

# 멀티미디어 클라이언트-서버 응용 개발 도구인 한우리/C의 미들웨어

임 채 덕<sup>†</sup>

## 요 약

본 논문에서는 기존의 클라이언트-서버 응용 개발 도구에 비해서 멀티미디어 처리 기능이 강화되고, 분산 시스템 소프트웨어의 건위(Front-End) 역할을 수행 하는 멀티미디어 클라이언트-서버 응용 개발 도구(한우리/C)를 제안한다. 한우리/C는 분산 처리 환경(Distributed Computing Environment: DCE) 위에서 멀티미디어 응용 개발을 지원하는 제 4세대 언어 형태의 클라이언트-서버 도구이다. 본 논문에서는 한우리/C를 구성하는 요소들 중에서 미들웨어 부분에 대한 설계와 구현된 내용을 설명한다.

## Middleware on Hanuri/C as a Multimedia Client-Server Application Development Tool

ChaeDeok Lim<sup>†</sup>

## ABSTRACT

The Client-Server model is a computing technique that efficiently uses distributed resources via networks. But, generating an application using Client-Server model requires many different expertises to developers in comparison to that of using the centralized computing method. That is, an application developer must be familiar with network programming and GUI(Graphical User Interface) techniques in addition to conventional programming skills. Accordingly, the time and man power problems have been issued for building a Client-Server system. To alleviate these problem, the Client-Server applications development tools are needed. To meet such a need, we developed a GUI based tool, called Hanuri/C, for generating Client-Server application programs. In comparison to existing Client-Server tools, Hanuri/C is reinforced Multimedia facilities. Hanuri/C is considered as a front-end part of distributed system software. That is, Hanuri/C is a Client-Server tool similar to Forth Generation Language(4GL) supporting multimedia application on top of the Distributed Computing environment(DCE). In this paper, we present the design and implementation method of the middleware part of Hanuri/C.

### 1. 서 론

컴퓨터를 이용하는 형태를 시대 별로 살펴보면, 1960년대는 일괄 처리(Batch Processing), 1970년대에는 컴퓨터의 능력을 작은 시간 대로

나누어 여러 사람이 사용하는 시분할 처리(Time-Sharing Processing), 1980년대에 개인용 컴퓨터나 워크스테이션의 등장으로 개인 처리(Personal Processing)가 가능하였고, 1990년대에 들어서는 분산 처리(Distributed Processing/Computing) 시대가 시작되었다[1].

정보 통신 분야에서 핵심적인 역할을 하는 컴퓨터의 시스템 구조가 독자적인 시스템 구조(Proprietary System)에서 개방적인 구조(Open Sy-

\* 이 연구는 정보통신부의 정보통신 연구개발 사업으로 수행한 연구 결과임.

† 정 회 원: 한국전자 통신연구소 연구원

논문접수: 1995년 7월 30일, 심사완료: 1995년 10월 20일

stem)로 전환되면서 컴퓨터 통신망 기술이 요구되고 있다. 또한, 응용 분야 측면에서 살펴보면 전산 처리 기법의 향상과 컴퓨터 기능들의 다운사이징 추세에 따라, 통신망 기술의 응용 분야는 온라인 트랜잭션 처리 분야에서 클라이언트-서버 모델을 지닌 응용 분야로 다양하게 변화되고 있다.

분산 처리란 “컴퓨터 통신망에 연결된 여러 대의 컴퓨터 시스템들이 서로 도와 하나의 특적인 서비스를 제공해 주고자 처리하는 방법”을 총칭하는 광범위한 의미를 지니고 있다. 분산 처리 기술의 목적은 통신망에 연결된 여러 자원들을 마치 사용자가 혼자서 사용하는 것과 같도록 지원해주는 것이다. 클라이언트-서버 기술[2,3]은 분산 처리 기술의 한 가지 특수한 형태로서 어떤 일을 처리하는 모형이다. 여기서 클라이언트는 응용 사용자, 응용 프로그래밍, 컴퓨터 시스템 등과 같이 일을 의뢰하는 자이고, 서버는 그 일을 처리하여 결과를 내놓는 서비스 제공자란 개념으로 응용 프로그램들 간을 상호 연결하는 방식이다.

분산 처리 기술로 해결하고자 하는 문제를 살펴보면, 기존의 정보 처리 기술로는 불가능한 새로운 영역도 있지만, 그보다는 문제를 보다 효율적으로 해결할 수 있다는 측면에 관점을 둔다. (그림 1)은 현재 상품화되었거나 상품화되고 있는 분산 처리 관련 시스템 소프트웨어 모습의 개략적인 구성도[10]이다. 그림 1을 살펴보면 온라인 트랜잭션 처리나 데이터베이스 응용이 여전히 가장 중요한 응용 분야이고, 가장 기본적인 구성 요소인 대표적인 분산 처리 기본 소프트웨어는 OSF/DCE(Open Software Foundation/Distributed Computing Environment)[4]와 Sun사의 ONC+(Open Network Computing)이다.

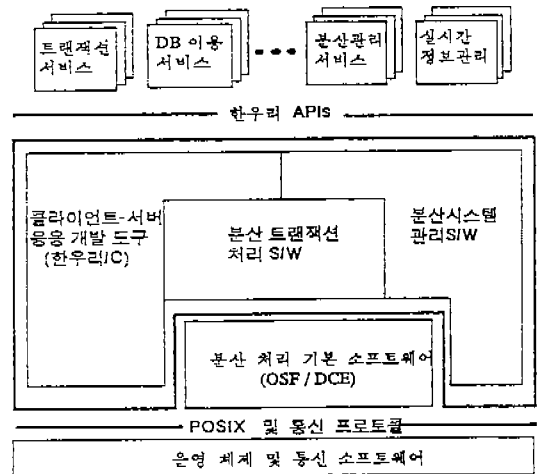
분산 트랜잭션 처리 소프트웨어는 분산 환경에서 여러 개의 이질적인 데이터베이스 관리 시스템(DBMS)들을 이용하면서 온라인 트랜잭션 처리 응용을 개발할 수 있게 하는 소프트웨어로서, 일반적으로 트랜잭션 처리 모니터(Transaction Processing Monitor)라고 한다.

분산 처리 환경은 자원들이 다양하고 구성 기종들이 이기종이면서 확장성을 지니고 있기 때문

에, 기존의 단일 시스템을 관리하던 방식은 더 이상 통신망을 이용하는 분산 환경에 적용되지 못한다. 따라서, 컴퓨터 통신망 관리와 시스템 자원 관리에 분산 처리의 특징을 고려한 새로운 관리 기법과 기술이 요구되는데, 이것이 분산 시스템 관리 소프트웨어이다.

분산 처리 환경을 이용한 응용들을 개발하기 위한 도구는 현재 가장 활발하게 발표되는 상품 분야로서, 제 4세대 언어 형식의 도구들이 제안되어 응용 개발을 지원해 주고 있지만, 이 분야에서의 주요 관건은 미들웨어이다. 현재 발표된 상품 제품들은 클라이언트-서버 모델 하에서 TCP/IP(Transmission Control Protocol/Internet Protocol) 통신망을 이용하고, 관계형 데이터베이스 접속 기능을 지원하고 있으나, 분산 처리 환경 소프트웨어나 분산 트랜잭션 처리 소프트웨어 등 분산 처리 소프트웨어와 연결시켜주는 기능을 지닌 제품은 아직 없다. 더욱이, 하이퍼미디어 기능을 지원해 주고 분산 시스템 관리 기능과 객체 지향형의 기술을 지원하는 도구는 현재 없다.

본 논문에서는 기존의 클라이언트-서버 응용 개발 도구에 비해서 멀티미디어 처리 기능이 강화되었고, 분산 시스템 소프트웨어의 전위 역할을 수행할 수 있는 멀티미디어 클라이언트-서버 응용 개발 도구(한우리/C)를 제안하였다. 한우



(그림 1) 한우리의 구성도 (Fig. 1) Structure of Hanuri

리/C는 분산 처리 환경 위에서 멀티미디어 응용 개발을 지원하는 제 4세대 언어 형태의 클라이언트-서버 도구이다. 특히, 이 논문에서는 한우리/C를 구성하는 요소들 중에서 데이터 관리 기능과 통신 관리 기능을 중점적으로한 미들웨어 부분에 대하여 설계 및 구현한 내용을 설명한다.

본 논문의 구성은 다음과 같다. 제 2절에서는 한우리/C의 각 모듈별 기능과 전체적인 구조에 대하여 설명하고, 제 3절에서는 한우리/C의 미들웨어에 대하여 설계된 내용을 설명하고, 제 4절에서는 설계된 내용을 구현한 결과에 대하여 설명한 후, 마지막 절에서는 설계 및 구현된 내용을 토대로 한우리/C와 유사한 클라이언트-서버 응용 개발 도구들을 비교하였고, 한우리/C를 이용한 응용 프로그램인 홈쇼핑에 대한 소개와 향후 연구되어야 할 부분에 대하여 언급하였다.

## 2. 한우리/C의 구성

한우리/C는 (그림 2)에서와 같이 여러 개의 모듈로 나뉜다. 이 모듈들은 크게는 유저웨어 (Userware), 미들웨어(Middleware)와 엔터프라이즈웨어(Enterpriseware)와 하드웨어(Hardware)로 나눌 수 있다[9].

유저웨어는 사용자와 직접적인 대화가 가능한 화면을 구성하는데 주로 관련이 되고, 미들웨어는 서버에 있는 특정 서비스를 이용하기 위해 중간자 역할을 하는 부분이다. (그림 2)에서 보고서 처리 모듈, 양식 처리 모듈, 객체 주고받기 모듈이 유저웨어에 속하고, 한우리/C 접속모듈, ODBC 접속 모듈 등이 미들웨어에 속한다.

미들웨어에 속해 있는 모듈들은 반드시 한우리

/C APIs(Application Programming Interfaces)를 통해 다양한 서비스를 받을 수 있게 된다.

엔터프라이즈웨어에는 한바다, Informix, Oracle, Sybase와 같은 DBMS가 해당된다. 하드웨어는 한우리/C를 개발하는데 필요한 통신망이나 장치 드라이버 등을 포함한다.

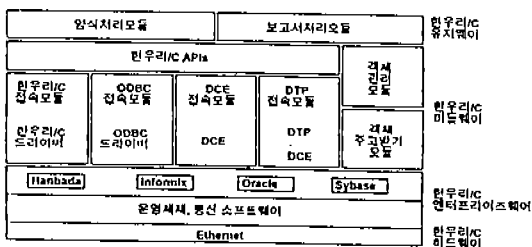
한우리/C의 모듈들을 각각 살펴 보면 다음과 같다.

**\*양식 처리 모듈:**이 모듈은 화면을 구성하고 있는 디스플레이 객체를 생성, 수정, 기능화하는 것을 담당한다. 하나의 양식에는 여러 개의 단위 객체, 동작 객체, 멀티미디어 객체, 데이터베이스 객체들로 구성된다. 따라서, 하나의 양식을 편집하는 데에는 해당 객체의 속성이나 값들을 설정하고 변경할 수 있는 개별 객체 기능과 양식 내의 객체들에 대한 편집 기능을 지니고 있어야 한다.

개별 객체 기능으로는 멀티미디어 데이터를 취급할 수 있는 기능, 서버의 데이터베이스를 취급할 수 있는 기능과, 양식 내에 객체들을 편집하는 기능인 위치 변환 기능, 크기 변경 기능, 복사 및 삭제 기능, 색상과 글꼴 및 글자 폰트를 지정하고 변경하는 기능 등 다양한 기능을 보유하고 있다.

**\*보고서 처리 모듈:**서버의 데이터베이스에 있는 데이터와 클라이언트 자체 데이터를 이용하여 다양한 형태의 보고서 양식을 생성, 수정, 삭제 및 생성된 보고서 양식에 따라 보고서를 작성하는 모듈이다. 작성된 보고서는 사용자의 필요에 따라 프린터나 화면 또는 화일로 출력할 수 있다. 한우리/C에서 지원 가능한 보고서의 형태는 일반 업무에 가장 많이 사용하는 보고서로서 보고서 처리 모듈의 다양한 기능들을 가장 많이 요구하는 일반 보고서, 회사의 인사 기록표와 같은 표 형태의 보고서, 다량의 우편 발송 등이 필요할 때 사용되는 우편 주소 형태 보고서, 보고서 처리 모듈에서 제공하는 기능을 이용하여 사용자가 정의한 형태의 사용자 정의 보고서가 있다.

**\*객체 관리 모듈:**이 모듈은 한우리/C의 여러



(그림 2) 한우리/C의 모듈 구성도  
(Fig. 2) Block Diagram of Hanuri/C

객체들의 정보를 관리하는 모듈이다. 즉, 양식 처리 모듈 혹은 보고서 처리 모듈에 의해 생성된 양식을 저장, 추출하는 기능을 효율적으로 수행하기 위한 모듈이다. 저장 시에는 스크립트(script) 형태로 저장되었다가, 추출 시에는 양식 처리 모듈에서 처리할 수 있는 스크린 데이터 형태로 변환된다. 또한, 필요에 따라 기존에 있는 편집기(editor)를 이용하여 스크립트 확일을 수정할 수도 있다.

\* 객체 주고받기 모듈: 양식 처리 모듈과 보고서 처리 모듈과 상용화된 다른 응용 프로그램 사이에 상호 필요한 데이터나 객체를 서로 주고 받을 수 있도록 지원해 주는 모듈이다. 예를 들어, 보고서 처리 모듈에서 사용하고 있는 데이터를 그래픽으로 그리는 프로그램에 데이터를 주고, 그 결과 그래프를 받아 들어는 역할을 한다. 이 모듈은 OLE(Object Linking Embedding)[5]와 DDE(Dynamic Data Exchange)[5]를 이용하여 다른 프로그램들과 객체 주고받기를 수행한다.

\* 한우리/C APIs: 한우리/C에서 제공하는 고유의 APIs로 양식 처리 모듈이나 보고서 처리 모듈이 서버와 통신하거나, 데이터베이스에 접근 및 데이터를 조작할 때, 하부의 네트워크나 접속방법을 알 필요없이 일관된 방법으로 사용할 수 있도록 해주는 APIs이다. 한우리/C APIs는 하부의 질의-네트워크 관련 모듈들과 연결되어 한우리/C의 미들웨어로서 투명성(transparency), 상호이식성(interoperability)등을 제공하게 된다.

\* 한우리/C 접속 모듈: 한우리/C에서 독자적인 방법으로 서버와 통신하거나, 데이터베이스에 접근 및 데이터를 조작할 수 있도록 한우리/C APIs와 서버 쪽의 서비스 드라이버와 인터페이스하는 역할을 담당한다. 한우리/C 접속 모듈은 개방형 데이터베이스 접근 방식인 ODBC(Open DataBase Connectivity)를 제공하지 않는 DBMS를 사용할 수 있는 가능성을 제시한 것이며, 데이터베이스 접근 속도 등을 높일 수 있도록 사용상의 효율성을 강조하여 개발하였다.

\* ODBC 접속 모듈: 한우리/C APIs를 통하여

개방형 데이터베이스 접근 방식인 ODBC를 이용할 수 있도록 지원해주는 모듈이다.

\* DCE 접속 모듈: 한우리/C APIs를 통하여 분산 환경에서 RPC(Remote Procedure Call), 디렉토리 서비스(Directory Service), 보안(security), 분산 화일 서비스(Distributed File Service), 분산 시간 서비스(Distributed Time Service), 스텟드 등의 DCE 응용 서비스를 이용할 수 있도록 지원해 주는 모듈이다.

\* DTP(Distributed Transaction Processor) 접속 모듈: 한우리/C APIs를 통하여, 한우리/C가 분산 처리 환경에서 분산 트랜잭션 처리를 이용한 다양한 응용을 만들 수 있도록 지원해주는 모듈이다.

\* 한우리/C 드라이버: 한우리/C에서 독자적인 방법으로 서버와 통신하거나 특정 데이터베이스에 접근 및 데이터를 조작할 수 있도록 각 DBMS 별로 제공되는 모듈이다. 한우리/C 드라이버는 클라이언트용과 서버용이 각 DBMS 별로 각각 제공된다.

\* ODBC: 마이크로소프트사에서 발표한 개방형 데이터베이스 접속 방식인 ODBC를 따르는 소프트웨어이다. 한우리/C가 ODBC를 지원하는 DBMS에 접근할 수 있도록 해준다.

\* DCE: OSF에서 발표한 분산 처리 환경 소프트웨어이다. 한우리/C가 DCE 환경에서 다양한 응용을 만들 수 있도록 해준다.

\* DTP: DCE 환경 하에서 분산 트랜잭션 처리용 소프트웨어이다. 한우리/C가 분산 처리 환경에서 분산 트랜잭션 처리기를 이용한 다양한 응용을 만들 수 있도록 해준다. 한우리/C는 화면을 구성할 수 있는 많은 객체들을 지원하고, 한우리/C에서는 하나의 화면 혹은 여러 개의 화면을 양식이라 정의하여 사용한다. 양식을 구성하는 객체들에는 선, 상자와 같은 단순 객체가 있고, 버튼(Button), 핫워드(Hot Word), 영역(Region), 주석(Annotation) 등의 동작 객체가 있고, 테이블, 리스트, 필드와 같은 데이터베이스 객체가 있고, 이미지, 오디오, 비디오와 같은 멀

티미디어 객체가 있다. 또한, 여러 개의 양식 화면은 자동적인 순차 디스플레이가 가능하다.

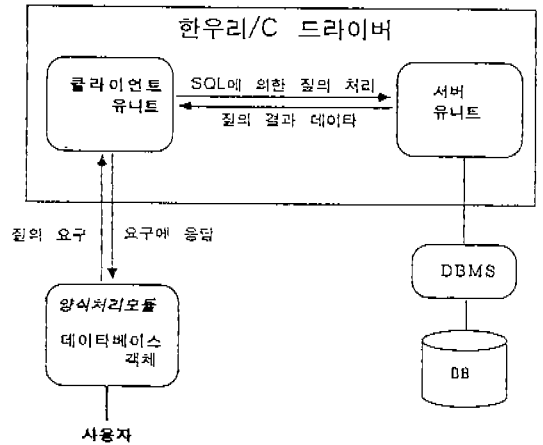
### 3. 데이터베이스 접근용 미들웨어: 한우리/C 드라이버

이 절에서는 (그림 2)에 있는 미들웨어를 구성하는 모듈들 중 한우리/C 드라이버에 대해 설명한다. 이 모듈은 양식 처리 모듈에 의해 생성된 데이터베이스 객체의 요청에 따라 TCP/IP를 통해 서버에 있는 데이터베이스를 접근하는 모듈이다. 멀티미디어 데이터를 표현하기 위해, 한우리/C 드라이버에서는 BLOB(Binary Large Object) 데이터를 처리할 수 있도록 지원한다. 한우리/C 드라이버의 내부를 살펴 보면 크게 네트워크 연결 모듈과 질의 처리 모듈로 나눌 수 있다. 네트워크 접속 모듈은 본산 환경에 있는 특정 서버와 연결하고, 질의를 처리하기 위해 데이터 패킷을 보내고, 서버로부터 데이터를 받는 역할을 담당한다. 질의 처리 모듈은 네트워크 접속 모듈을 통해 서버에 있는 데이터베이스관리 시스템을 접근함으로써 SQL 문을 처리할 수 있도록 한다. 이 모듈은 또한, SQL 문을 생성, 편집, 삭제, 열거하는 기능을 가지고 있다. 질의 처리 모듈과 네트워크 접속 모듈은 클라이언트와 서버에 각각 존재하고 있어서 네트워크를 통한 질의 처리를 가능하게 해준다. 본 논문에서는 클라이언트에 있는 있는 부분은 클라이언트 유니트, 서버에 있는 한우리/C 드라이버는 서버 유니트라 명명하였다. (그림 3)에서 보듯이, 클라이언트 유니트는 양식 처리 모듈의 함수 호출에 의해 동작된다. 양식 처리 모듈에서는 질의 처리를 위해 서버와의 연결, 사용자 로그인 등의 함수 호출을 한 다음, 질의 처리를 위한 함수 호출을 계속할 수 있다.

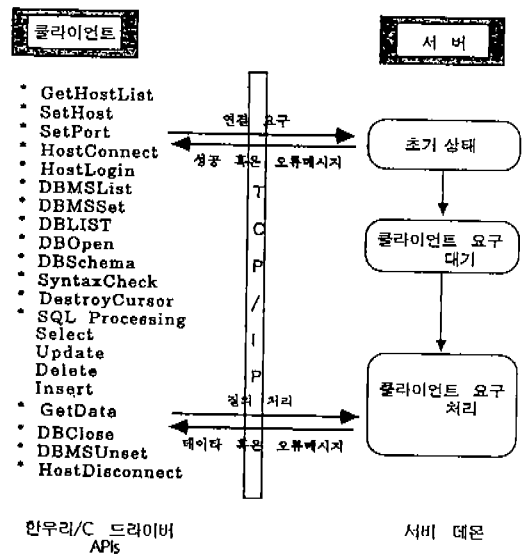
클라이언트 유니트는 양식 처리 모듈로부터 매개 변수를 받아 적절한 패킷을 만들어 TCP/IP를 통해 서버에게 보낸다. 그러면 서버 유니트는 클라이언트 유니트로부터 받은 패킷의 요구 사항에 맞는 처리를 수행한 다음, 그 결과를 패킷으로 만들어 클라이언트 유니트에 송신한다. 서버로부터 받은 결과는 최종적으로 양식 처리 모듈

에게 전달된다. 한우리/C 드라이버의 클라이언트 유니트와 서버 유니트의 기능적 흐름은 (그림 4)와 같다.

클라이언트 유니트에서 만드는 패킷은 명령어 부분과 그 명령어를 수행하는 데 필요한 데이터 부분으로 구성된다. 클라이언트에서 보낸 패킷을 받은 서버 유니트는 명령어 부분과 데이터 부분을 분리하는 작업을 수행한 다음, 그 명령어를



(그림 3) 한우리/C 드라이버와 양식 처리 모듈과의 관계  
(Fig. 3) Relationship between Hanuri/C Driver and Form Processing Module



(그림 4) 질의 처리를 위한 한우리/C 드라이버의 기능들  
(Fig. 4) Functional Flow for Query Processing in Hanuri/C Driver

- GetHostList
- SetHost
- SetPort
- HostConnect
- HostLogin
- DBMSList
- DBMSSet
- DBLIST
- DBOpen
- DBSchema
- SyntaxCheck
- DestroyCursor
- SQL Processing
- Select
- Update
- Delete
- Insert
- GetData
- DBClose
- DBMSUnset
- HostDisconnect

처리할 수 있는 상태로 천이한다. 서버 유니트는 패킷의 요구를 모두 처리한 다음, 그 결과를 패킷으로 만들어 클라이언트 유니트에 응답한다.

본 연구에서는 클라이언트 유니트를 함수 라이브러리 형식으로 구현한 반면, 서버 유니트는 데몬 형식으로 구현하였다.

서버 유니트는 패킷의 명령어 정보에 따라 적절한 상태로 천이하는 것을 반복한다. 각 상태에서는 여러 개의 명령어가 수행될 수 있는데, 예를 들어, "HMDInit" 상태에서 "User"라는 명령어가 수행될 수도 있고, "Quit"이라는 명령어도 수행될 수 있다. 즉, "HMDInit" 상태에서 "User"라는 명령어를 만나면 "User" 상태로 천이하고, "Quit"이라는 명령어를 만나면, 서버 유니트는 일의의 클라이언트의 요구를 벗기(편의상 NoState라 명명하였음)하는 상태에 있게 된다. 또한, "HMDInit" 상태에서 오류가 발생하면, 서버 유니트는 다시 "HMDInit" 상태로 들어가 다음 명령어를 기다리게 된다[표 1 참조]. (그림 4)에 있는 클라이언트 유니트의 각 함수들을 간략히 설명하면 다음과 같다. 모든 함수들에 대하여 오류가 발생하였을 경우에는 해당 오류 메시지를 클라이언트 유니트에 전달한다.

- GetHostList: 이 함수는 네트워크에 연결되어 있어 클라이언트에서 연결이 가능한 서버급

호스트들의 리스트를 반환한다.

- SetHost: 이 함수는 GetHostList에 의해 반환된 호스트들 중 하나를 선택한다.
- SetPort: 이 함수는 클라이언트에게 서버의 TCP/IP 서비스 포트를 알려주는 역할을 한다. 이 서비스 포트 번호는 양식 처리 모듈이 갖고 있다.
- HostConnect: 이 함수는 지정된 서버의 서비스와 연결하는 기능을 제공한다. 클라이언트 유니트와 서버 유니트의 실제적인 연결은 이 함수에 의해 이루어진다.
- HostLogin: 이 함수는 일의의 사용자가 서버에 있는 DBMS에 로그인하는 역할을 담당한다. 즉, 이 함수는 일의의 사용자가 DBMS를 사용할 수 있는 권한을 가졌는지 검사하는 기능을 제공한다.
- GetDBMSList: 이 함수는 DBMS 리스트를 반환한다.
- SetDBMS: 이 함수에 의해 DBMS가 선택되면, 해당 DBMS 연결이 동작한다. DBMS를 사용하기 위해서는 SQL\*EXEC path와 같은 환경 변수가 이미 설정되어 있어야 한다.
- GetDBList: 이 함수는 해당 DBMS에 있는 데이터베이스 리스트를 반환한다.
- DBOpen: 이 함수는 하나의 데이터베이스를 선택하여 데이터베이스를 연다.
- GetDBSchema: DBMS로부터 데이터베이스 스키마 정보를 얻기 위해 클라이언트 유니트는 질의문을 작성해서 서버에 질의 처리를 요청하고, 서버로부터 받은 결과를 해당 자료 구조로 바꾸는 역할을 담당한다.
- SyntaxCheck: 이 함수는 질의문의 문법이 맞는지를 확인하는 역할을 담당한다.
- Select: 이 함수는 스크롤 커서를 사용하여 SELECT문을 처리하는 역할을 한다. BLOB 데이터를 처리할 수 있는 프로그램 구조를 가지고 있다. 일단, 커서가 준비되면, FETCH 함수를 이용하여 데이터를 가져온다.
- Delete, Insert, Update: 이 함수들은 각각 DELETE, INSERT, UPDATE문을 처리한다.
- GetData: 이 함수는 DBMS의 FETCH 함수와 같은 기능으로서, 스크롤 커서에서 사용자가 원

〈표 1〉 한우리/C 드라이버의 서버 데몬의 상태 천이도 (Table. 1) Stwte Transition Diagram on Hanuri/C Driver

State	Commands	Next State	Error State
HMDinit	User	User	HMDinit
	Quit	NoState	NoState
User	SetDBMS	Session	Session
	Quit	NoState	NoState
Session	DBEnv	Session	Session
	GetDBList	GetDBList	Session
	DataBase(u)	CMD	Session
	CloseCrsr	CMD	CMD
	Quit	NoState	NoState
GetDBList	DataBase(d)	CMD	GetDBList
	CloseCrsr	CMD	CMD
	Quit	NoState	NoState
CMD	DBSchma	CMD	CMD
	HSQL	CMD	CMD
	Fetch	CMD	CMD
	CloseCrsr	CMD	CMD
	DBClose	NoState	NoState
	Quit	NoState	NoState

하는 만큼의 데이터 레코드를 가져올 수 있다.

- DestroyCursor: 이 함수는 사용 중이던 스크롤 커서를 닫는다.
- DBClose: 이 함수는 사용 중이던 데이터베이스를 닫는다.
- DBMSUnset: 이 함수는 사용 중이던 DBMS 엔진을 정지시킨다.
- HostDisconnect: 이 함수는 연결된 서버와의 통신을 끝내는 역할을 한다.

#### 4. 구현 결과

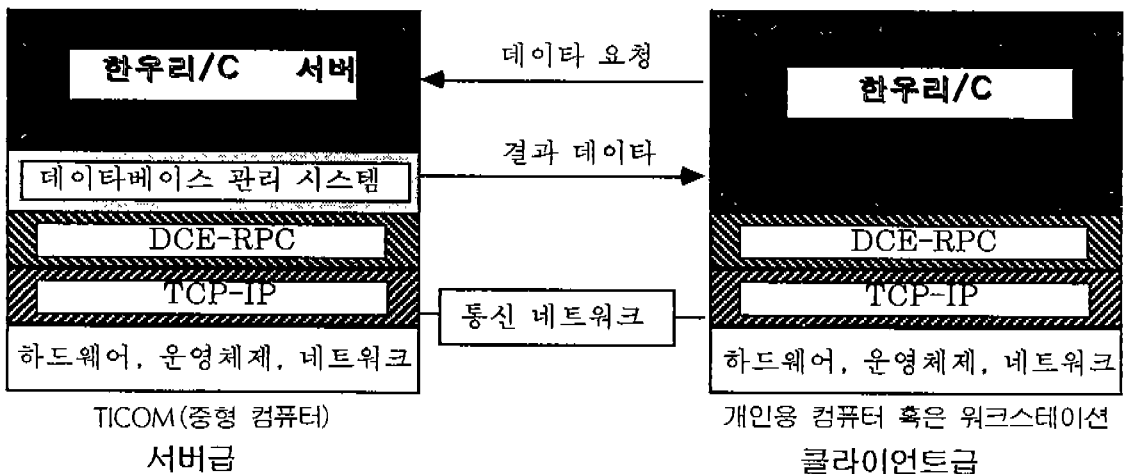
한우리/C는 (그림 5)와 같은 운용 환경을 지원할 수 있도록 구현되었다. 한우리/C 서버는 관계형 DBMS가 탑재되어 있고, 통신 방식은 TCP/IP를 사용한 TICOM이고, 한우리/C 클라이언트는 MS-DOS/Windows 3.1 하에서 CNFS (Chameleon Network File System)을 지니고 있는 PC/486이다.

한우리/C 미들웨어 계층 중에서 구현 완료된 한우리/C 드라이버 모듈과 한우리/C 사용자웨어 계층의 양식 처리 모듈 간의 질의 처리 및 통신용과 데이터 송수신 흐름은 상당히 간단하다. 즉, 제일 먼저 서버측에 내장된 데이터베이스에 접근하기 위해서 새로운 양식을 생성하거나 기존 양식을 파일로 연 후에 서버측의 DBMS와 연결

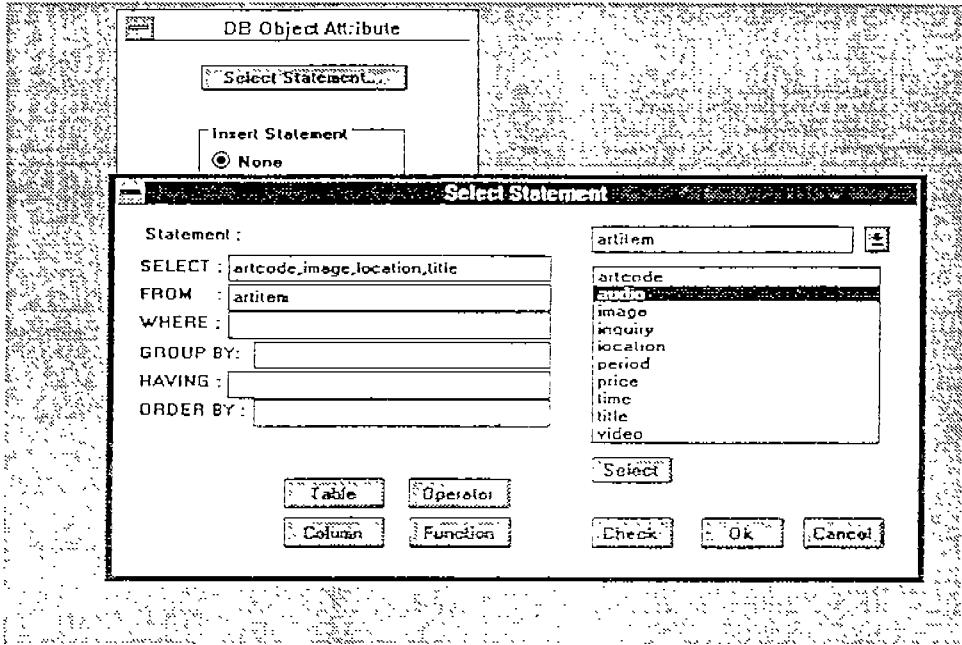
(3절의 GetHostList, SetHost, SetPort, HostConnect, HostLogin, GetDBMSList, SetDBMS가 담당)한다. DBMS가 연결되면, 사용하고자 하는 데이터베이스를 선택(3절의 GetDBList, Disconnect이 담당)하여 문장을 사용(3절의 Select, Insert, Update, Delete가 담당)하여 질의를 시작하여 처리한다.

이에 대한 실례를 살펴보면, 그림 6의 SELECT 문을 편집하는 화면에서 화면의 왼쪽 부분은 질의할 질의문의 구조가 나타나며, 오른쪽 부분은 질의 처리를 하고자 선택한 데이터베이스의 스키마 정보가 표시된다. (그림 6)에서의 각각의 버튼들의 기능을 살펴보면, "Table" 버튼은 선택한 데이터베이스의 테이블 리스트를 보여주고, "Column" 버튼은 선택한 테이블의 컬럼 리스트를 보여주는 역할을 담당한다(3절의 GetDBSchema가 담당).

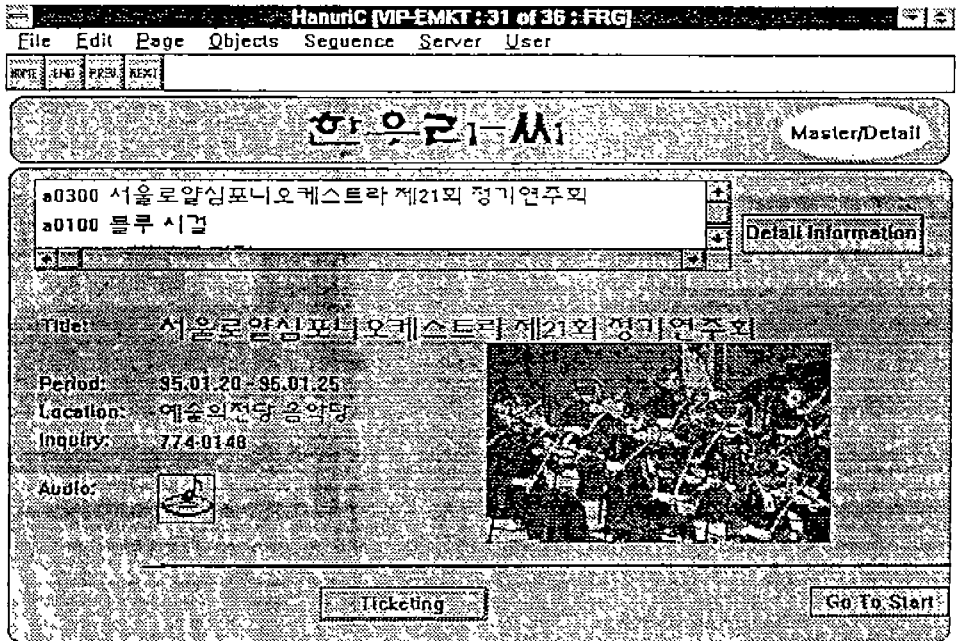
"Operator" 버튼은 "+", "-"와 같은 산술 연산을 하기 위한 정보를 제공하고, "Function" 버튼은 문의 합계와 평균 등과 같은 내장 함수를 사용하기 위한 정보를 제공하는 기능을 지니고 있고, "Select" 버튼은 오른쪽 상자 안의 데이터베이스 스키마 정보를 이용하여 질의문을 작성할 때 필요한 칼럼 이름을 지정할 때 사용할 수 있는 기능을 제공한다. 또한, "Check" 버튼은 질의문을 서버를 통해 수행시켜보기 전에, 문법에 오



(그림 5) 한우리/C의 개발 환경  
(Fig. 5) Developing Environment of Hanuri/C



(그림 6) SELECT문을 편집하는 화면  
 (Fig. 6) Screen for Editing a Select Statement



(그림 7) 서버로부터 가져온 데이터를 보여주는 화면  
 (Fig. 7) Screen for showing the Data Accessed from Server



류가 있는 지를 먼저 확인(3절의 SyntaxCheck 가 담당)하여, 질의하는 데 소요되는 시간을 줄이기 위해 두었다.

양식 처리 모듈은 (그림 6)의 화면을 통해 SQL 문을 편집하여 한우리/C 드라이버의 클라이언트 유니트에 정보를 송신하고, 클라이언트 유니트에서는 이를 한우리/C 드라이버 서버 유니트에 송신하여 SQL 문장을 처리한다. 서버 유니트가 한우리/C 서버의 데이터베이스와 송수신하여 처리된 결과는 다시 양식 처리 모듈까지 수신되어서 표현된 결과 화면의 실행은 (그림 7)과 같다. (그림 7)에서 보면 특기할 사항이 한우리/C 드라이버는 질의문을 효율적으로 처리하고자 여러 개의 데이터베이스 객체들을 조인(Join)함으로써 마스터/디테일관계(Master/Detail Relationship)을 설정하는 기능을 지니고 있는 점이다. (그림 7)을 좀 더 설명하면, 상단은 데이터베이스 객체들 중에서 테이블 객체를 사용하고 있는데, 테이블 객체는 대량의 데이터를 한번에 표현하는 데 유용하다. 하단은 리스트 객체를 사용하고 있고, 리스트 객체는 데이터를 하나의 레코드(또는 튜플) 단위로 표현하는 데 유용하다.

상단의 테이블에 하나의 레코드를 선택한 후, "Detail Information"버튼을 선택하면 해당 레코드에 대한 상세 정보가 하단에 표현된다.

### 5. 결 론

본 논문에서는 분산 처리 기술과 클라이언트-서버 기술이 접목되었고, 멀티미디어 처리 기능을 지니고 있는 제 4세대 언어 형태의 클라이언트-서버 도구인 한우리/C의 미들웨어의 설계 및 구현된 내용에 대하여 설명하였다. SQL 문장 처리 기능 뿐만 아니라 하이퍼미디어 처리 기능과 분산 트랜잭션 처리 기능을 포함하고 있는 한우리/C는 패스트 프로토타이핑(Fast Prototyping) 모델을 지니고 객체 지향형 프로그래밍 기법을 적용시켜 구현되었다. 이러한 특성을 지닌 한우리/C를 기존의 유사한 기능을 지닌 클라이언트-서버 응용 개발 도구들과 비교해 보았다. 본 논문에서는 한우리/C와 가장 유사한 도구인 Powerbuilder와 Objectview에 대하여 기능적인 분석을 한 결과[6,7,8], Objectview는 응용 사용자 보다는 응용 개발자인 전문가 위주의 도구이



(그림 8) 한우리/C의 응용 예: 홈쇼핑  
(Fig. 8) Example of Hanuri/C Application: Homeshopping

드로 사용자 인터페이스 기능이 상당히 용이하지 못하며 다양한 그래픽 기능을 지원하지 못할 뿐만 아니라, 객체 지향적인 방법을 지원하고 있지 못하다. Powerbuilder는 그래픽 기능, ODBC 접속 기능, 객체 지향적인 개발 기법 제공하고 응용 개발자 뿐만 아니라 응용 사용자들에게도 사용이 편리한 기능을 제공하는 등 강력한 도구이지만, 한우리/C의 미들웨어 기능 측면에서는 부족하다. 한우리/C는 크게 4계층의 참조 모델을 지니고 있다. 즉, 가장 사용자와 가장 밀접한 기능들을 지닌 모듈들로 구성된 한우리/C 사용자웨어 계층, 한우리/C의 가장 핵심 모듈들을 지니고 있는 한우리/C 미들웨어 계층, 제 3자 업체들이 제공하는 소프트웨어들이 내장되어 있는 한우리/C 엔터프라이즈웨어 계층, 그리고 가장 알단 계층으로서 통신 망이나 장치 드라이버들이 들어 있는 한우리/C 하드웨어 계층이다. 본 논문은 이들 중에서 한우리/C의 가장 핵심 계층이 되는 한우리/C 미들웨어를 대상으로 설명하였다. 한우리/C 미들웨어는 데이터 관리 기능, 통신 관리 기능, 객체 관리 기능을 지니고 서비스를 제공한다. 이들 한우리/C 미들웨어의 기능들 중에서 본 논문에서는 데이터 관리 기능과 통신 관리 기능을 지닌 한우리/C 드라이버 모듈에 대한 설계와 구현에 대하여 설명하였다. 그림 2에서 보듯이 한우리/C 미들웨어는 논문에서 제안한 한우리/C 드라이버 모듈 외에 다양한 DBMS를 지원할 수 있는 ODBC 접속 모듈, 분산 처리 환경을 지원하는 DCE 접속 모듈, DTP 접속 모듈로 확장 중이다. 구현 완료한 한우리/C 1.0을 이용하여 (그림 8)에서 보듯이, 지역 특산물 판매와 문화예술 입장권 예매를 대상으로 홈쇼핑 응용 프로그램을 개발하여 한우리/C가 홈쇼핑 서비스에 사용될 수 있음을 보였다[11].

**참 고 문 헌**

[ 1 ] A. Umar, "Distributed Computing A Practical Synthesis," pp. 229-289, Prentice-Hall, New Jersey, 1993.  
 [ 2 ] C. Chappell, C. Guilfoyle and J. Hewett, "Client-server Computing:commercial st-

ategies," pp. 27-56, Ovum, London, 1991.  
 [ 3 ] Dewire, "Client/Server Computing," pp. 29-40, McGraw-Hill, U.S.A., 1993.  
 [ 4 ] OSF, "Distributed Computing Environment:An overview," pp. 1-1-1-33, OSF, Cambridge, MA.,Jan. 1992.  
 [ 5 ] Al Williams, "OLE 2.0 and DDE Distilled A programmer's Crash Course," pp. 29-37, pp. 95-116, Addison Wesley, U.S.A., 1994.  
 [ 6 ] PowerSoft, "Introducing PowerBuilder," pp. 11-39, PowerSoft, Burlington, MA., 1992.  
 [ 7 ] 격주간 하이테크정보, No. 122(pp. 80-90), No. 123(pp. 90-115), No. 124(pp. 66-76).  
 [ 8 ] Ovum evaluates, 4GLS and Client-server Tools. pp. 15-19, Ovum, London, May 1994.  
 [ 9 ] Paul E. Renaud, "INTRODUCTION TO CLIENT/SERVER SYSTEMS:A Practical Guide for Systems Professionals", Wiley, New York, 1993.  
 [10] 김명준, et al., "분산 처리 시스템 소프트웨어 기술," 컴퓨터 기술지 Vol.8, No.1, pp. 61-73, 대한전자공학회 전자계산연구회, Dec. 1994.  
 [11] Soran Ine, et al, "The Distributed Client-Server Tool for Developing the Homeshopping Service," 1st international Workshop on High Speed Network and Open Distributed Platform, Jun. 1995.

**임 채 덕**



1989년 전남대학교 전산통계학과 졸업  
 1989년~현재 한국전자통신연구소 연구원  
 관심 분야:분산 컴퓨팅, 정보통신, 브레인 컴퓨팅