름, 데이타베이스에 대한 설명, 필드이름 등)를 제공받을 수 있도록 클라이언트가 요구하는 확인 서비스이다.

10) 확장서비스기능(Extended Services Facility)

확장 서비스가 정의되어 있으며, 이 서비스는 클라이언트가 서버에 있는 결과세트나 질의를 저장하거나, 데이타베이스를 갱신할 수 있도록 클라이언트가 요구하는 확인 서비스이다.

11) 해제 기능(Termination Facility)

종료(Close) 서비스가 정의되어 있으며, 이 서비스는 클라이언트 또는 서버가 모든 연산을 중지하고 Z39.50세션을 해제할 수 있도록 하는

기능을 제공하기 위하여 클라이언트 또는 서버가 요구하는 확인 서비스이다.

3. 기존의 전자 도서관 구현 사례

Z39.50프로토콜에 기반하여 전자 도서관 원형 구축이 현재 활발히 진행되고 있다. 그 중에서 NIST에서 개발한 Zprize와 워싱턴대학에서 구축한 Willow(the Washinton Looker-upper Layered Over Windows)시스템, 그리고, CNIDR(Clearinghouse for Networked Information Discovery and Retrieval)에서 개발한 I-Site를 소개하며, 마지막으로 국내의 충남대학교에서 개발한 MIRAGE(Multimedia Information Retrieval and Gathering Environment)시스템에 대해 살펴보겠다.

3.1 Zprise 시스템

ZPRISE[9]은 UNIX 환경에서 클라이언트/서버를 기반으로 하는 정보 검색 응용계층 프로토콜인 Z39.50-1995 문서중에서 일부만을 구현한 것이다. ZPRISE는 검색 엔진(PRISE)및 독자적으로 인덱싱 기능을 하는 모듈을 가지고 있다. 서버는 Z39.50 프로토콜의 기능과 검색 엔진이 분리되어서, 서버내의 검색 엔진 인터페이스를 만들어 PRISE가 아닌 다른 검색 엔진과 연동하기 편하도록 설계되어 있다.

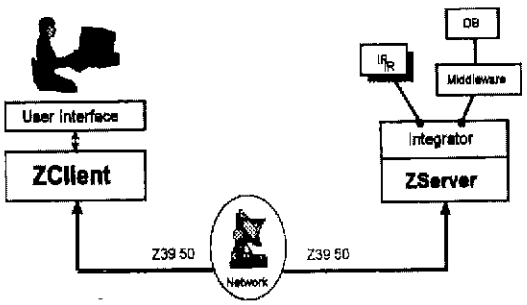

그림 4 Zprise의 모델 구조

그림 4와 같이 ZPRISE는 사용자의 요구를 받아들이는 클라이언트 부분과 클라이언트에서 온 연산을 수행하는 서버 부분으로 나누어져 있다. 그리고 각 클라이언트와 서버는 PDU를 인코딩/디코딩하기 위한 모듈이 있고, 서버에는 대상 문서들을 색인하여 사용자의 정보검색을 제공하는 모듈이 있다.

3.2 Willow 시스템

Willow[14]은 워싱턴 대학의 도서관에서 정보를 추출할수 있는 일반적인 목적의 전자 도서관이다. Willow는 많은 내용의 텍스트로 된 문서를 기반으로한 서지 데이테베이스에 접근하여 검색하며, 사용하기 편한 그래픽 인터페이스를 제공한다. 이것은 Motif를 사용하여 구현하였으므로 많은 기종의 컴퓨터에서 X-emulator만 가지고 있으면 어디서나 Willow의 서비스를 받을 수있는 장점을 가지고 있다.

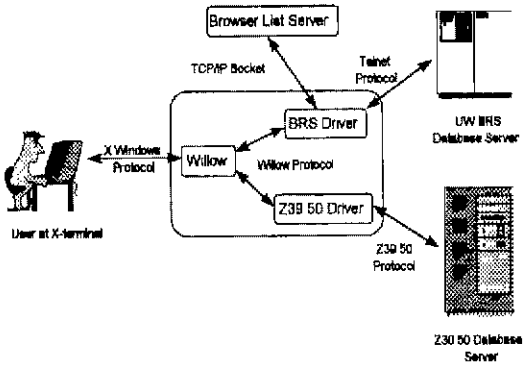

그림 5 Willow의 모델 구조

그림 5에 Willow의 모델구조가 나타나 있다. Willow는 클라이언트내에 Z39.50 프로토콜 드라이버를 가지고 있어서 이 프로토콜을 지원하는 다른 시스템들과 통신을 할수 있으며, 또한 만약 어떤 데이타베이스에 접근하려 한다면 Willow는 디바이스 드라이버 대신에 데이타베이스 드라이버를 사용하여 그 데이타베이스의 검색을 지원하고 있다.

3.3 Isite 시스템

Isite[7]는 NSF(Natinal Science Foundation)의 지원으로 CNIDR에서 개발되었다.

Isite는 표준 인터넷 프로토콜인 Z39.50, World-Wide Web(WWW), Electronic Mail 그리고 Gopher를 기반으로한 여러가지의 접근

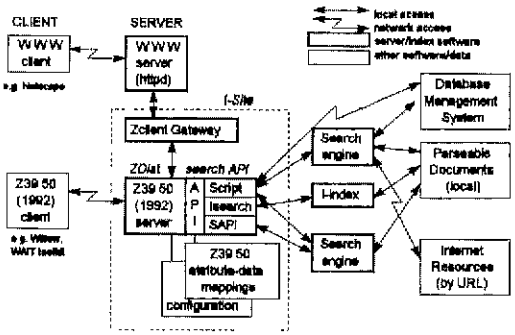

그림 6 ISite의 모델 구조

방법을 제공한다. 그 중 ANSI/NISO의 Z39.50 프로토콜이 Isite의 주요한 접근 방법이며, 다른 프로토콜들은 게이트웨이(gateway) 소프트웨어를 통해서 접근할 수 있다. Isite는 HTTP 서버에 의해서 받은 검색 질의를 Zclient 게이트웨이를 통하여 Z3.50 서버가 이해할 수 있도록 변환한다(그림 6 참조).

Isite에서는 또한 지원하는 검색 API를 통해서 다른 형태의 데이타베이스에 접근할 수 있는 방법을 제공한다. 즉, 새롭게 설치된 데이타베이스에도 Isite에 의해서 제공되는 모든 데이타 추출 통신 프로토콜을 통해서 즉시 접근할 수 있다. 이 검색 API(SAPI)는 Isite에 있는 Z39.50 서버와 데이타베이스 시스템사이에 추상적인 계층을 제공하며, 각 데이타베이스가 제공해야 할 기능들에 대한 집합을 정의하고 있다.

3.4 MIRAGE 시스템

MIRAGE [1]는 그림 7과 같이 크게 사용자 인터페이스, 통신 프로토콜, 정보검색, 데이타베이스 등 총 네개의 부분으로 나누어져 있다.

사용자 인터페이스는 멀티미디어 검색엔진의 기능과 전자 도서관 표준 프로토콜과 관련된 기능을 지원하고 사용자가 쉽게 익혀 사용할 수 있도록 그래픽 인터페이스로 작성 되었다. 정형질의와 비정형 질의가 연결되는 각각 데이타베이스와 정보검색엔진으로 전달될 수 있도록 구성되었다. 또한 화면에 디스플레이된 문서에서 다른 참고문서나 화상, 음성등을 직접 얻을 수 있는 hyper-link기능을 제공한다.

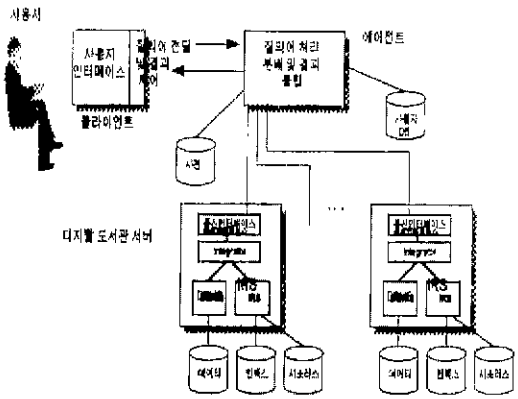

그림 7 MIRAGE의 모델 구조

통신 프로토콜은 클라이언트/서버 모델을 기반으로한 전자 도서관 표준 프로토콜인 Z39.50을 따라 설계되었으며, Zprise 시스템을 기초로 하여 한글 및 멀티미디어 데이타를 이용가능하도록 이를 수정, 보완하였다.

정보검색분야에서 방대한 양의 문서로 부터 사용자가 원하는 정보만을 검색하기 위하여 텍스트정보로부터 자동색인하고 이를 이용하여 사용자가 비정형질의를 할수 있게 하였다.

4. 웹상에서의 Z39.50 프로토콜 구현

디지탈도서관 표준프로토콜을 구현하고 이를 용이하게 사용할 수 있도록 하기 위해서는 사용자 인터페이스의 구현이 필수적인 단계이다. Z39.50의 클라이언트 사용자 인터페이스를 만드는 방식으로는 Unix의 X 윈도우나 PC의 Win95환경을 직접 이용하는 방식과 인터넷에서 광범위하게 사용되는 웹 브라우저를 이용하는 방식이 있다. 전자의 방식은 디지탈도서관 프로토콜에 적합한 인터페이스를 작성할 수 있다는 장점이 있는 반면, 사용자가 새로운 인터페이스의 사용법을 익혀야 하는 단점이 있다. 따라서 인터넷상에서 실제적으로 표준화되고 있는 사용자 인터페이스인 웹 환경에서, 디지탈도서관의 클라이언트 인터페이스 구현방안이 요구되므로, 이에 대한 가능한 구현 모델들을 살펴보고, 이를 기반으로 본 연구에서 구현한 디지탈 도서관 표준프로토콜의 구현방안을 소

개하기로 한다.[4]

4.1 세 가지 구현 모델 분석

구현 방식에 따라서 분류된 구현 가능한 세 가지 모델이 그림 8에 나타나 있다. 그림 8의 (1)은 CGI(Common Gateway Interface) 방식을 사용한 모델이며 나머지는 자바 에플릿(applet)을 사용한 모델이다. 그림 8의 (2)는 클라이언트가 Z39.50 프로토콜을 사용하는 에이전트(Agent)를 통하여 사용자 인터페이스 기능만을 가진 에플릿으로 디지탈 도서관 서비스를 받는 모델이다. 그림 8의 (3)은 Z39.50 프로토콜 기능을 가진 에플릿으로 서비스를 사용하는 클라이언트/서버형 모델로써, 이 모델을 세 가지 방법으로 구현할 수 있는데 그 중의 하나인 자바 클래스 라이브러리 방식의 그림이다.

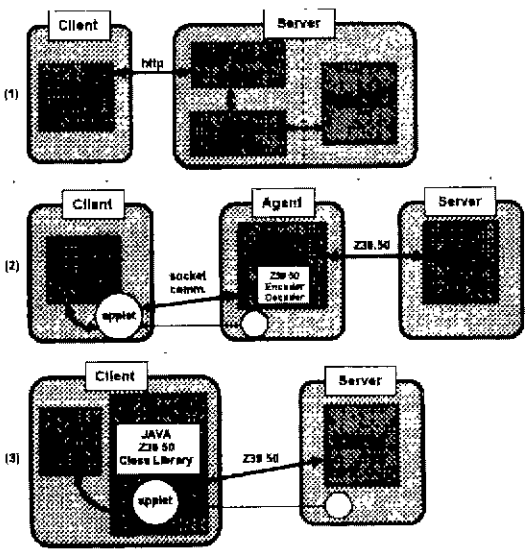

그림 8 전자 도서관 프로토콜 구현 모델들 :
(1) CGI 실행화일 방식 (2) 에이전트 방식
(3) 자바에플릿(자바클래스라이브러리)방식

가) CGI 실행화일 방식

이 모델은 웹을 통해 접근하려는 클라이언트의 요구를 CGI 실행화일 방식으로 처리하여 서비스를 제공하는 방식을 사용한다.

이 모델은 기존의 구현 모델을 큰 어려움 없이 쉽게 CGI 실행화일 방식으로 변경할 수 있는 장점이 있지만, 프로세스 교체 및 데이타 복사 등 CGI 방식 자체의 문제 때문에 성능이 좋지 않으며, 특히 Z39.50의 기능 중에는 대용량의 결과 데이타를 서버가 분할하여 전송하는 세그먼트의 기능이 있는데 이 방식에서는 그 기능을 사용할 수 없는 단점이 있다. 웹에서 디지탈 도서관을 접근하기 위한 방안으로 I-Site에서 이 방식을 사용하고 있다.

나) 에이전트 기반의 모델

이 모델은 클라이언트와 서버 중간에 에이전트를 두어서 에이전트와 서버는 Z39.50 표준 프로토콜 통신을 하며 클라이언트와 에이전트는 단순한 사용자 인터페이스 기능을 가진 자바 에플릿을 통해 통신하는 모델이다.

이것은 CGI 실행화일 방식과 마찬가지로 기존에 구현한 Z39.50 프로토콜을 큰 변경없이 에이전트와 서버사이에 직접 사용할 수가 있고, 사용자는 단지 에플릿만 다운로드 받기만 하면 즉시 서비스를 이용할 수 있는 장점도 있지만, 서버에서 제공하는 Z39.50 프로토콜 서비스 기능을 에이전트가 클라이언트에 적절하게 전달하는 데에 어려움이 있다. 예를 들어, 에이전트가 클라이언트에게 세그먼트 기능 등 여러 기능들을 제공하려면, 클라이언트와 에이전트간에는 또 다른 프로토콜이 존재해야만 한다.

다) 자바 에플릿 방식

이 모델은 Z39.50기능이 있는 자바 에플릿을 서버로부터 다운로드받아서 서버와 직접 Z39.50 프로토콜을 사용하기 때문에, 클라이언트는 서버가 제공하는 Z39.50 프로토콜 서비스 기능을 직접 전달받을 수 있는 모델이다. 그래서 대용량 및 실시간 멀티미디어 데이타의 경우에 이전의 다른 모델보다 더 효과적인 전송을 할 수가 있다.

이 모델을 세 가지 방식으로 구현할 수가 있다. 먼저, Z39.50 프로토콜 기능을 전부 자바 언어만으로 구현하는 방식, 그리고 C나 C++로 구현된 Z39.50 프로토콜 기능을 자바 라이브러리로 클라이언트에 등록시켜 놓고 호출하는 Z39.50 원시함수 호출(Native Method) 방

식, 그리고 마지막으로 Z39.50 프로토콜 기능을 자바 클래스로 구현하는 자바 클래스 라이브러리 방식이 있다.

4.2 자바 클래스 라이브러리 방식에 기반한 구현

4.1에서 제시한 구현 모델을 비교 검토한 결과, 자바 에플릿 방식이 디지탈 도서관 표준 프로토콜의 구현방안에 가장 적합하므로 이 방식을 기반으로 구현하였다. 자바 에플릿 방식 중에서 실질적인 구현방식은 자바 클래스 라이브러리 방식을 채택하였다. 전부 자바 언어로만 구현하는 방식은 ASN.1 컴파일러가 현재까지 자바언어를 지원하지 못하고 있는 등 약간의 문제가 있기 때문에 이 방식을 배제했으며, 자바 클래스로 구성하는 것이 원시함수호출 방식보다 훨씬 더 정형화된 방식이므로 이 방식을 기반으로 하여 디지탈 도서관 프로토콜 구현 모델을 구현하였다.

Z39.50 프로토콜을 구현하기 위해 필요한 구현 모델 구성이 그림 9에 나타나 있다. 클라이언트에는 Z39.50 프로토콜 기능 및 데이타를 부호화/복호화하는 기능이 정의된 자바 클래스 라이브러리가 구현되어 있고, 이것은 C로 된 자바 패키지(package)로 구현된다. 서버에는 Z39.50 프로토콜 기능 및 데이타를 부호화/복호화하는 기능이 구현되어 있으며, 또한 클라이언트에게 사용자 인터페이스로 역할할 에플릿이 저장되어 있다.

클라이언트는 WWW 브라우저상에서 이 에플릿을 다운로드 받아서 질의를 하며, 그 질의는 이 에플릿을 통해 서버로 전달되어, 결국,

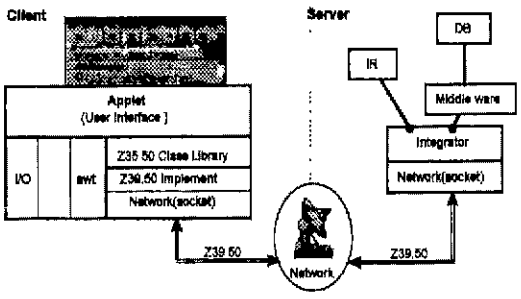

그림 9 전자 도서관 표준프로토콜 구현 모델 구성

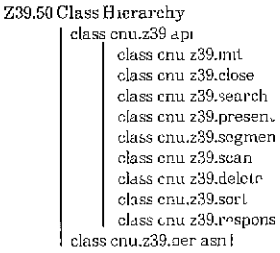

그림 10 자바 Z39.50 클래스 계승 트리

그 결과는 에플릿을 통해 클라이언트로 전달된다. 자바 Z39.50 클라스는 그림 10과 같은 트리를 갖는다.

5. 결 론

현재의 정보 사회에서 자신이 원하는 정보를 손쉽고, 빠르게 얻는 것은 매우 중요한 일이다. 이런 사용자의 요구를 만족시키고, 더 많은 서비스를 제공하기 위하여 전자 도서관의 기능은 점점 더 커질 것이다. 이에 따라 가장 많은 사용자가 이용하고 있는 웹에서 전자 도서관을 효율적으로 접근하게 하는 것은 매우 중요한 문제이다.

본고에서는 Z39.50이라는 전자 도서관 표준 프로토콜에 대한 기능들을 소개하였으며, 기존의 전자 도서관 구현사례를 살펴보았다. 또한 웹 상에서 직접 전자 도서관을 효율적으로 접근할 수 있도록 하기 위하여, 구현 가능한 구현 모델들을 제시했고, 이를 Z39.50이라는 전자 도서관 표준 프로토콜 구현 관점에서 분석하여 바람직한 모델을 제안하였다.

향후 연구 과제로써는 전자 도서관 서버가 증가함에 따라 클라이언트와 서버를 관리할 수 있는 에이전트 모델이 요구되는데, 이때 다양하고 유용한 서비스를 제공해줄 수 있는 에이전트 설계가 필요하다. 그리고, SGML(Standard Generlized Markup Language), MHEG(Multimedia and Hypermedia information coding Expert Group) 문서들 같은 여러가지 다른 문서 타입을 제공할 수 있는 다양한 문서 형식을 지원하기 위한 연구가 요구된다. 아울러, 전자 도서관 프로토콜 상에서 대용량 멀티미디어 정보를 실시간 전송하기 위한

연구가 필요하며, 초고속통신망과의 연계방안도 필요하다.

참고문헌

[1] 맹성현, 전우직 외 다수, 전자 도서관 : 분산멀티미디어 정보 서비스 시스템, 충남대학교 최종 연구개발 결과 보고서, 1996.

[2] 임유진, Isite를 이용한 전자 도서관 구현, KRNET 96, 발표자료집II, July 1996, pp. 731-741.

[3] 원종상, 안개일, 전우직, 전자 도서관 표준프로토콜 Z39.50 구현, 정보과학회 가을학술발표논문집(B), 제23권 제6호, 1996.

[4] 안개일,원종상,전우직,웹 환경에서의 전자도서관 표준프로토콜 구현방안, 정보과학회 가을학술발표논문집(B), 제23권,제6호, 1996.

[5] ANSI/NISO. Information Retrieval (Z39.50) : Application Service Definition and Protocol Specification July, 1995.

[6] Bruce Schatx. Building the Interspace : The Illinois Digital Library Project, Communications of the ACM, Vol. 38, No. 4, April 1995.

[7] Douglas D. Nebert, James Fullton, Use of the Isite Z39.50 software to search and retrieve spatially-referenced data, Proceedings of the Digital Libraries 95 Conference, June 1995.

[8] Jonathan Puray, the British Librarys Initiatives for Access Projects, Communications of the ACM, Vol. 38, No. 4, April 1995.

[9] The NIST group, Guide to Z39.50/ PRISE 2.0 Installation, Use & Modification , http : //potomac.ncsl.nist.gov/~over/zp2/main.html.

[10] Peter J. Nurnberg, Richard Furuta, John J. Leggett, Catherine C. Marshall, Frank M. Shipman III, Digital Libraries : Issues and Architectures, Proceedings of the Digital Libraries 95 Conference, June 1995.

[11] Robert Wilensky, UC Berkeleys Digital Library Project, Communications of the ACM, Vol. 38, No. 4, April 1995.

[12] The Stanford Digital Libraries Group, The Stanford Digital Library Project, Communications of the ACM, Vol. 38, No. 4, April 1995.

[13] Terence R. Smith and James Frew, Alexandria Digital Library, Communications of the ACM, Vol. 38, No. 4, April 1995.

[14] The WILLOW group, Using Willow with the Z39.50 Information Retrieval Protocol, http : //www.washington.edu/willow/z3950.html.

안 개 일

1993 충남대학교 컴퓨터공학과 (학사)
1995 충남대학교 컴퓨터공학과 (석사)
1996~현재 충남대학교 컴퓨터공학과 박사과정
관심분야 : DAVIC, 전자 도서관

전 우 직

1982 서울대학교 컴퓨터공학과 (학사)
1984 서울대학교 컴퓨터공학과 (석사)
1984~1987 한국전자통신연구소 데이터통신연구실
1989 미국 Delaware 전산학과 (석사)
1992 미국 Delaware 전산학과 (박사)
1992~1993 한국전자통신연구소 정보통신표준센타
1993.4~현재 충남대학교 컴퓨터공학과 조교수
관심분야 : 컴퓨터 네트워크, 프로토콜 공학, DAVIC, 초고속 프로토콜 구현 기법, 전자 도서관