

CORBA기능을 이용한 정보검색시스템 통합에 관한 연구

A Study on Information Retrieval Systems Integration Using Common Object Request Broker Architecture

최한석(Han-Suk Choi)*, 김상미(Sang-Mi Kim)**

남태우(Tae-Woo Nam)***, 손덕주(Deok-Joo Son)****

목 차

1. 서 론	통합 모델
2. 정보검색 특성 및 표준 프로토콜	4.1 DDIR/ORB 구조
2.1 기존의 정보검색시스템들의 특성	4.2 DDIR/ORB의 객체 모델
2.2 Z39.50 정보검색 표준 프로토콜	4.3 DDIR/ORB의 질의 처리
3. CORBA기능	5. 실험 및 검토
3.1 CORBA 특성 및 주요 인터페이스	5.1 실험 환경
3.2 CORBA 인터페이스 기능	5.2 실험 결과
3.3 CORBA기능을 이용한 Z39.50 구현 방안	5.3 검토 의견
4. DDIR/ORB : CORBA 기반의 정보 검색시스템	6. 결 론

초 록

본 논문에서는 정보검색을 원하는 이용자들에게 시스템 및 DBMS의 이형성, 서로 다른 검색시스템 사용에 관계없이 단일 사용자 인터페이스를 통해 일관성 있는 질의 및 검색결과를 제공할 수 있는 CORBA기반의 정보검색시스템(DDIR/ORB) 통합모델을 제안한다. 본 논문에서 제안한 DDIR/ORB는 질의를 요구하는 클라이언트와 검색을 실행하는 응용서버 사이에 미들웨어중개자를 두는 3-Tier 방식의 클라이언트-서버 구조이다. DDIR/ORB의 미들웨어중개자는 MARC기반의 서지 데이터베이스와 CD-ROM 텍스트 데이터베이스에 대한 접근 투명성을 보장하고, 정보검색 결과에 대한 자유로운 데이터 교환 및 변환을 제공하며, 기존의 정보검색시스템의 재사용을 보장한다. DDIR/ORB는 시스템 설계 및 구현에서 OMG IDL을 사용함으로써 인터페이스 복잡도가 감소되었고 구성요소들의 구현 비용을 최소화하였다.

ABSTRACT

This study proposes an integration model of information retrieval systems using a standard distributed object computing technology in digital library environments. In the proposed integration model called DDIR/ORB, the middle-ware broker based on CORBA is designed for the transparent access to the distributed information repositories and the consistent view of the information retrieval by applying Z39.50 protocol. The DDIR/ORB is an adaptable open architecture that allows for the following benefits : bibliographic and abstract information retrieval simultaneously, interoperability between application servers and clients, consistent view of search results, complexity reduction of integration interfaces, and easy to use.

* 키워드 : 디지털도서관, 정보검색, CORBA, 분산객체컴퓨팅, Z39.50, 시스템 통합

* 목포대학교 전산통계학과 부교수

** 광주대학교 문헌정보학과 강사

*** 중앙대학교 문헌정보학과 교수

**** ETRI 분산처리연구실 책임연구원

■ 논문접수일 : 1996년 11월 13일

1. 서 론

최근의 인터넷 활용 및 디지털 도서관 등장은 전통적인 도서관에서의 정보검색 기능에 커다란 변화를 요구하고 있다. 전통적인 도서관에서 이용자는 주로 단순 서지정보 및 문헌들의 위치 정보를 검색하였다. 그러나 현재 대부분의 이용자는 원하는 정보를 시간과 공간을 초월하여 다양한 형태의 정보 서비스를 제공받고자 한다. 이러한 이용자의 요구 변화를 수용하고, 분산컴퓨팅 기술, 정보통신 기술, 전자출판 기술을 통합하여 새로운 형태의 정보서비스 응용으로 나타난 것이 오늘날의 디지털도서관이다. 그러나 분산환경에서 클라이언트-서버(Client-Server)구조를 기반으로 하는 디지털도서관의 가장 큰 문제는 컴퓨터 하드웨어의 이형성(Heterogeneity), 서로 다른 플랫폼(Platform) 사용, 이형의 DBMS, 서로 다른 탐색 엔진 및 정보검색시스템 사용, 네트워크 전송 프로토콜의 다양성, 다양한 형식의 데이터 표현 방법 등이다.

클라이언트-서버 기반의 분산 환경에서 시스템 통합시 발생하는 시스템들의 이형성, 분산성, 상호운용성(Interoperability) 문제를 미들웨어(Middleware) 기술을 사용하여 해결하려는 연구가 현재 활발히 진행중이다. Wiederhold(1994; 1995; 1996)는 중재자(Mediator)를 사용하여 분산된 자원들로부터 특정 응용에 필요한 지식을 획득하는 방법, 응용들간의 상호운용성 제공방법, 디지털도서관 구현 방법, 분산 환경에서의 시간적 데이터베이스(Temporal Database)들의 통합 방법을 제안하였다. 분산 환경에서 클라이언트-

서버 기반의 응용들을 좀더 쉽게 개발하기 위한 미들웨어 도구 및 서비스 구현 모델로는 OSF(Open System Foundation)의 DCE(Distributed Computing Environment)와 OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture)가 있다. OSF의 DCE는 분산 응용들을 효과적으로 개발하기 위해서 여러 가지 개발환경과 분산서비스들을 제공하고 OMG의 CORBA는 응용 객체들의 요구를 객체구현 서버에 투명(Transparent)하게 전달하여 구현 결과를 위치 및 네트워크 전송 프로토콜과는 독립되게 전송하는 통신 하부구조이다(OMG 1993).

본 연구의 목적은 미들웨어 기반기술을 활용하여, 디지털도서관 환경에서 이용자들에게 하드웨어, 운영체제, DBMS, 검색시스템들의 이형성에 관계없이 단일의 사용자 인터페이스를 통해 일관성있는 질의 및 검색결과를 제공할 수 있는 CORBA 기반의 정보검색시스템 통합모델을 제안하고 구현 및 실험을 통해서 시스템 특성을 평가하는 것이다. 본 연구에서는 정보검색시스템 통합을 구현하기 위해서 실버플래터(Silver Platter)사의 정보검색시스템인 SPIRS(Silver Platter Information Retrieval System) 검색 S/W와 목포대학교 서지정보 검색시스템인 MNUIRS를 시스템 통합에 활용한다. SPIRS는 클라이언트-서버 환경에서 대용량의 자료들을 안정적으로 저장한 CD-ROM 데이터베이스들을 효과적으로 탐색 또는 검색할 수 있는 검색도구이다. SPIRS는 WinSPIRS, UNIX-SPIRS, PC-SPIRS, MacSPIRS, WebSPIRS, SPIRS-

FT(Fulltext) 등의 다양한 플랫폼들을 지원할 수 있는 검색도구로서 자동주제탐색, 동시다중DB탐색, 불리언 및 근사탐색, 와일드카드 탐색, 인덱스로부터 자동탐색, 시소러스로부터 통제 어휘 항목들의 직접선택, 하이퍼텍스트 탐색 기능 등을 제공한다(Silver Platter 1995, 8). MNUIRS는 MARC(MACHine Re-able Catalog)기반의 서지정보 검색시스템으로 저자, 서명, 주제명, 출판년 등 문헌의 특성을 나타내는 단순 서지정보 등을 검색한다.

본 연구에서 제안한 CORBA기반의 정보검색시스템(Distributed Document Information Retrieval Based on CORBA : DDIR/ORB) 통합모델은 질의를 요구하는 클라이언트와 검색을 실행하는 응용서버 사이에 미들웨어중개자(Middleware Broker)를 두는 3-Tier 방식의 클라이언트-서버 구조이다. 본 연구에서의 미들웨어중개자는 MARC기반의 서지 데이터베이스와 CD-ROM 텍스트 데이터베이스를 사용자 입장에서는 투명적으로 접근할 수 있음을 보장하고, 정보검색 결과에 대한 자유로운 데이터 교환 및 변환을 제공하며, 기존의 정보검색시스템의 재사용을 보장한다. DDIR/ORB의 미들웨어중개자는 지리적으로 분산되어 있는 정보검색 응용 서버들의 위치 정보를 제공하는 로케이션 서비스(Location Service), 각 정보검색 응용 서버들의 스키마 정보 및 전송될 데이터의 형식 정보를 제공하는 메타데이터 서비스(Metadata Service), 클라이언트 및 응용서버들이 원하는 데이터 형식으로 변환하는 변환 서비스(Conversion Service), 검색 결과의 생성 및

전송에 필요한 메모리 공간을 제공하는 데이터 객체(Data Objects), 데이터 객체들의 생성, 삭제, 복사 등을 처리하는 팩토리(Factory)로 구성된다.

또한, 본 연구에서는 이형의 분산환경에서 다양한 문헌 DB들의 서지정보를 탐색할 수 있게 하고 사용자에게 일관성 있는 검색 결과를 제공하기 위해서 표준기반의 분산객체컴퓨팅 기술을 이용하여 Z39.50 프로토콜 구현 방법을 제시한다. 본 연구에서 제안하는 CORBA기반의 정보검색시스템 통합모델은 프레임워크기반의 하이브리드구조로 설계되고 OMG의 인터페이스정의언어(Interface Definition Language : IDL)로 필요한 연산들을 정의함으로써 정보검색시스템 통합에 필요한 인터페이스 복잡도를 감소 시키고 시스템 구성요소들의 설계 및 구현 비용을 최소화한다.

본 논문의 구성은 제 2장에서 통합하고자 하는 기존의 정보검색시스템 특징을 논의하고 정보검색 표준 프로토콜인 Z39.50의 발전 과정 및 특성을 소개한다. 제 3장에서는 분산객체 컴퓨팅 환경의 공통 통신채널인 CORBA기능 및 인터페이스 구조를 논의하고 CORBA 기반의 Z39.50 표준 프로토콜 구현 방안을 제안한다. 제 4장에서는 CORBA기반의 정보검색시스템 통합 모델을 제안하고 미들웨어중개자의 주요 서비스들 및 객체 모델, 질의처리 과정을 논의한다. 제 5장은 본 연구에서 제안한 정보검색 통합 모델의 구현 및 실험 결과를 논의하고 통합 시스템의 특성을 평가한다. 제 6장에서는 본 논문의 결론과 향후 연구 방향을 기술한다.

2. 정보검색 특성 및 표준 프로토콜

2.1 기존의 정보검색시스템들의 특성

현재 대부분의 도서관에서는 문헌들의 서지적 정보를 체계적으로 정리하여 제공하는 방법으로 기계 가독형 목록인 MARC (MACHine Readable Catalog)를 사용하고 있다. MARC는 도서관에서 상호 정보 공유를 위한 문헌의 서지정보를 전달하는 표준 형식이다. MARC의 모든 레코드는 고정장 필드와 가변장 필드로 구성되는데 리더와 디렉토리는 고정장 필드, 제어 필드와 데이터 필드는 가변장 필드이다. 리더는 레코드 처리에 필요한 정보를 갖고, 디렉토리는 레코드에 출현하는 가변장 필드의 정보를 갖고 있다. 제어 필드는 서지 데이터의 처리에 필요한 정보를 의미하고 데이터 필드는 서지 데이터 자체를 말한다(국립중앙 1993).

MARC기반의 정보검색시스템은 테이블 형태로 데이터를 관리하기 때문에 사용하기 편하고 유연성이 좋다. 특히, 정형화된 문자나 숫자를 처리하는 데는 검색 성능이 뛰어나다. 그러나 MARC기반의 정보검색시스템은 다음과 같은 몇 가지 문제점을 가지고 있다. 첫째, 하이퍼텍스트 및 멀티미디어를 포함한 다양한 속성을 지닌 데이터 모델을 처리하는 데 많은 한계가 있다. 둘째, 서명, 저자명, 출판사항 등 문헌의 특성을 나타내는 단순한 서지정보만 제공하고 문헌의 초록 및 전문(Full-text) 등 이용자들이 요구하는 다양한 정보를 충족시키지 못한다. 셋째, MARC 레코드에는 실제로 쓰이지 않는 데이터 요소 및 이산적

요소들이 존재한다. 넷째, 네트워크 상에 분산되어 있는 이질적인 정보를 이용자가 원하는 장소에서 원하는 정보를 자유롭게 이용할 수 있는 전문 데이터베이스 구축이 어렵다. 다섯째, 문헌의 구조적 표현 및 각 항목간의 관계를 효과적으로 나타낼 수 없다. 그러나 현재 대부분의 전통적 도서관에서는 MARC기반의 정보검색시스템을 사용하고 있다. 따라서 본 논문에서는 이러한 정보검색 S/W들을 재사용할 수 있는 방법을 제시한다.

정영미(1996, 20)는 Gale Directory of Databases 1995년 1월호에 실린 데이터베이스를 정보 유형별로 분석한 결과를 보면 텍스트 데이터베이스가 전체의 72%를 차지하고 있으며, 이를 세분하면 서지정보가 26%, 전문 정보가 49%, 디렉토리 정보가 23%, 특허 및 상표 정보가 1%, 사전 정보가 1%로 나타나 있다고 하였다. 특히, 주제 색인어를 통해 정보의 내용을 적절히 표현할 수 있는 서지 및 초록 DB와 전문 DB는 전체 DB의 60%나 되며, 특히 텍스트 DB의 정보검색시스템은 텍스트의 내용을 요약하여 표현하는 색인어를 어떻게 선택하느냐에 따라 검색 효율이 결정된다고 하였다.

또한 Fidel(1991)의 연구에서는 탐색어 선택에 영향을 미치는 요인들을 정보요구 관련 특성(높은 재현율/정확률), 데이터베이스 관련 특성(시소러스의 유무, 복수 데이터베이스 탐색의 필요성), 탐색자 관련 특성(개인적 성향)의 세 가지 범주로 나누어 조사하였는데, 데이터베이스 관련 특성이 가장 큰 영향을 미치는 것으로 나타났다. 이 연구가 제시하는 중요한 결론은 특정한 데이터베이스와 관련된

색인어의 질이 탐색어 선택에 가장 큰 영향을 미친다는 것이다. 즉, 탐색자들은 시소러스와 통제언어 색인의 질이 만족스러울 때에는 통제언어를 탐색어로 사용하며, 복수 데이터베이스의 탐색이 필요한 정보 요구에 대해서는 자연어 탐색을 선호한다는 것이다. 따라서 쉽게 사용할 수 있는 고품질의 시소러스와 복수 데이터베이스 탐색시 도움이 되는 언어변환 장치의 개발이 매우 중요함을 지적하고 있다(정영미 1996). 본 연구에서는 전세계적으로 가장 많이 사용되고 있는 텍스트 데이터베이스 정보검색시스템인 실버플래터사의 SPIRS 검색 S/W를 시스템 통합에 활용한다.

2.2 Z39.50 정보검색 표준 프로토콜

컴퓨터 및 네트워크 발전으로 지리적으로 분산되어 있는 다양한 정보자원들이 분산 컴퓨팅 기술을 이용하여 쉽게 공유할 수 있게 되었다. 그러나 서로 다른 시스템 사이의 이형성 및 데이터 교환을 위한 상호운용성 문제를 효과적으로 해결하기 위해서 표준화된 정보검색 프로토콜이 요구되었다. 이에 따라 미국 국가표준기구(NISO)는 1979년에 정보검색용 프로토콜에 관한 연구를 담당할 표준화 위원회를 발족시킴으로써 이 문제에 정면 대응하기 시작했다. 이 위원회의 연구 결과, Z39.50-1988이 처음으로 만들어졌는데 이 버전은 현재 버전과 상당한 차이가 나지만 정보검색 도구 개발의 기초가 되었다. 1988년에 표준으로 승인된 이후에 Z39.50 개발자들은 Z39.50 표준 프로토콜의 이용을 촉진시키는 작업이 시작되었으며, NISO는 1992년에

Z39.50의 개정을 실시하였고 시스템 개발자들이 Z39.50 표준을 계속 발전시킴에 따라 1995년에 버전 3 개정판이 만들어졌다(안현수 1996, 19).

이와는 별도로 정보검색 표준에 관한 국제 표준화기구(ISO)의 노력 결과, 1991년에 Z39.50과 호환이 되는 두 가지의 국제 표준이 완성되었는데 이들은 ISO 10162(Search and Retrieval Application Service Definition)와 ISO 10163(Search and Retrieval Protocol Specification)이다. ANSI/NISO의 Z39.50-1992는 미국 표준과 국제 표준을 모두 수용함으로써 미국 표준이 국제 표준과 호환성을 갖도록 하였다. 이에 따라 Z39.50을 기반으로 만들어진 정보검색시스템은 ISO의 10162/10163을 기반으로 검색시스템과 상호운용성을 갖는다.

Z39.50은 이형의 클라이언트-서버 환경에서 질의요구 및 결과를 주고받는 표준화된 정보검색 프로토콜로서 정보검색 절차와 방법이 표준화가 되기 때문에 분산환경에서 대규모 문헌 데이터베이스를 쉽게 활용할 수 있다. Z39.50의 장점은 클라이언트측의 이용자 인터페이스를 정보 서버, 검색 엔진, 그리고 데이터베이스와 분리시키는 데 있다. 즉, Z39.50을 이용할 경우 다양한 정보자원의 정보를 일관되게 탐색할 수 있도록 해주며 클라이언트로 하여금 여러 데이터베이스와 서버의 정보들을 통합할 수 있도록 해준다.

Z39.50의 일반적인 특징을 요약하면 다음과 같다. 첫째, Z39.50은 두 대의 컴퓨터가 서로 상호 작용하는 형식과 절차들을 다루는 네트워크 규약이다. 이 프로토콜은 세션지향

적(Session-oriented)이며 연속적이다. 둘째, Z39.50은 네트워크 정보검색용 표준 프로토콜이다. 이용자는 Z39.50을 이용하여 특정 레코드들을 식별할 수 있는 기준을 설정하고 식별된 레코드의 일부 혹은 전부를 전송하도록 요구함으로써 원격 데이터베이스 레코드에 접속할 수가 있다. 셋째, Z39.50은 클라이언트-서버 모형을 기본으로 하고 있다. 일반적으로 클라이언트 프로그램은 이용자가 사용할 수 있는 인터페이스를 제공하고 이용자의 질의를 프로토콜이 요구하는 형식으로 변환시키고 검색 결과를 이용자에게 제공한다. 서버 프로그램은 클라이언트 프로그램이 전송한 질의를 기다리고 있다가 이들 질의가 도착하면 처리를 하고 그 결과를 클라이언트 프로그램에 보내 주는 기능을 한다. 넷째, 현재 Z39.50 프로토콜을 구현한 사례가 급격히 늘어나고 있다. 이들 개발사례를 유형별로 살펴보면 Geac, NOTIS, VTLS와 같은 도서관자동화 시스템 개발 업체, CARL, OCLC, RLG와 같은 공동편목서비스 제공 기관, DIALOG와 같은 상용 데이터베이스 서비스 업체 등이 있다(안개일, 1996; 원종상, 1996). 따라서 Z39.50 프로토콜은 클라이언트-서버 모델을 기반으로 하는 정보검색 S/W의 필수적인 구현 모듈이다.

3. CORBA기능

3.1 CORBA 특성 및 주요 인터페이스

분산 환경에서 시스템간의 이형성 및 분산

성을 투명하게 해결하고 시스템 및 응용 서버들간의 상호 운용 기술을 제공하려는 연구가 객체지향 기술에 근거한 표준 기반의 분산 컴퓨팅 기반 구조이다. OMG는 개방 분산 컴퓨팅 환경에서 독립적으로 개발된 응용 사이의 상호운용성 제공을 위한 표준 인터페이스 규격을 마련하기 위하여 1989년 국제적 산업체 컨소시엄으로 설립되었다.

OMG 참여자들의 주요 표준화 작업 결과는 1990년 9월에 객체 관리 구조(Object Management Architecture : OMA) 가이드 버전 1.0이 발표되었고, 1991년 12월 OMA 기본 구조 중에서 공통객체요구중개자(CORBA)의 기본 인터페이스 규격이 정의되었으며, 1993년에는 CORBA 1.1의 문제점을 개선하여 CORBA 1.2를 발표하였다. 1995년에는 ORB간의 상호 운용성, 인터페이스 저장소, C++과 Smalltalk 언어와의 바인딩을 정의한 CORBA 2.0이 발표되었다(OMG 1992). 1994년 3월에는 객체들 간의 상호작용을 위한 공통객체서비스규격(Common Object Service Specification : COSS) Vol. I이 발표되었고, 1995년 8월에 객체 서비스 기능들이 추가된 COSS Vol. II가 발표되었다(OMG 1994).

OMA는 OMG의 표준 규격을 따르는 모든 소프트웨어 구성요소들을 재사용할 수 있는 객체 모델을 정의하고 분산 객체들의 효과적 통합 방법을 제공한다. OMA의 중심적 구성 요소는 객체요구중개자(Object Request Broker : ORB), 객체서비스, 공통이용서비스, 응용객체들로 구성된다. ORB는 이형 분산 환경에서 객체들의 요구들에 투명하게 반

응하는 통신기반 구조이다. 이러한 ORB의 표준화된 공통 규격을 정의한 것이 CORBA이다. 객체서비스는 이름 서비스, 생명주기 서비스, 사건 서비스, 지속 서비스 등 객체 생성 및 사건 처리 같은 저수준의 객체서비스들의 모임으로서, 객체 사용 및 구현을 위한 기본 기능을 지원한다. 공통이용서비스는 출력 또는 전자 메일, 사용자 인터페이스 같은 서비스들의 모임으로서, 여러 가지 특정 응용들이 일반적으로 활용할 수 있는 서비스들이다. 응용객체들은 개발자들의 응용 프로그램, 상용 소프트웨어 및 정형화된 시스템등이 포함된 최종 사용자들의 응용 S/W를 말한다.

CORBA 2.0까지 정의된 주요 인터페이스들은 IDL, ORB 인터페이스, 인터페이스저장소(Interface Repository : IR), 동적기동인터페이스(Dynamic Invocation Interface : DII), 기본객체어댑터(Basic Object Adapter : BOA), IDL과 프로그래밍 언어와의 사상(C, C++, Smalltalk), ORB간 프로토콜(Inter-ORBs Protocols : IOPS) 등이다(Choi 1996).

3.2 CORBA 인터페이스 기능

CORBA의 기능은 객체들의 요구를 생성하고 요구에 대한 구현 결과를 객체들의 위치 및 네트워크 전송 프로토콜과 관계없이 투명하게 전달하는 것이다. 즉, CORBA는 분산 환경에서 서로 다른 시스템 간의 상호운용성 및 응용 객체들의 상호 연동을 무리 없이 제공한다. CORBA는 이러한 목적 달성을 위하여 객체들의 클래스 정의 및 공통 연산 정의

를 OMG IDL을 이용하여 정의한다. IDL은 객체들의 인터페이스들을 정의하는 데 사용되는 일종의 메타언어이다. OMG IDL의 기본형은 다음과 같다.

```
interface <obj-name> {
    [op-attribute]<op_type_spec> <identifier>
    (param1, ..., paramN)
    [raises( except1, ... exceptM)]
    [context( name1, ... nameN)]
};
```

IDL의 연산 자료형은 기본형, 구조체, 스트링, Sequence, 다차원 배열을 지원하고 raise문은 예외 처리시 사용되며 컨텍스트문은 객체에 의해서 요구되는 실행 결과를 출력 파라미터를 사용하지 않고 사용자 컨텍스트에 전달하고자 할 때 사용된다. IDL은 다른 객체 지향 언어에서 제공하는 다중 상속 특성을 지원하지 않는다. 다음은 ex3 객체가 ex1을 상속받고 예외를 처리하는 IDL문이다.

```
interface ex3 : ex1 {
    exception BadCall { string<80> reason ; }
    oneway void op1( in long p1 ) ;
    ex1 op2( in long arg1, out long arg2,
    inout arg3) ;
    raises( BadCall ) ;
};
```

IDL은 특정 프로그래밍 언어와 독립되게 객체들의 인터페이스들을 정의한다. 따라서 IDL로 기술된 화일을 IDL 컴파일러로 컴파

일하면 원하는 프로그래밍 언어(C, C++)로 사상된다. 이때 생성되는 것이 클라이언트 IDL 스텝(Stub)과 서버 스켈리튼(Skeleton)으로서 ORB의 정적 인터페이스 기능을 제공한다(IONA 1995).

CORBA의 주요 구성요소 및 ORB 인터페이스 구조는 다음 <그림 1>과 같다. 클라이언트 객체들과 객체구현과의 통신은 ORB를 통해서 이루어지고 ORB는 정적 및 동적 인터페이스 호출 방법을 이용하여 통신한다.

ORB는 클라이언트와 서버 객체들 간의 상호작용을 관리하고, 객체들의 위치, 클라이언트 요구와 서버 반응에 대한 전송 기능을 수행한다. 객체들의 모든 활동이 요구자에 대하여 은폐되어 있으므로 ORB 기능의 대부분은 객체어댑터, 스텝, 스켈리튼, 동적기동 인터페이스를 통해서 이루어진다.

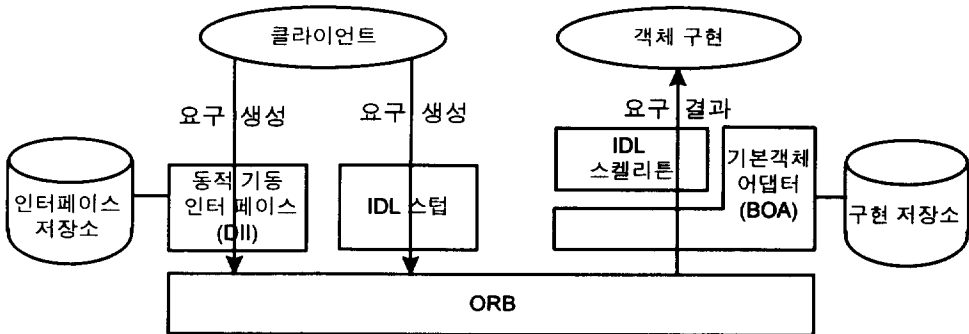
CORBA의 동적기동인터페이스는 클라이언트가 런타임에 동적으로 객체들을 생성하고 요구할 수 있는 방법이다. 동적 기동 순서는 클라이언트가 먼저 필요한 객체 참조를 바인

딩하고 요구를 생성한다. 동적기동에 필요한 연산 요구는 객체 참조, 연산명, 인자리스트로 구성되고 각 인자는 인자명, 자료형, 인자 모드(in, out, inout)로 구성된다. 클라이언트의 요구에 대한, 인자 리스트 목록과 대응되는 값들은 인터페이스 저장소로부터 얻어진다.

인터페이스 저장소는 IDL로 정의된 객체들의 연산명, 속성명, 자료형에 관한 모든 정보를 저장하고 런타임에 요구된 인터페이스 정보들을 제공한다. 인터페이스 저장소는 ModuleDef 객체, InterfaceDef, ConstantDef, TypeDef, AttributeDef, OperationDef, ExccertionDef 및 기타 필요한 인터페이스들이 계층 구조로 구성되어 있다(최한석 1995).

3.3 CORBA기능을 이용한 Z39.50 구현 방안

본 장의 3.1절 및 3.2절에서 논의하였듯이 CORBA의 핵심 기능은 이형 분산 환경에서



<그림 1> ORB 인터페이스의 요구 기동

클라이언트들의 요구를 생성하고, 요구에 대한 응용 서버들의 실행 결과를 클라이언트 및 응용 서버들의 위치 및 네트워크 전송 프로토콜에 관계없이 투명하게 전달하는 것이다. 본 연구에서는 이러한 CORBA기능을 정보검색 표준 프로토콜인 Z39.50 구현에 적용한다.

컴퓨터 시스템, 검색 엔진, 데이터베이스 등이 서로 다른 경우에도 클라이언트와 서버가 상호 통신하고 작용할 수 있는 방법을 표준화하고 있는 Z39.50은 원래 국제표준화기구에서 개발한 개방형 시스템상호접속(OSI)의 기본 참조 모형(Basic Reference Model)에서 응용 계층(Applications Layer)의 프로토콜에 해당된다. 응용 계층 프로토콜은 특정 작업을 수행하는 클라이언트와 서버에 설치되어 있는 응용 S/W들의 통신 요구사항을 지원하면서 이들 응용들과 직접 상호 작용하기도 한다. 응용 계층 프로토콜로서의 하부 구조에 존재하는 Z39.50은 네트워크 전송 프로토콜에 거의 독립적으로 동작한다. 즉, Z39.50은 현재 인터넷 환경하의 TCP/IP 전송 서비스 계층 위에서 광범위하게 이용되고 있다(안현수 1996, 20). Z39.50 프로토콜을 사용하는 클라이언트와 서버 사이의 통신 과정을 요약하면 다음과 같다.

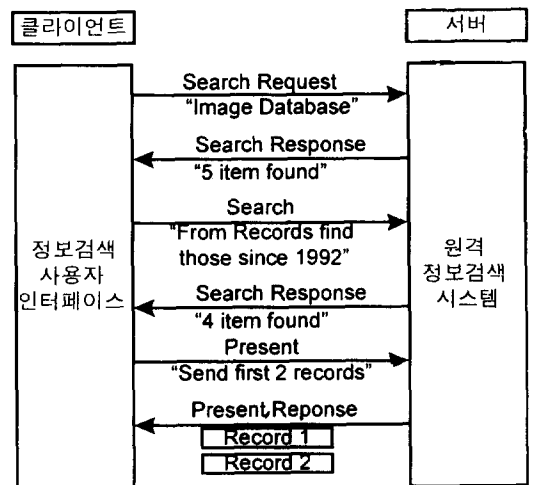
첫째, 클라이언트는 초기화 요구(Initialization Request)를 통해 특정 서버와 Z39.50 세션을 시작하고 서버는 초기화 응답(Initialization Response)을 통해 대응하게 된다. 이 과정에서 서버에서 클라이언트로 전송되는 레코드들의 최대 크기, 지원되는 프로토콜의 버전 등 앞으로 발생할 동작들에 대한 정보를 서로 교환하게 된다.

둘째, 이러한 초기화 과정이 끝난 후, 클라이언트는 서버에 질의를 보내게 된다. Z39.50 클라이언트는 질의를 표준화된 표현으로 변환시킨 후 이 표현을 Z39.50 서버에게 넘겨 준다.

셋째, 클라이언트는 검색 결과 집합으로부터 레코드들을 요구하거나 혹은 서버에게 검색 결과 집합에 대해 추가적인 처리를 하도록 요청할 수도 있다. 레코드들을 접수한 후 클라이언트는 이들 레코드들을 처리하고 이들을 이용자에게 디스플레이할 수 있다.

다음 <그림 2>는 이러한 과정을 요약적으로 보여준다. Z39.50에서 제공되는 주요 기능들 및 CORBA 인터페이스 정의를 요약하면 <표 1>과 같다. <표 1>의 Z39.50 주요 인터페이스들은 안개일(1996)에 정의된 기능들을 IDL로 정의한 것이다.

본 연구에서는 정의된 Z39.50기능 중 초기화, 탐색, 검색, 종료등 일부 기능만 C++언어



<그림 2> Z39.50의 클라이언트 서버 통신 과정

로 구현된다. 설명 기능 및 확장 서비스 기능은 미들웨어중개자에서 독립된 분산 서비스로 제공되도록 하였다.

4. DDIR/ORB : CORBA기반의 정보검색시스템 통합 모델

4.1 DDIR/ORB 구조

본 연구에서는 MARC기반의 서지정보검색 시스템과 텍스트 데이터베이스 정보검색시스템을 효과적으로 통합할 수 있는 CORBA기

반의 분산 문헌 정보검색시스템 (DDIR/ORB) 통합 모델을 제안한다. DDIR/ORB는 3-Tier 방식의 기존 클라이언트-서버 모형을 변형하여 중간에 미들웨어중개자를 통해서 클라이언트 검색 요구를 서비스하는 구조이다 (Choi 1996 ; Mowbray 1995). 이 구조의 특징은 클라이언트에게 하나의 일관성 있는 사용자 인터페이스를 통해 다양한 문헌 DB 및 이형의 시스템 자원들의 접근에 대한 투명성을 제공하는 것이다. 즉, 클라이언트는 미들웨어중개자의 구성요소인 로케이션 서비스, 메타데이터 서비스, 변환 서비스 등의 분산 서비스들을 제공받고 CORBA기반의 Z39.50 게이트웨이를 통해서 다른 응용 서버들과 통

〈표 1〉 Z39.50의 주요 서비스 기능에 대한 CORBA 인터페이스 정의

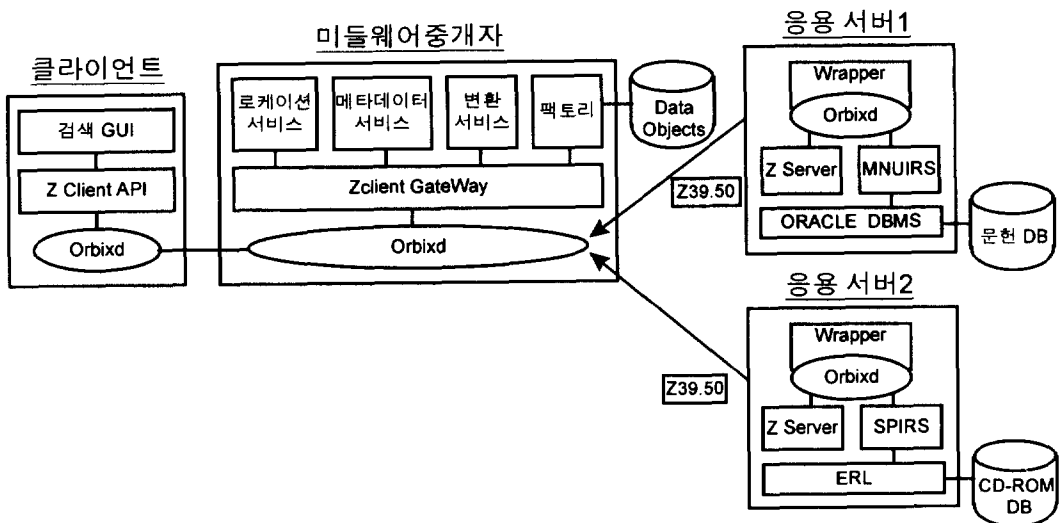
기능	서비스	CORBA 인터페이스 정의
초기화 기능	init 서비스	bind(), impl_is_ready()
탐색 기능	search 서비스	Zsearch(in string DBname, in string query, out object resultset)
접근 제어 기능	access 서비스	Zaccessstatus(in string userid, out boolean status)
검색 기능	present 서비스	Zpresent(in object resultset, in string rename, out int briefordetail)
	segmentation 서비스	Zsegment(in object resultset, in int segrec, out string Segrec, out int briefordetail)
결과 삭제 기능	delete 서비스	Zdelete(in object result, out object deletedset)
브라우징 기능	scan 서비스	Zscan(inout object indexlist)
정렬 기능	sort 서비스	Zsort(in object resultset, out object sortresult)
자원 제어 기능	resource_control, trigger_resource_control, resource_report 서비스	Zresource(inout string utilityinfo) Ztrigger(inout string triggerresource) Zreport(inout string costreport)
설명 기능	explain 서비스	ZdescribeMeta(in string objname, out object objref)
확장 서비스 기능	extended_service 서비스	bindName(in string objname, out Object objref)
종료 기능	termination 서비스	release()

신한다. 이러한 DDIR/ORB의 시스템 구성도는 <그림 3>과 같다.

<그림 3>에서 미들웨어 중개자의 각 구성요소 및 기능을 다음과 같다. DDIR/ORB에서 클라이언트는 검색 인터페이스와 Z 클라이언트 통신 인터페이스로 구성되고 Orbixd라는 공통 통신 채널을 통해 응용 서버 및 미들웨어 중개자와 통신한다. 클라이언트는 정보검색을 원하는 일반 사용자로서 그래픽 사용자 인터페이스를 통해 서명별, 저자별, 주제별, 문헌 정보검색을 요구할 수 있다. 응용 서버들은 각각의 지역 DBMS에 의해서 문헌 데이터베이스들이 생성되고 관리된다. 또한, 각 지역의 데이터베이스들은 고유의 정보검색 S/W들에 의해서 이용자 질의를 탐색하고 결과를 반환한다. Z 서버는 클라이언트의 검색 요구 및 결과를 미들웨어중개자의 Z 클라이언

트 게이트웨이를 통해 전달한다. 본 연구에서는 관계형 데이터 모델을 사용하는 오라클 DBMS 기반의 목포대학교 정보검색시스템(MNUIRS)에 대한 객체 Wrapping 인터페이스와 실버플래터 회사에서 제공하는 SPIRS 정보검색시스템에 대한 객체 Wrapping 인터페이스를 구현하여 각각의 검색 연산들을 기동 시킨다(Katz, 1993).

미들웨어중개자의 로케이션 서비스 및 메타데이터 서비스는 Z39.50의 버전 3에 추가된 설명 기능을 두 가지로 구분하여 설계한 분산 서비스 구조이다. 로케이션 서비스는 특정 응용 서버에 관한 호스트 및 객체구현 정보를 가지고 있어 클라이언트의 초기세션 설정시, 응용 서버의 객체가 바인딩되고, 바인딩된 객체 참조를 가지고 클라이언트는 검색 질의를 요구한다. 메타데이터 서비스는 이용 가능한



<그림 3> DDIR/ORB의 시스템 구성도

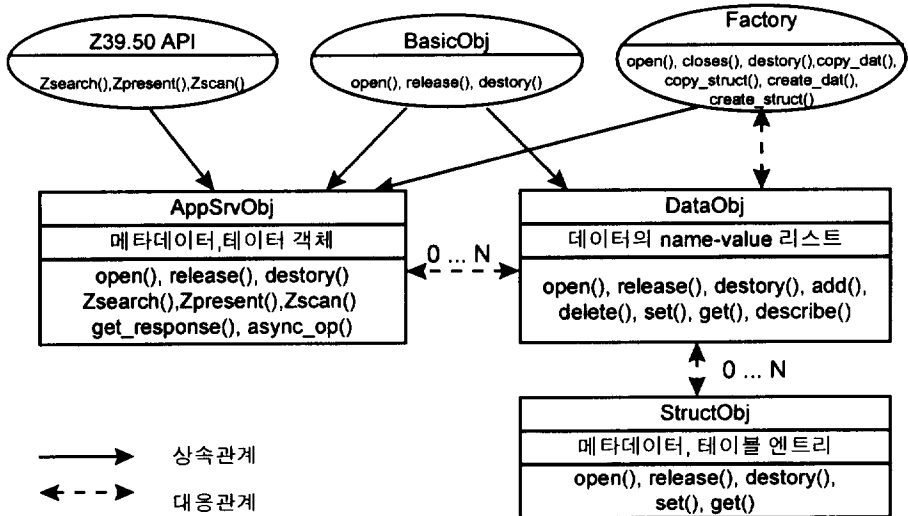
데이터베이스들의 스키마 정보, 지원되는 데이터들의 속성, 레코드 구문 구조, 구성요소들의 사양 정보등이 기록된다. 변환 서비스는 검색 질의 표준 형식으로 변환하거나 검색 결과로 생성되는 출력 형식을 클라이언트가 요구하는 형식으로 반환한다. 또한, 스키마에 대한 서로 다른 시맨틱(Semantic)을 동일한 시맨틱 구조로 변환하는 서비스이다. 즉, 서로 다른 정보검색시스템의 실행결과로 변환되는 정보를 표준 또는 일관성 있는 출력 형식으로 변환하여 클라이언트에게는 서지정보 및 초록 정보를 제공한다.

팩토리는 생성되는 데이터 객체들에 대한 생명 주기를 제어하는 서버이다. 즉, 팩토리 객체는 데이터 객체들의 생성, 복사, 제거 등에 관한 연산을 수행한다. 데이터 객체는 데이터 교환에 필요한 임시 저장소로서 검색 요

구와 결과 전달시 필요한 자료구조이다. 데이터 객체는 데이터명 및 값의 단순 쌍으로 구성된 단순 데이터 객체 구조와 복잡한 레코드 구조를 처리할 수 있는 구조적 데이터 객체 구조로 구분된다.

4.2 DDIR/ORB 의 객체 모델

본 연구에서 제안한 DDIR/ORB 구성요소들의 객체 모델은 OMG IDL을 이용하여 정의함으로써 인터페이스 복잡도가 감소된다. 즉, 기존의 MARC기반 정보검색시스템과 텍스트 데이터베이스 정보검색시스템의 통합 비용을 최소화하기 위해 클라이언트와 응용 서버간의 공통 인터페이스들을 기본 객체로 정의하고 다른 객체 서비스들이 연산들을 상속 받을 수 있게 한다.



〈그림 4〉 DDIR/ORB 객체들의 클래스 계층 관계

다음 <그림 4>는 애플리케이션 객체, 데이터 객체, 팩토리 객체간의 종속 관계 및 대응 관계를 보여준다. 각 응용 서버는 데이터 객체를 하나도 참조하지 않을 수도 있고, 매우 많이 참조할 수 있다. 같은 의미로 단순 데이터 객체는 구조적 데이터 객체를 검색되는 레코드 수에 따라서 많이 참조할 수도 있고 전혀 참조하지 않을 수 있다. 따라서 특정 검색 질의에 대해서 데이터 객체들의 생성 숫자 및 검색 결과의 레코드 수는 다르게 나타난다.

4.3 DDIR/ORB의 질의 처리

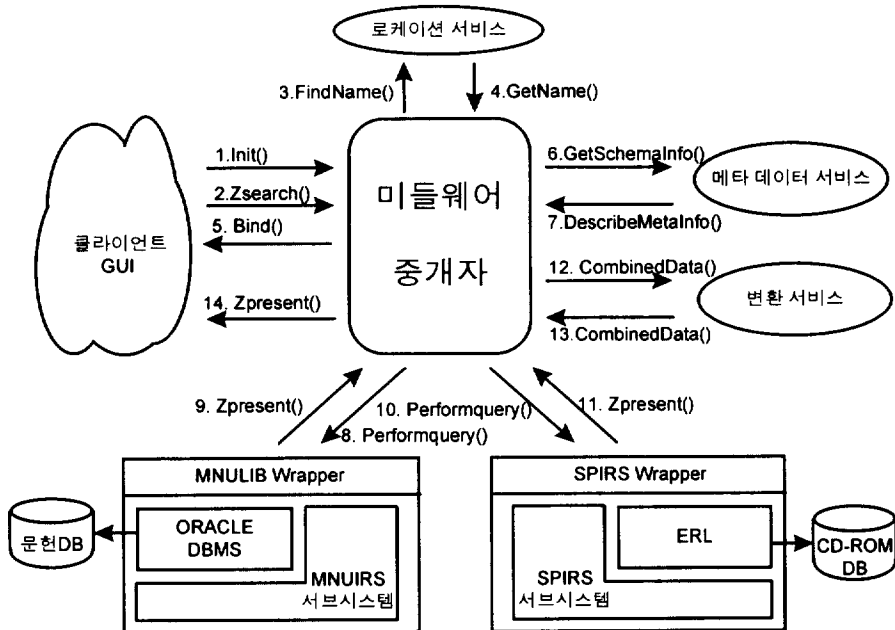
클라이언트의 검색 질의에 대해, DDIR/ORB가 기존의 정보검색시스템들의 연산을

기동하는 과정은 다음과 같다.

첫째, 클라이언트는 일관성있는 검색 GUI를 통해 주제별, 서명별, 저자별에 따른 검색 질의를 요구한다.

둘째, 미들웨어중개자의 로케이션 서비스는 클라이언트의 요구에 대한 검색 서버의 위치 정보를 찾고 로케이션 서비스에 등록된 객체 구현의 객체 참조를 반환한다.

셋째, 미들웨어중개자는 메타데이터 서비스로부터 각 지역에 구축되어 있는 문헌 DB의 스키마 정보 및 정보검색 인터페이스 정보를 얻는다. 즉, 메타데이터 서비스는 클라이언트가 주제별에 따른 문헌 검색을 요구하면 각 지역의 정보검색시스템에 구현된 주제별 정보 검색과 일치하는 인터페이스명을 제공한다.



<그림 5> DDIR/ORB의 미들웨어 중개자의 검색 질의 처리과정

넷째, 미들웨어중개자는 로케이션 서비스로부터 얻어진 객체 참조와 메타데이터 서비스로부터 얻어진 검색 연산을 performquery() 연산의 스크립트로 제공하고, 각 지역의 정보 검색 엔진이 실행되도록 한다.

다섯째, 미들웨어중개자는 각각의 객체 Wrapping 인터페이스를 통해서 각 지역의 정보검색시스템들이 해당되는 연산을 수행하도록 요구한다. performquery() 연산 수행 결과로 생성되는 검색 결과 레코드는 데이터 객체에 저장되고 전송될 데이터 객체는 팩토리에 의해서 관리된다. 여기서 팩토리는 검색된 레코드 수의 데이터 객체를 생성하고, 각 데이터 객체는 레코드의 필드 수만큼의 Name-Value 리스트를 생성한다.

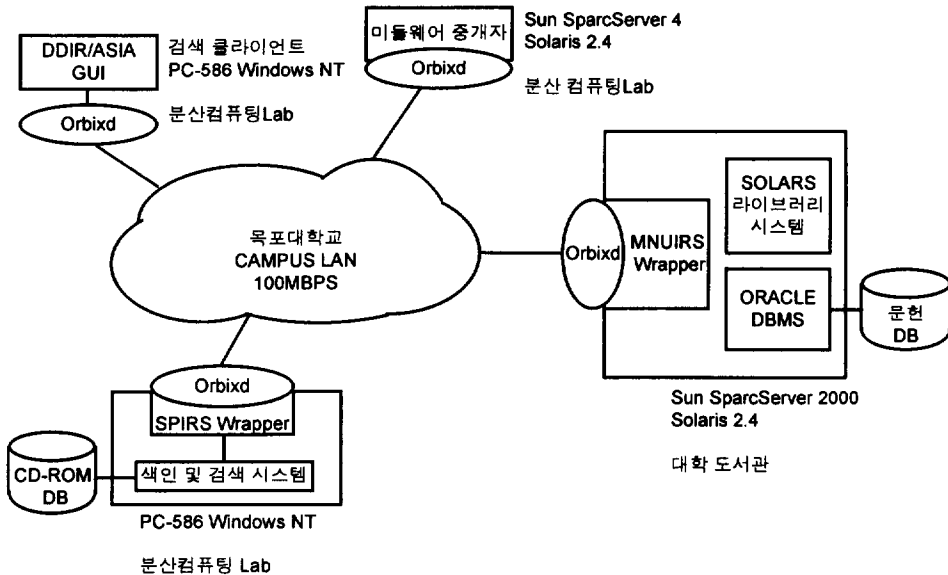
여섯째, 미들웨어중개자는 두 정보검색시스

템으로부터 얻어진 검색 결과를 변환 서버를 이용하여 각 레코드들을 합병하고 합병된 레코드들을 Zpresent() 연산에 의해서 클라이언트에게 전달한다.

일곱째, 클라이언트는 검색 질의 결과를 검색 결과 화면에 디스플레이한다. 만일 미들웨어중개자 또는 각 응용서버와의 통신 도중에 오류가 발생하면 예외 메시지를 디스플레이한다.

이와 같은 DDIR/ORB의 검색 처리 과정을 객체들의 연산들로 표현하면 <그림 5>와 같다.

5. 실험 및 검토



<그림 6> DDIR/ORB 실험 환경 구성도

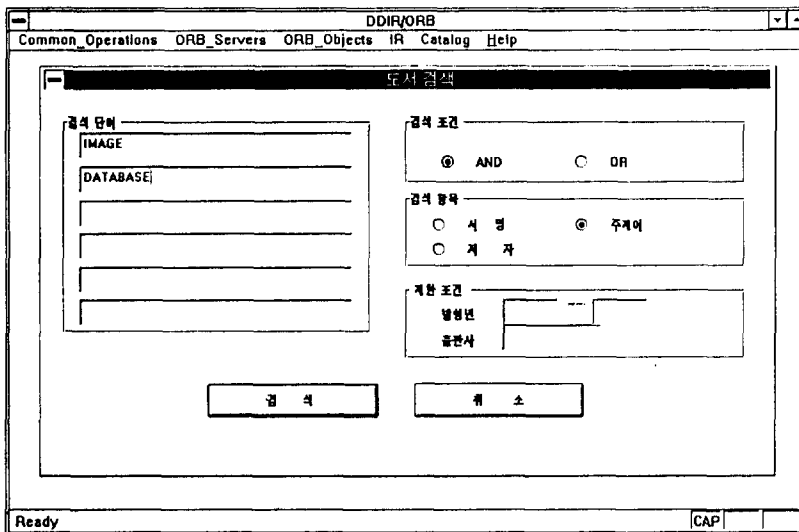
〈표 2〉 DDIR/ORB 실험 환경

구 분	H/W	S/W	설치장소	통신모듈
클라이언트	PC 586 Pentium	Windows NT3.0, Orbix1.3 Visual C++	분산컴퓨팅 실험실	TCP/IP
서버 1	PC 586 Pentium	Window NT3.0, Orbix 1.3, ERL DBMS SPIRS, 학위논문 CD-ROM DB	분산컴퓨팅 실험실	TCP/IP
서버 2	Sun Sparc Server 4	Solaris 2.4, Orbix 2.0 Middleware Broker	분산컴퓨팅 실험실	TCP/IP
서버 3	Sun Sparc Server 2000E	Solaris 2.4, Orbix 2.0 Oracle DBMS, MNUIRS	목포대학교 도서관	TCP/IP

5.1 실험 환경

CORBA기반의 정보검색시스템 통합 과정을 실험하기 위하여 〈그림 6〉 같은 환경을 구축하였다. 구축된 환경의 응용 서버1에는 오라클 DBMS에서 작동되는 서지 정보검색시스템을 설치하였고, 응용 서버2에는 미국 실버

플래터 회사의 ERL(Electronic ReferenceLibrary)에 의해서 관리되는 학위논문 CD-ROM DB와 SPIRS 정보검색시스템으로 구성된다. ERL에 의해 관리되는 학위논문 DB는 1992년부터 1995년까지의 약 20만 레코드의 석박사 학위논문이 CD에 저장되어 있다. 또한 목포대학교의 문헌 DB에는 약 5만



〈그림 7〉 클라이언트 검색 질의 요구 화면

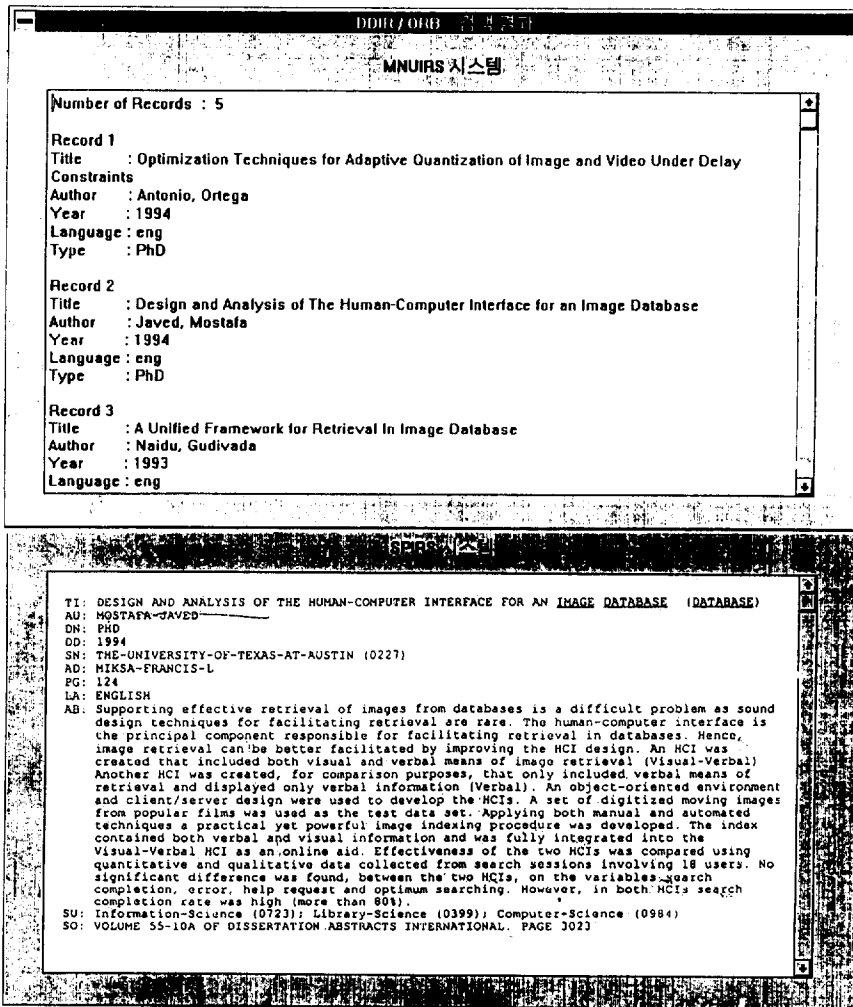
권의 단행본 및 50편의 정보 시스템 관련 학위 논문이 수록되어 있다.

다음 <그림 6>은 본 연구의 실험 환경 구성도이다. 본 연구에서 실험한 시스템 환경을 요약하면 <표 2>과 같다.

5.2 실험 결과

본 연구에서는 이용자에게 검색 질의 및 결과 출력에 대한 일관성을 유지하기 위하여 비주얼 C++를 이용하여 사용자 편의의 그래픽 인터페이스를 구현하였다. <그림 7>은 클라이언트의 검색 질의 요구 화면이다.

클라이언트의 검색 질의 요구는 서명, 주제명, 저자명에 의해서 이루어지고 제한 조건은



<그림 8> DDIR/ORB의 정보검색 결과 화면

출판년도 및 출판사로 한정하였다. 이러한 검색 질의 요구에 대하여 DDIR/ORB의 미들웨어중개자는 로케이션 서비스에 등록된 두 개의 정보검색시스템을 기동하고, 검색 결과 레코드들을 사용자가 원하는 형식으로 합병하여 텍스트 파일 형식으로 클라이언트에게 전송한다. <그림 8>은 주제명에 의한 정보검색 결과 화면이다.

5.3 검토 의견

DDIR/ORB는 지금까지 수년간 발전해 온 기존의 서지 정보검색시스템과 CD-ROM으로 구축된 텍스트 데이터베이스 시스템과의 통합을 통하여 기존 정보검색시스템의 재사용성을 만족시키고, 기존의 한 시스템이 제공할 수 없는 유용한 정보를 응용 서버들 간의 상호운용성을 통해 보여준다.

DDIR/ORB를 이용한 정보검색시스템 통합 실험 결과는 다음과 같은 특성을 제공한다. 첫째, DDIR/ORB는 이용자들에게 서명, 저자명, 출판사명 등으로 구성된 단순 서지정보의 리스트를 제공하고, 이용자들의 추가 요구에 의해서 서지정보와 동시에 초록을 제공함으로써 검색 즉시 선택된 문헌의 적합성 판단 정보를 제공한다. 둘째, 이용자들은 DDIR/ORB의 일관성있는 사용자 인터페이스를 통해 지리적으로 떨어져 있는 이형의 데이터베이스들을 쉽게 접근할 수 있고, 정보검색 결과를 클라이언트 요구에 맞는 데이터 형식으로 변환되어 디스플레이됨으로써 분산 자원들에 대한 접근 투명성 및 상호운용성을 효과적으로 제공한다. 셋째, 기존의 두 정보검

색시스템들의 객체 Wrapping 인터페이스 구현을 통해서 시스템 통합을 함으로써 실제 검색시스템 개발에 소요되는 비용을 최소화하였다. 즉, 실제 구현에 필요한 소스 코드 수를 대폭 감소할 수 있었다. 넷째, DDIR/ORB 구성요소 및 Z39.50 프로토콜 구현시 객체지향 기술을 활용함으로써 인터페이스 내부에 정의된 데이터 및 연산들에 대해 은폐성을 제공한다. 다섯째, 시스템 통합 모델 설계시 IDL을 사용함으로써 클라이언트와 서버에게 공통의 검색 인터페이스를 제공하고, DDIR/ORB의 주요 구성요소들은 공통의 기본 연산들을 상속받음으로써 통합 시스템의 구현비용을 최소화하였다.

DDIR/ORB는 위의 특성들을 제공하는 반면에 중간 서비스를 추가함으로써 응용 서버들에 대한 바인딩 시간 및 검색 연산들의 기동 시간에 대한 오버헤드(Overhead)가 존재한다. 그러나 본 연구에서 제한한 DDIR/ORB는 이용자가 느낄 정도의 실행 오버헤드는 나타나지 않았다.

6. 결 론

클라이언트-서버 기반의 분산환경에서 디지털도서관 이용자들은 컴퓨터 하드웨어 및 플랫폼들의 이형성, DBMS 이형성, 서로 다른 정보검색 S/W들의 사용, 이질적인 네트워크 전송 프로토콜의 사용에 관계없이 일관성 있는 질의처리 및 정보검색 결과를 요구한다. 본 연구에서는 이러한 이용자들의 요구를 만족하면서 지리적으로 분산된 정보자원들에 대

한 접근 투명성 및 검색 결과에 대한 일관성을 제공할 수 있는 CORBA기반의 문헌 정보 검색시스템(DDIR/ORB) 통합 모델을 제안하였다.

본 연구에서 제안한 CORBA기반의 정보검색시스템 통합모델은 질의를 요구하는 클라이언트와 검색을 실행하는 응용서버 사이에 미들웨어중개자를 두는 3-Tier 방식의 클라이언트-서버 구조이다. DDIR/ORB는 MARC기반의 서지 데이터베이스와 CD-ROM 텍스트 데이터베이스를 투명적으로 접근할 수 있음을 보장하였고, 정보검색 결과에 대한 자유로운 데이터 교환 및 변환을 제공하였으며, 기존의 정보검색시스템의 재사용 방법을 제공하였다. 또한 본 연구에서는 이형의 분산 환경에서 다양한 문헌 DB들의 서지정보를 탐색할 수 있게 하고 사용자에게 일관성 있는 검색 결과를 제공하기 위해서 표준기반의 분산 객체 컴퓨팅 기술을 이용하여 Z39.50 프로토콜 구현 방법을 제시하였다.

정보검색시스템 통합 실험 결과 나타난 DDIR/ORB의 특성은 다음과 같다.

첫째, 이용자들에게 서명, 저자명, 출판사항 등으로 구성된 단순 서지정보의 리스트를 제공하고, 이용자들의 추가 요구에 의해서 서지정보와 동시에 초록정보를 제공함으로써 검색 즉시 선택된 문헌의 적합성을 판단할 수 있게 하였다.

둘째, DDIR/ORB는 디지털도서관 환경에서 해결해야 할 시스템들의 분산성, 이형성, 상호운용성 문제를 해결하였다.

셋째, 시스템통합 모델 설계시 OMG IDL을 사용함으로써 인터페이스 복잡도가 감소되었으며, 각 구성요소들의 구현 비용을 최소화하였다.

본 연구의 제한점은 클라이언트와 서버 사이에 미들웨어중개자를 추가로 구현함으로써 미들웨어중개자의 실행 오버헤드가 발생하였다는 점이다. 따라서 본 연구와 관련된 다음 연구는 미들웨어중개자의 오버헤드를 최소화할 수 있는 방법을 계속 연구해야 되고 이미지, 그래픽, 오디오, 비디오 등을 검색할 수 있는 멀티미디어 정보검색시스템과의 통합 방법을 연구해야 된다.

참고문헌

- 국립중앙도서관. 1993. 한국문헌자동화목록형식 : 단행본용. 서울 : 국립중앙도서관.
- 김성혁. 1995. "인코딩 모뎀 및 방법론" 서지와 본문 데이터베이스 구축 방법론 세미나 자료집, 한국 과학기술원 과학도서관.
- 김태수, 최석두. 1996. "MARC와 SGML의 통합에 대한 연구" 제3회 한국정보관리학회 논문대회 논문집 : 67-70.
- 안개일, 원종상, 전우직. 1996. "웹 환경에서의 디지털도서관 표준 프로토콜 구현 방안 연구" 1996년도 한국정보과학회 가을 학술발표논문집 23, 2 :

- 1139-1142.
- 안현수. 1996. "정보검색 프로토콜(Z39.50)에 관한 고찰" 국회도서관보 33, 2 : 17-33.
- 원종상, 안개일, 전우직. 1996. "디지털도서관 표준 프로토콜 Z39.50 구현" 1996년도 한국정보과학회 가을 학술 발표논문집 23, 2 : 1143-1146.
- 정영미. 1996. "국내 문자정보 데이터베이스의 색인에 관한 연구" 정보관리학회지 13,1 : 19-45.
- 최한석. 1995. 정보통신 서비스를 위한 개방 분산 시스템구조에 관한 연구(2). 최종연구 보고서, 한국전자통신연구소.
- Choi, H.S., Yoo, Y.S. and Chang, O.K. 1996. "A New Control Service Model Based on CORBA for Distributed Multimedia Objects" ; Proceedings of 22nd EUROMICRO Conf. IEEE : 467-474.
- Choi, H.S. et al. 1996. "ASIA : An Advanced System Integration Architecture Based on CORBA for Interoperability among distributed Objects." EUROMICRO '96 Short Contribution, IEEE.
- Cole, T.W. and Kazmer, M.M. 1995. "SGML as A Component of The Digital Library." Library Hi Tech 13, 4 : 75-90. Fidel, R. 1991. Searchers' selection of search keys : II. Controlled vocabulary or free-text searching. JASIS 42,7 : 501-514.
- Geneserth, Michael and Ketchpel, Steven. 1994. "Software Agents." Comm of ACM 37, 7 : 48-53, 147.
- IONA. 1995. Distributed Object Technology. IDL to C++ mapping. IONA Technologies Ltd. ISO. 1986. Information Processing - Text and Office Systems - Standard Generalized Markup Language(SGML). International Organization of Standardisation.
- Katz, M., Cornwell, D. and Mowbray, T.J. 1993. "System Integration with Minimal Object Wrappers" Proceedings of Tools 93.
- Mowbray, Thomas and Zahavi, Ron. 1995. The Essential CORBA : Systems Integration using Distributed Objects, Object Management Group. John Wiley & Sons, Inc.
- Object Management Group. 1992. Object Management Architecture Guide. OMG TC Document 92.12.1.
- Object Management Group. 1993. The Common Object Request Broker : Architecture and Specifica-

- tion, Revision 1.2. OMG TC Document 93.12.43. Object Management Group. 1994. Common Object Services Specifications, Volumn I. OMG Document Number 94.1.1.
- SilverPlatter. 1995. SilverPlatter Directory : A Guide to a Worldwide Library. Norwood : SilverPlatter Information.
- Widerhold, Gio. 1995. "Digital Libraries, and Productivity." Comm. of the ACM, 38, 4 : 85-96.
- Wiederhold, Gio. 1994. "Interoperation, Mediation, and Ontologies." Proceedings International Symposium on Fifth Generation Computer Systems (FGCS94). Workshop on Heterogeneous Cooperative Knowledge-Bases, Vol. W3 : 33-48, ICOT, Tokyo, Japan.
- Wiederhold, Gio. 1996. "Value-added Mediation in Large-Scale Information Systems" Computer Science Department, Stanford University.