

□ 기술해설 □

클라이언트/서버 환경 구축과 활용 1

명지대학교 이 기 현*

● 목	차 ●
1. 서 론	2.4 클라이언트/서버의 궁극 목표
1.1 클라이언트/서버의 등장배경	3. 클라이언트/서버의 환경 구축
1.2 앞으로의 쟁점들	3.1 클라이언트/서버의 수행단계
2. 클라이언트/서버	3.3 클라이언트/서버 도구 사용
2.1 클라이언트	3.4 클라이언트/서버 응용작성 단계
2.2 서버	3.5 데이터베이스 환경
2.3 클라이언트/서버의 기능	

1. 서 언

1.1 클라이언트/서버의 등장 배경

클라이언트/서버의 등장은 크게 보아 세계적으로 선진국을 중심으로 일어나고 있는 정보시스템 환경의 변화라는 두가지 측면에서의 일대 변화로부터 연유된 현상으로 볼 수 있다.

최근, 전세계적인 경기침체와 국경없는 무역으로 무한 경쟁에 돌입하게 된 각국의 기업들은 생존을 위하여 다음과 같은 변화에 따르지 않으면 안되게 되었다.

가. 기업경영 여건의 변화

- (1) 경영의 국제화(Globalization)
- (2) 범세계적 정보획득과 상품경쟁
- (3) 제품과 서비스의 라이프 사이클 단축
- (4) 비용절감 필요성
- (5) 의사결정 속도와 질향상 필요성
- (6) 시스템 응용과 통신망 등의 통합관리에 대한 고객의 요구

나. 정보시스템 환경의 변화

- (1) 하드웨어의 수준향상과 가격의 상대적 하락
- (2) 사용자 중심의 다양한 소프트웨어
- (3) 망 접속의 확대
- (4) 정보와 시스템의 공유
- (5) 자료의 복잡성 증대
- (6) 세계 표준의 확산

1.2 앞으로의 쟁점들

다음에 열거한 사항들은 클라이언트/서버 개념의 등장 당위성과 앞으로 이 분야의 추이를 예견하게 하는 90년대의 주요 쟁점들이다.

- 정보시스템(IS) 서비스의 비용
- 정보시스템(IS)의 복잡성
- 분산된 정보자원의 관리와 연결
- 온라인 거래처리(OLTP)의 증가
- 중요업무 응용 프로그램(Mission-Critical Application Program)[1]의 개발과 유지관리

* 중신회원

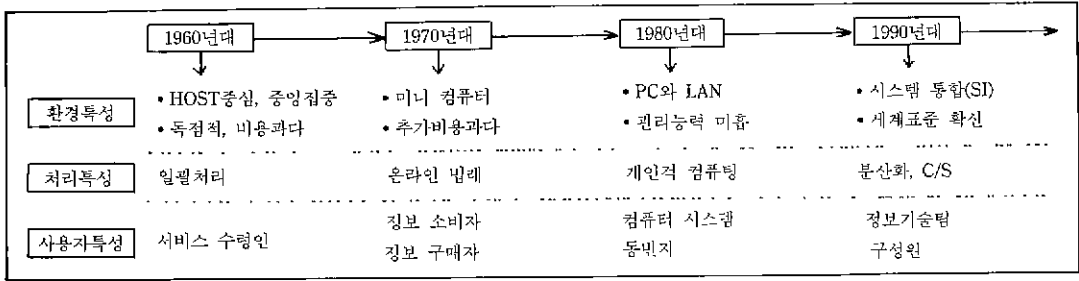


그림 1-1 정보시스템의 발전추이

표 1-1 형태의 변화

대상항목	구분기준	80년대 이전	80년대 이후
<ul style="list-style-type: none"> 조직 권한/책임 시스템 형태 컴퓨터 S/W 판매자 제품 기술/구조 사용자 인터페이스 정보형식 사용자 기술·응용 		수직형 중앙집권 센터중심형 (호스트 중심) 메인프레임 판매자 의존 동종균질 판매자 제품 의존 문자 문자기호 운영자 중심 기술위주	수평형 부문별 분산 클라이언트/서버형 (망 중심) 워크스테이션 판매자 탈피 멀티벤더 개방형 표준 그래픽 텍스트, 이미지, 음성 고객 중심 응용위주

새로운
시대
예고

- 의사결정지원 시스템과 집행정보 시스템 사용의 증가
- 방법론의 표준화
- 사업 해결책의 구현

2. 클라이언트/서버

클라이언트/서버는 망(Network)을 통하여 응용(Application)을 분담처리하는 컴퓨팅 기술의 한 형태로서 다음과 같은 특성을 갖는다.

2.1 클라이언트(Client)

가. 클라이언트는 단일 사용자 워크스테이션(Single-User Workstation)으로서의 서비스와 적절한 컴퓨팅과 연결, 데이터베이스 서비스와 비즈니스 요구에 관련된 인터페

이스를 제공한다.

나. 클라이언트는 서버에게 서비스를 요청하고 그로부터 서비스를 받는다.

다. 단순한 dumb 터미날이 아니며, 자체의 CPU, 기억장치와 고성능의 그래픽스 기능을 갖는 완전한 컴퓨터 시스템으로 근거리 또는 원격지의 디스크를 사용한다.

라. 서버의 지원을 받아 엔드 유저(End-User) 컴퓨팅을 실현시키는 최종사용자 지향적 시스템이다.

마. 단순히 LAN상에 연결된 PC가 아니다. 따라서 바람직한 클라이언트시스템을 위하여는 다음과 같은 사항들이 요구된다.

- 최종사용자의 요구를 만족시킬 것
- 생산성 증대를 위하여 GUI(Graphical User Interface), 4GL, 멀티미디어, 객체지향적 기법 등을 구사할 수 있을 것

- 국제 표준을 따르고 이식성과 호환성이 우수한 개방형 시스템 지향적일 것

2.2 서버(Server)

서버는 공유 기억장치를 갖는 하나 또는 그 이상의 이중 사용자 프로세서로서 데이터베이스와 화일관리를 위한 중앙의 자원을 보유하고 컴퓨팅, 데이터베이스 서비스 등을 제공하며 클라이언트의 서비스 요청을 받고 이를 수행한다.

따라서 서버 시스템에서 요구되는 사항은 다음과 같다.

- 클라이언트에 만족한 응답을 제공할 수 있을 것
- 국제표준에 따르며 호환성이 우수한 개방형 시스템 지향적일 것
- 자료 전달의 효율성이 보장되는 투명성을 가질 것
- 확장성이 있을 것

서버는 보는 관점에 따라 여러가지로 구분할 수 있으나 크게 두가지로 나누어 볼 수 있다. 하나는 행정처리기능 단위에 따른 적용업무별 관점과 다른 하나는 정보처리 기능별 관점에서 본 요소별 구분이다[2].

가. 적용업무별 요소

- 기업차원 서버(본부)
- 부서 서버(사업단위)
- 기능 서버(부서별 단위)
- 지역 서버(지역단위)

나. 정보처리 기능별 요소

- 터미널 서버
- 이미지 서버
- 프린터 서버
- 영상 서버
- 파일 서버
- 팩스(FAX) 서버
- 디스크 서버
- 계산 서버
- 데이터 서버
- 통신 서버
- 데이터베이스 서버

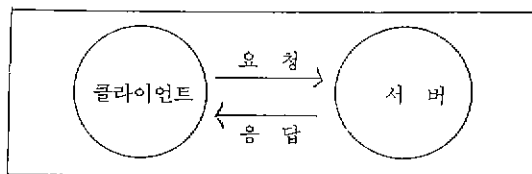


그림 2-1 클라이언트/서버 관계

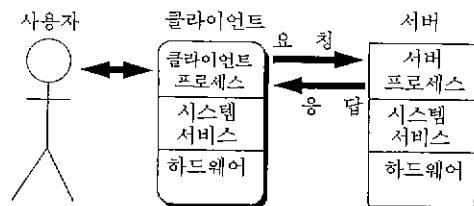


그림 2-2 클라이언트/서버 시스템[3]

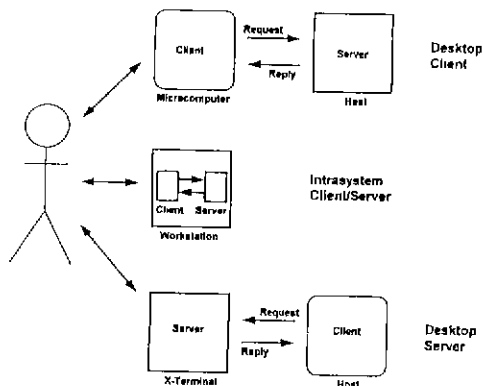


그림 2-3 서로 다른 유형의 클라이언트/서버 시스템[4]

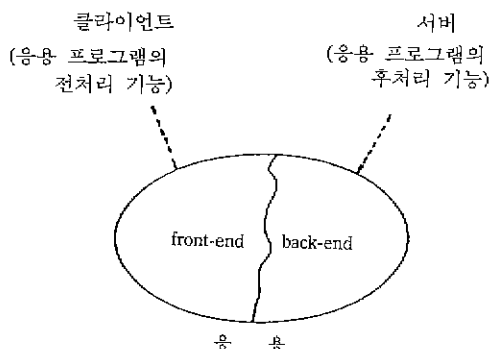


그림 2-4 클라이언트 서버 기본 구조

- 거래(Transaction) 서버
- 우편 서버

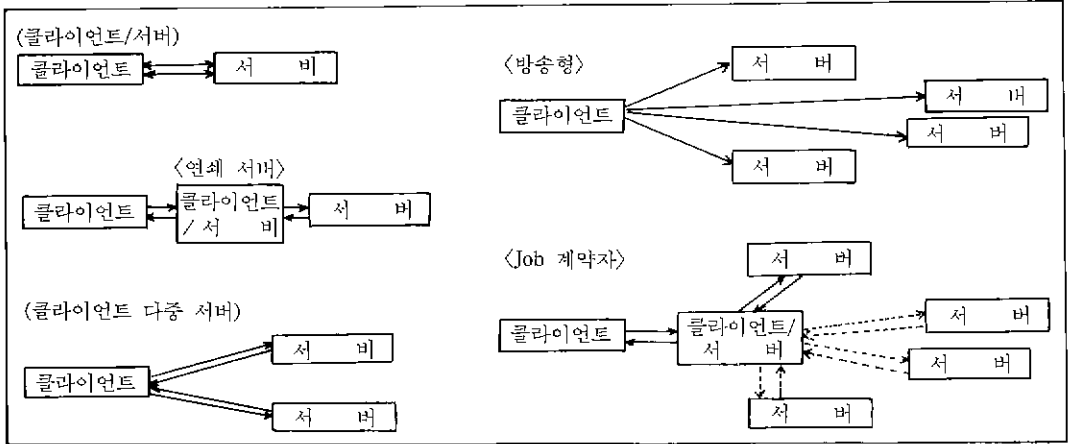


그림 2-5 클라이언트/서버의 여러모델[5]

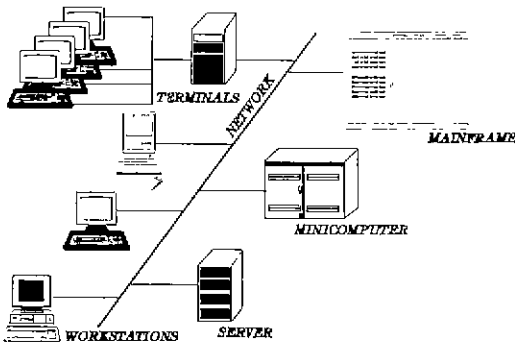


그림 2-6 최근의 클라이언트/서버 구성 형태[6]

- 응용(Application) 서버
- 음성 서버
- 처리 통제 서버

2.3 클라이언트/서버의 기능

클라이언트/서버의 기능은 그림 2-7에서와 같이 신뢰성이 우수한 초고속의 전송속도를 갖는 망을 통하여 클라이언트의 요청에 서버가 적절한 서비스로 응답하고, 자원관리 등 제반관리를 담당한다.

2.4 클라이언트/서버의 궁극 목표

클라이언트/서버 시스템을 구축하는 목적은

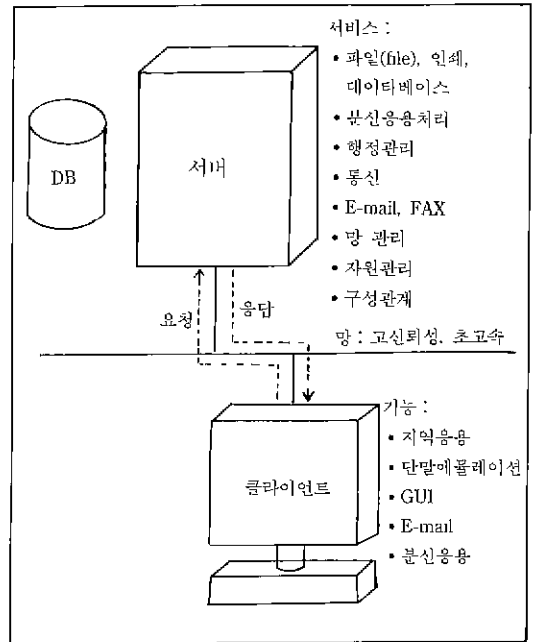


그림 2-7 클라이언트/서버 기능

여러가지를 들 수 있겠으나 대개 다음과 같은 사항들로 압축할 수 있다.

- 시스템 투자의 절감
- 유연한 시스템 구축
- 컴퓨팅 자원의 효율적 사용
- 기능형 분산시스템의 실현
- 최종사용자 위주의 시스템 구축

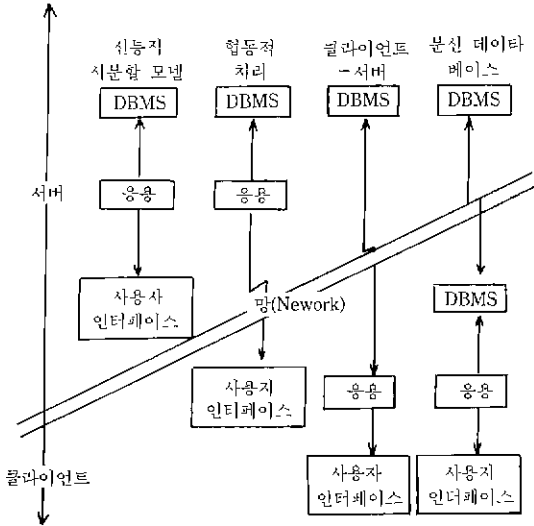


그림 2-8 응용구조의 발전

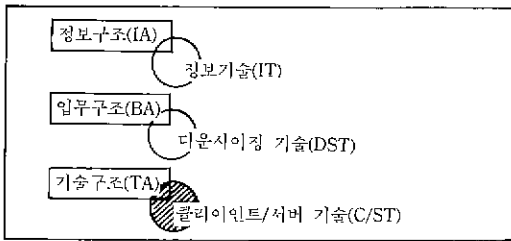


그림 2-9 구조에서 파생되는 기술

3. 클라이언트/서버 환경 구축

클라이언트/서버 환경 구축을 위하여는 클라이언트/서버 컴퓨팅 환경을 크게 몇가지 범주로 나누고, 이를 위한 설계와 그보다 앞서 수행계획을 세우는 것이 능률적이다.

이 글에서는 환경의 범주를 데이터베이스 환경, 분산처리 환경, 개방접속 환경 등으로 나누고 이를 실행하기 위한 사전 수행단계들을 정리해 본다.

3.1 클라이언트/서버의 수행 단계

그림 3-1 의 클라이언트/서버로의 단계별 진행순서는 필자가 임의로 소프트웨어 수명주기(Life cycle)에 맞춰 단계별로 계획을 세워 본 것이다.

그림 3-2 에서 업자선정은 그 중요성을 생각하여 의식적으로 삽입시켰다. 현재로서는 사용자나 시스템 공급자가 모두 경험의 부족 상태에서 클라이언트/서버 시스템 구축에 대한 성급한 의욕이 앞서서 자칫 필요 이상의 시행착오를 범할 수 있으므로 이와같은 단계별 진행순서를 정하고, “돌다리도 두드려 건너다”는 철저한 개발 자세가 필요하다.

특히 망 구축에는 여러종류의 제품과 프로토콜들이 쓰이므로 전문적인 지식과 경험이 필요하다. 그러나 사용자들은 스스로 해결할 능력이 미흡하므로 이 분야의 전문적 컨설턴트의 자문(consulting)을 받을 필요가 있다.

컨설턴트의 유형은 전문성과 그 규모와 수준에 따라 다음과 같이 여러가지로 나누어 볼 수 있으며, 각각에 대한 자세한 비교 설명은 생략한다.

- Contract Programming Houses
- Management Consultants
- Outsourcing Firms
- System Integration Houses
- Downsizing Specialists
- LAN System Consultants
- Telecommunications Consultants
- Training Houses

사용자들은 필요에 따라 적합한 컨설턴트를 선정하는 것이 바람직 하다.

그림 3-3에서 UE(Usability Engineering)부분은 시스템 개요 설계와 상세설계의 두 부분의 설계시에 모두 사용시의 능률성과 관리의 효율성을 유념하여 설계하지 않으면 안됨을 나타낸 것으로 공학적 공정과 기술이 요구되는 부분이다.

3.2 클라이언트/서버 환경 구축을 위한 설계

3.2.1 설계시 고려사항

설계시 유념할 사항은 개발자 측면과 관리자 측면의 두가지 측면에서 다음과 같은 사항들을 들 수 있다.

- Pioneer Q+E
 - VisionWare SQL Retriever
 - Information Builders EDA/SQL
- } 중간매체
S/W의 예

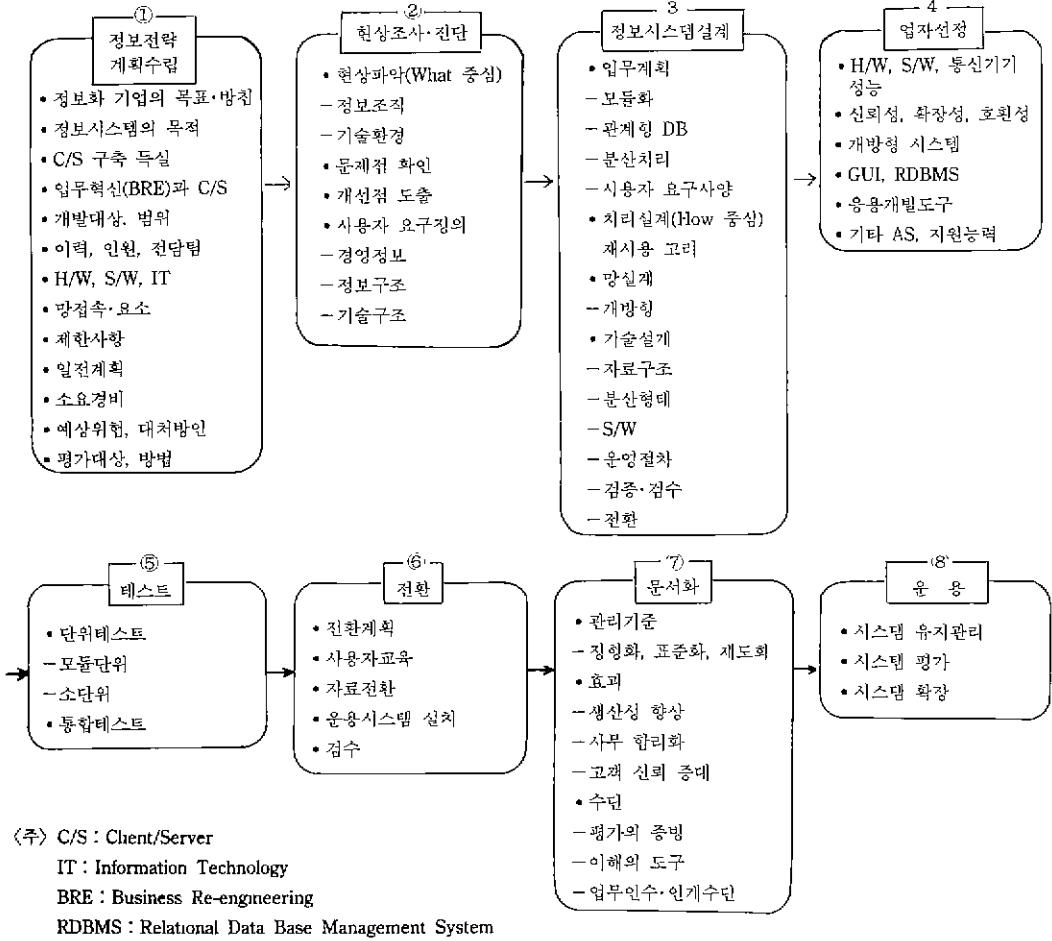


그림 3-1 클라이언트/서버로의 단계별 진행순서

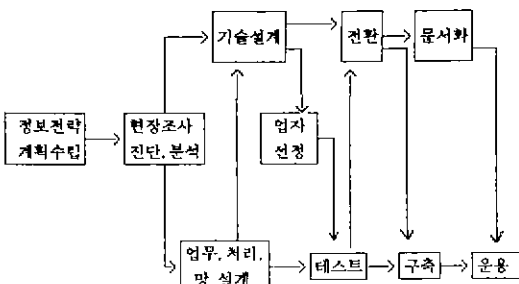


그림 3-2 단계별 공정 흐름도

관리를 말한다.

표 3-1의 관리자측면 가운데 구성관리 기능은 OSI 망 관리 기능 5가지(성능관리, 장애관리, 구성관리, 보안관리, 계정관리) 중 하나인 구성

효율적인 클라이언트/서버 환경을 구축하기 위하여는 먼저 시스템 구조 설계(Client-Server Architectural Strategy : CASE)¹⁾가 필수적이다. 이 부분이 잘못되면 실패할 수도 있다.

3.2.2 GUI설계[11]

가. GUI설계 지침

사용자 설계는 사용성 공학(Usability Engineering : UE)에 입각한 특정 공정이자 할 수 있으며 GUI 설계는 특정의 공학공정과 기술을 필요로 한다.

GUI설계를 위한 일반적인 지침으로서의 사용자 인터페이스 설계를 위한 두가지 기본원리는

1) Patrick Smith, 캐나다 SHL 시스템 하우스 수석부회장

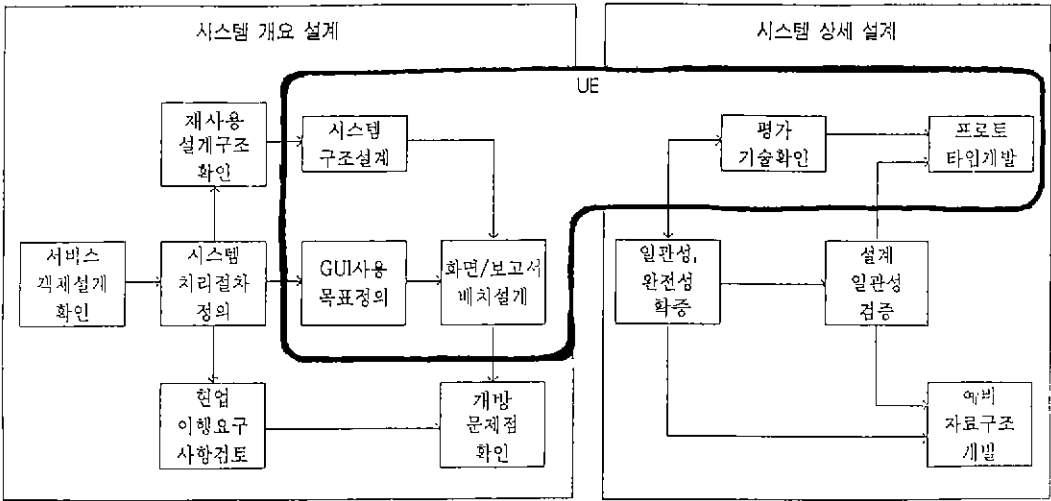


그림 3-3 공정구조 내에서의 사용성 공학(UE)[9]

표 3-1 설계시 고려 사항

개발자 측면	관리자 측면
<ul style="list-style-type: none"> - 업무 요구사항 - 성능 - 중간매체S/W(Middle-ware)* 의 기능 - 시스템 하부구조 - 보안성 - 관리의 용이성 	<ul style="list-style-type: none"> - S/W 분산 - 구성관리(Configuration management)기능 - 안정성 - 망과 시스템 관리기능

첫째, 사용자가 정확한 개념모형을 개발할 수 있도록 지원하고, 둘째, 사용자가 응용을 통제할 수 있도록 설계해야 한다. 다시말하여 사용자 위주의 인터페이스를 설계해야 하고 은유법(Metaphor)사용이 가능하도록 하며, 일관성을 갖도록 하고, 사용자로 하여금 모드(Mode) 사용의 절제를 통하여 사용자 통제권을 갖도록 설계해야 한다. 그밖에 사용자 인터페이스를 명료하게 해야 하며 또한 메시지 활용이 가능하도록 설계해야 한다. 그리고 다양한 수준의 사용자들이 사용할 수 있도록 하며 즉각적인 반응을 제공할 수 있도록 설계해야 한다.

나. 대화(Dialog)설계 지침

대화설계를 위한 주요한 지침으로는 다음의 8가지 사항을 들 수 있다.

(가) 일관성 유지 노력

- (나) 숙달된 사용자에게는 지름길을 택할 수 있도록 함
- (다) 행동(Action)에 대한 귀환(Feedback)제공
- (라) 항상 끝맺음을 할 수 있도록 하는 기능 제공
- (마) 간단한 오류처리 기능 제공
- (바) 원상복귀 기능 제공
- (사) 내부통제 기능 지원
- (아) 기억부담을 해소할 수 있도록 단순한 설계

다. 윈도우(Window) 설계 지침

(1) 윈도우 초기 크기

사용자가 작업할 응용(Application)을 수용할 수 있도록 충분한 크기로 한다.

(2) 형식화(Formatting)와 전단(Clipping)

윈도우의 모양이나 크기가 변할 경우에는 화면상의 정보를 재형식화(Reformatting) 하기보다는 잘라내도록 한다.

(3) 응용의 갯수

- 최대한 5~9개 사이의 대상(Object)을 수용한다.
- 정보가 잘 집단화(Grouping)된 경우보다 많은 정보도 수용할 수 있도록 한다.

(4) 색(Color)의 사용

- 너무 많은 색의 사용은 사용자를 혼란하게 하므로 색의 사용을 절제한다.

- 제정한 색에 대해서는 문화적 의미를 파악하도록 한다.
- 사용자가 색을 바꿀 수 있도록 한다.

라. 자료 출력(Data Display) 설계 지침

(1) 자료 출력의 목표

자료출력의 일관성을 유지하며 자료의 배열은 사용자에게 친숙한 형태로 하고 사용자의 기억 부담을 최소화 하도록 한다. 자료입력과 호환성을 유지하도록 하고 사용자가 편리한 형태로 정보를 볼 수 있도록 사용자에게 통제권을 부여한다.

(2) 자료 출력의 지침

표식(Label)과 도형작성의 습관을 일관성 있게 하고, 약어를 사용 할 때는 표준화하여 사용하도록 한다.

모든 화면은 일관성 있는 형태를 유지하도록 하며, 각 화면은 페이지 번호를 표시하고 그에 따라 검색할 수 있는 기능을 부여한다.

숫자정보의 사용은 그 숫자가 특정한 의미가 있을 경우에만 사용하고, 도형에 의한 정보제공을 많이 활용한다.

마. 자료 입력설계 지침

(1) 자료 입력 처리의 일관성

동일한 작업 순서와 동일한 약어, 동일한 한정자(Delimiter)를 사용한다.

(2) 사용자 입력의 최소화

단일 키 입력이나 목록선택 방식 등 운영방법을 개선하고 중복 자료의 입력을 피한다.

(3) 사용자 부담 감소

(4) 자료 입출력의 호환성 유지

(5) 자료 입력에 대한 사용자 통제의 융통성 부여 사용자가 스스로 통제할 수 있는 순서로 정보를 입력할 수 있도록 설계한다.

3.3 클라이언트/서버 도구 사용

클라이언트/서버 도구라 함은 클라이언트/서버를 위한 분석·설계를 지원하고, GUI설계수단을 제공하며, 다수의 플랫폼(Platforms)을 위한 데이터베이스와 코드를 생성하고, 플랫폼 간의 공유를 촉진하는 기능을 가진 개발도구라 할 수 있다. 잘 알려진 도구로는 다음과 같은 것들을

들 수 있다.

- Screen Scraper:

UI 전처리(front end)시스템

- I-CASE:

• 통합된 전수명주기(Full Life-Cycle) CASE

• 분석공학(Reverse Engineering) 기능 제공

- Back End CASE

• 응용생성기(Application Generator)

• 분석공학 기능

이러한 클라이언트/서버 도구들을 분석과 설계과정에서 사용함으로써 개발효율을 향상시킬 수 있다.

개발도구들을 해당 분야별로 나누어 좀 더 구체적으로 살펴보면 다음과 같다.

3.3.1 하드웨어

서버로는 Intel의 i386, i486 또는 i586 기종을 사용하거나 RISC칩을 사용한 기종이나, 멀티태스킹을 할 수 있는 OS/2나 UNIX기종, 미니컴퓨터나 메인프레임을 사용한다. Business Research Group의 조사에 의하면 OS/2(8 Mbyte RAM)에 망 소프트웨어를 얹은 Intel i386 이나 i486기종, 또는 UNIX를 얹은 미니 컴퓨터를 서버로 많이 사용 하며, DOS Version 3.2(640 Kbyte RAM)이나 OS/2 Version 1.1의 Intel i286, i386, i486을 클라이언트로 많이 사용한다. 또한 Compaq System PRO(16 Mbytes)의 시스템에 LAN Manager와 함께 SQL Server database를 이용하기도 한다. 근래들어 클라이언트/서버 환경을 위해 새로운 유형의 “수퍼서버(Super Server)”들이 시장에 나타나와 있는데, 이 “수퍼서버” 시스템의 기본적 사양은 다음과 같다(표 3-2).

- chip: MIPS, Intel, Sparc, RS/6000

- 32-bit/64-bit parity-protected bus

- 32 Mbytes 이상의 RAM memory

- Dual CPU

- 관계형 데이터베이스 시스템(RDBMS)

- UNIX, SCO UNIX, Microsoft의 SQL, OS/2, LAN Manager, Netware

- Fault-tolerant 기능

표 3-2 마이크로 서버와 슈퍼서버[10]

	Micro/servers		Superserver	
	Low	high	Low	High
Users	2-16	16-64-64	64-128	128-1000
Price	\$5,000	\$15,000	\$25,000-50,000	\$60,000-300,000
Chip	386	2×386-486	Mips, Intel, Sparc	Mips, Intel, Sparc, RS/6000
RAM	8-16 Mbytes	16-64 Mbytes	32-128 Mbytes	64-256 Mbytes
Disk	80-180 Mbytes	180-840 Mbytes	640 Mbytes-2 Gbytes	1-32 Gbytes
OS	DOS, OS/2, NetWare	OS/2 UNIX, NetWare	UNIX, SCO UNIX	UNIX, SCO UNIX

Source:Computer Associates International Inc.

표 3-4 클라이언트/서버를 위한 선택[12]

GUI	클라이언트 OS	서버 OS	망 OS	거래 관리자 (Transaction Manager)	DBMS	TOOLS	
*WINDOWS	*DOS	*OS/2	*NETWARE	*TUXEDO	*SYBASE	*VISUAL BASIC	*FOUNDATION /CP
• PM	• OS/2	*NETWARE-ENLM	• LAN MANAGER	• TOP END	*SQL SERVER	• ACCESS	• OBJECT VISION
• MOTIF	*NT	*NT	• VINES	• CICS	*ORACLE	• SQL WINDOWS	• SMALL TALK
• MAC	• UNLX	*SOLARIS	• LAN SERVER	• ENCINA	• GUPTA	• ELLIPE	• PARTS
• OPENLOOK	• MAC	*HP UX	*TCP/IP	*ELLIPSE/PS	• INFORMIX	• PROGRESS	• ENFIN
• X-WINDOWS		*ALX			• INGRES	• POWERBUILDER	• ACCCELL
						• JAM	

*표는 미국에서 다른 것들 보다 더 많이 사용되는 것임

- 확장성

- Auto reboot/Error logging

또한 UNIX를 탑재한 서버용 미니 컴퓨터들이 컴퓨터 업체마다 많이 발표되고 있는 것도 최근의 시장 동향이다. 그리고 PC보다는 UNIX를 탑재한 미니 컴퓨터가 서버용으로는 더 안정되었다는 인식하에 서버용 미니 컴퓨터의 활용이 크게 증대되고 있다.

3.3.2 소프트웨어

클라이언트/서버 환경을 구축하기 위해 개방용

도구들이나 응용 프로그램들도 많이 발표되었다. 이들 개발용 도구들은 다음과 같이 세가지로 나눌 수 있다.

가. 서버 데이터베이스

데이터베이스는 클라이언트/서버 환경에서 가장 중요한 요소이다. 이 분야에서는 SQL(Structured Query Language)을 많이 사용한다. 왜냐하면 SQL database는 물리적으로 데이터를 응용에서 분리시켜주며 융통성을 제공하기 때문이다.

표 3-3 클라이언트/서버 도구 비교

Function	Mozart	View Point	EASEL Wrkbnch	Power Builer	Guta Family	Ellipse
Development Environment						
DOS	Y	N	N	N	N	N
MS Windows	Y	Y	N	Y	Y	N
OS/2	N	N	Y	N	Y	Y
Unix System 5.4	N	N	N	N	N	P
Runtime Environment						
DOS	Y	N	Y	N	N	N
MS Windows	Y	Y	Y	Y	Y	Y
OS/2	Y	N	Y	N	Y	Y
Unix System 5.4	N	N	N	N	N	P
Development Functionality						
Smart Display Objects	P	Y	N	P	P	Y
Smart SQL Interface	N	P	N	Y	P	N
Smart OLTP Objects	N	N	N	N	N	Y
High-Level Language	Y	Y	Y	Y	Y	Y
C Language interface	Y	Y	Y	Y	Y	Y
Interactive Debugger	Y	N	Y	Y	Y	Y
Object-Oriented	Y	P	P	Y	Y	Y
Repository an a LAN	N	N	Y	Y	Y	Y
Portable to Other Env.	Y	N	Y	P	-	Y
Captures 3270 Screens	Y	Y	Y	N	N	N
Scripting Language	Y	Y	Y	Y	Y	Y
Bit-Mapped Graphics	Y	Y	Y	Y	Y	N
Database Support						
Sybase/MS SQL Server	Y	Y	Y	Y	Y	Y
ORACLE	Y	N	Y	Y	N	Y
OS/2 DBM	Y	N	Y	N	Y	Beta
Gupta SOLBase	N	Y	N	Y	Y	N
Mainframe DB2	N	Y	3rd Pty	Y	Y	3rd Pty
VSAM	N	N	3rd Pty	Y	Y	3rd Pty
Transaction Processing						
Security	P	P	P	P	P	P
Restart/Rollhack(Rebuild	N	N	N	N	N	Y
Transactions Across Multiple LANs and servers						
Communications						
3270 HLLAPI/EHLLAPI	Y	Y	Y	Y	N	N
IBMA/PCOX	Y	Y	Y	Y	N	N
Novell Token Ring	Y	Y	Y	Y	Y	Y
APPC/LU6.2	N	Y	Y	Y	Y	Y
Asynchronous(VT100)	Y	Y	Y	N	3rd Pty	N
Product Cost						
Development Workstation	\$8K	\$9.5	\$12K	\$3.4K	\$3K	\$15K(2)
Run-Time Component	\$500	\$395	\$350	\$995	0	\$3495(10)

per server

(*) Y : Yes N : No P : Partial

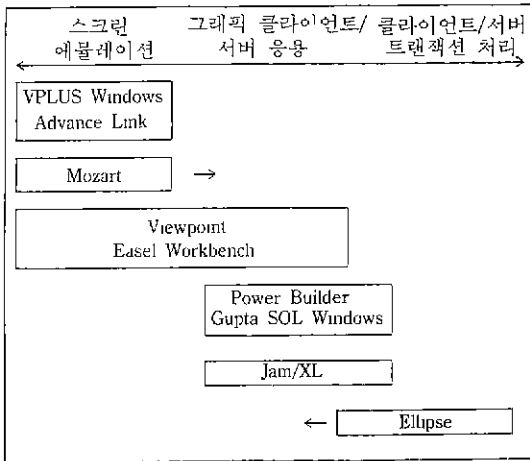


그림 3-5 C/S 응용 작성 단계

클라이언트/서버 시장을 위해 나온 데이터베이스는 다음과 같은 것들이 있다. ()내는 개발 또는 공급 회사이다.

- Ingres distributed relational DBMS(Ask Computer)
- Rdb relational DBMS for VMS(Digital Equipment)
- SQL Base distributed RDBMS(Gupta Technologies)
- Extended Edition Database Manager(IBM (Armonk, NY))
- Focus 4GL/DBMS, EDA/SQL(Information Builders)
- SQL Server multiuser PC-/MS-DOS DBMS(Microsoft)
- Oracle Server(Oracle)
- Sybase SQL Server(Sybase)
- XDB Server, SQL database server(XDB)
- ALLBASE/SQL(HP)
- UniSQL(UniSQL Inc.)

나. Computer-Aided Software Engineering (CASE)

근래들어 Software Engineering과 Reengineering이 클라이언트/서버 환경을 구축하는 큰 장애점이 되고 있다. 특히 Reengineering에서는 기존 응용을 새로운 환경으로 옮길 때 소프트웨어를 평가하고 수정하는 작업도 지원해야만 한

다.

이 클라이언트/서버 환경을 지원하는 CASE 도구로는 다음과 같은 것들이 있다.

- Viewpoint : an object-oriented, event-driven application development environment (Knowledge Ware)
- PACLAN : OS/2 based server software (CGI System)
- Ingres : application development tool(Ingres)
- InFront/DS 4GL : for converting existing CICS application to client/server environment(Multi Soft)
- System Architect(Popkin)

다. Graphical User Intergface(GUI)

GUI는 눈으로 보기에만 좋은 것이 아니라 클라이언트/서버 환경에서는 필수적인 것이 되고 있다. 왜냐하면 여러가지 다른 소프트웨어나 하드웨어를 사용자가 사용하기 쉽게 통합해야 하기 때문이다.

클라이언트/서버 환경을 지원하는 object-oriented GUI 개발 도구로는 다음과 같은 것들이 있다.

- Toolbook object-oriented programming language(Asymetrix)
- Omni 5(Blyth Software)
- CASE : W and CASE : PM(Caseworks)
- Easel GUI development tool(Easel)
- Mozart GUI(Mozart)
- Object store(Object Design)
- Open Interface(Neuron Data)
- Power Builder(Power Soft)
- Gupta SQL Windows(Gupta Technologies)
- Extensible Vitual Toolkit :GUI programming tools(XVT Software)

3.3.3 응용

클라이언트 응용 GUI, 프로시저 로직, 호스트 컴퓨터나 파일 서버의 자료에 접근하는 방법 등을 제공하며, MS-Windows나 OS/2 환경하에서 완벽한 그래픽 디스플레이를 표현할 수 있게 해준다. 또한 하나의 그래픽 화면하에서 여러 서

버나 호스트 컴퓨터들에게 데이터를 요청하고 보여주는 front-end 윈도우 기능을 수행할 수 있어야 하며, Windows 3.1의 OLE(Object Linking and Embedding)과 DDE(Dynamic Data Exchange) 기법을 이용하여 동시에 여러 클라이언트 응용간에 자료를 공유할 수 있어야 한다.

데이터베이스, 네트워크 관리, 시스템 응용, 시스템 관리, 예비조치, 복구, 보안 등의 기능은 OS/2나 UNIX를 탑재한 LAN파일 서버나 호스트 컴퓨터에서 운용된다.

대부분의 클라이언트/서버 응용은 SQL 인터페이스를 가진 서버용 관계형 데이터베이스 관리 시스템을 사용하며 SQL Server(Sybase/Microsoft), ORACLE(Oracle Corp.), OS/2 Database Manager(IBM), INFORMIX(Informix Software, Inc.), Rdb(Digital), INGRES(Ask Computer), ALLBASE(Hewlett Packard), HP ALLBASE/SQL PC API 등이 있다.

3.3.4 클라이언트/서버 시스템 관리

클라이언트/서버 시스템에서 주의할 것은 소프트웨어 특허와 망 소프트웨어나 하드웨어 구입, 망 문제해결 등 망 관련 문제나 사용자 계정을 유지 관리 하는 일이 번거롭다는 것이다. 또한 서로 다른 다수의 시스템과 운영체제, 소프트웨어가 전사적으로 운용 되므로 이들을 효과적으로 묶어 관리할 수 있는 도구들이 요구된다.

이러한 도구들은 다음과 같은 것들이 있다.

- Auspex Performance Monitor(Auspex Systems)
- Ellipse : client/server transaction processing(Cooperative Solutions)
- LANStream : PC-DOS-based client server storage management system(Maynard Electronics)
- Microsoft LAN Manager(Microsoft)
- LANalyzer : for PC-DOS network analysis
- LANtern : PC-DOS network performance monitor(Novell)
- LANcentral : network monitoring(Racal Interlan)

다음과 같은 도구들은 데이터베이스나 LAN 보안 도구로 쓰인다.

- CertusLAN, CertusPC(Certus International)
- Top Secret and ACF2 object-oriented security and ProSecure for DB2(Computer Associates International)

3.3.5 클라이언트/서버 도구

- Mozart(Mozart System Corporation)
- Viewpoint(Viewpoint Systems)
- Easel Workbench(Easel Corporation)
- Gupta SQL Windows(Gupta Technologies, Inc.)
- Power Builder(Powersoft Corp.)
- Ellipse(Cooperative Solutions, Inc.)
- VPLUS/Windows, Advance Link, Jam/XI (Hewlett Packard)

3.4 클라이언트/서버 응용작성 단계

클라이언트/서버 응용의 작성은 일반적으로 3 단계로 나누며 각 단계는 다음과 같다.

<1단계> 스크린 에뮬레이션 단계:

- 가장 간단하나 제한이 많음
- 기존의 응용이나 DB를 코드 변경 없이 그래픽 화면에 이동시키는 기능 부여함
- 기존 Host 응용의 화면 인터페이스만 교체함
- “Screen Scraper”, “Frontware”라 함

<2단계> 그래픽 C/S 응용 작성 단계:

- 기존의 3270/5250 지원 S/W를 사용하지 않고, C/S용 응용을 다시 만듦
- GUI 구현, RDB 사용 지원 기능 부여
- 시중의 대부분의 제품이 여기에 해당함

<3단계> C/S트랜잭션 처리 응용 작성 단계:

- 진정한 의미의 C/S응용 작성이 이루어짐
- 기존 자료를 C/S환경으로 다운로드하여 개방형 DB를 사용함
- 새로운 응용프로그램의 전면적 작성
- 매우 복잡함
- 분산 트랜잭션 지원 기능 부여

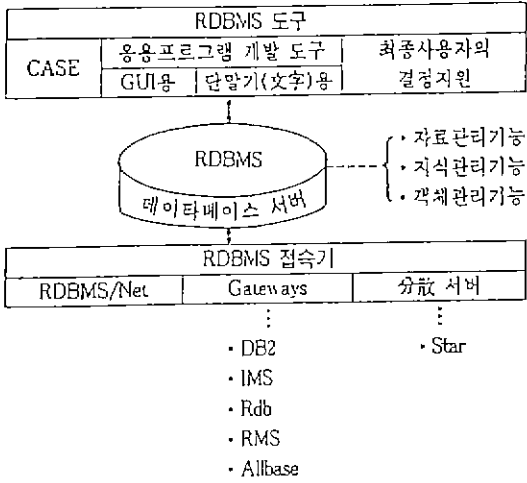


그림 3-6 C/S RDBMS 구조의 3요소[13]

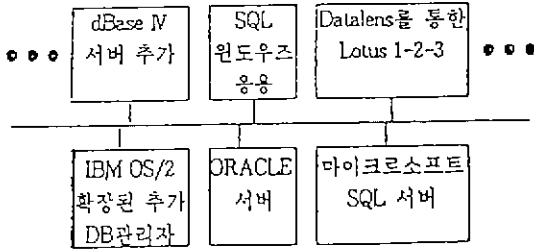


그림 3-7 동일망에서 서로 다른 클라이언트 운용[14]

3.5 데이터베이스 환경

데이터베이스 없이 클라이언트/서버는 존재할 수 없다고 할 만큼 데이터베이스 구축은 필수적이다.

데이터베이스 모형은 일반적으로 망형(Network Model)과 계층형(Hierarchical Model), 관계형(Relational Model) 등으로 나누며 각각 장단점이 있으나 클라이언트/서버 환경에서는 관계형 데이터베이스 관리 시스템(Relational Database Management System : RDBMS)을 구축하는 것이 바람직하다.

RDBMS는 주어진 하나 또는 하나 이상의 표(Table)로부터 고수준의 관계연산자(Relational Operators)들을 이용하여 연산 결과로써 “새로운 관계(New Relation)”를 표로 얻는다.

클라이언트/서버 구조에서 RDBMS를 이용하

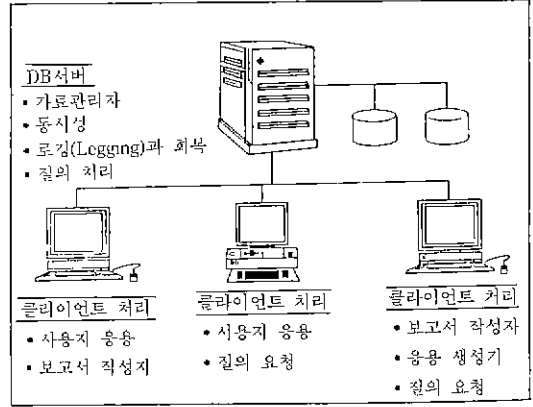


그림 3-8 DB서버를 가진 클라이언트/서버 구조의 예

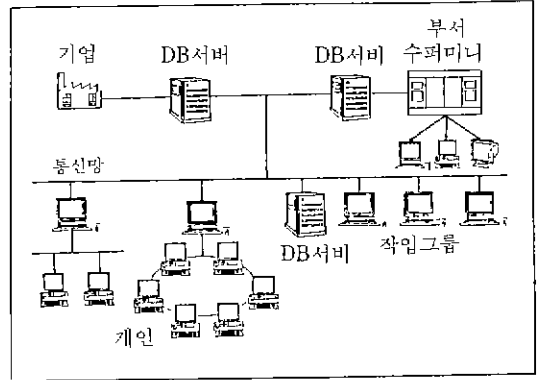


그림 3-9 개방 클라이언트/서버 환경

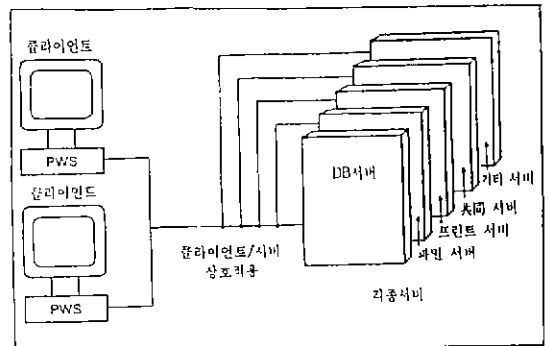


그림 3-10 클라이언트와 서버들의 상호작용[17]

기 위하여는 그림 3-6에서와 같이 최종 사용자가 사용할 RDBMS 도구와 자료, 지식, 객체(대상)들을 관리할 데이터베이스 서버로서 RDBMS,

그리고 RDBMS에 접속(Access)하기 위한 접속기가 반드시 필요하다.

-관계형 데이터베이스 서버의 평가요소

- 1) 다른 클라이언트/서버 구성 요소들과의 호환성
- 2) 성능
- 3) 동시성 제어
- 4) 통신과 접속
- 5) 확장성과 이식성
- 6) 회복과 거래 무결성
- 7) 보안성

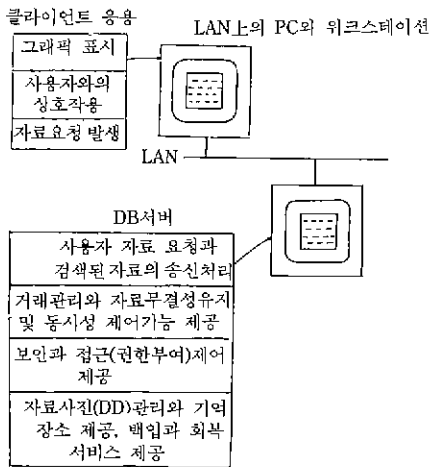


그림 3-11 클라이언트 응용과 DB서버[15]

8) 서버 행정관리

분산 DBMS(Distributed Database Management System : DDBMS)는 일반적으로 세가지 수준(3 Levels)에서, 그리고 자료 독립성(Data Independence)의 관점에서 정의할 수 있다.

이를 좀 더 자세히 보면, 자료가 저장된 보조 기억장치와 물리적수준(Physical Level)에서 볼 때 DDBMS는 다중사용자 환경(Multi-User Environment)과 지역적으로 분산되어 있는 지리적 환경(Geographical Environment)하에서 신뢰성과 지속적인 대량자료이면서 공유된 자료의 관리를 뜻한다.

자료의 배치구역을 나타내는 위치수준(Location Level)에서 보면 분산된 자료의 분산처리로서 자료 저장 장소나 위치에 있어서 자치성(Site Autonomy)을 갖는 지역적으로 독립적인

분산 DB의 장단점

장 점	단 점
<ul style="list-style-type: none"> - 지역적 자치성 - 성능 향상 - 신뢰도와 이용성 향상 - 경제성 - 확장이 용이 - 자료 공유 	<ul style="list-style-type: none"> - 아직 경험이 적다. - 복잡하다 - 비용증가의 위험 관리, 보안이 어렵다. - 바꾸기 힘들다.

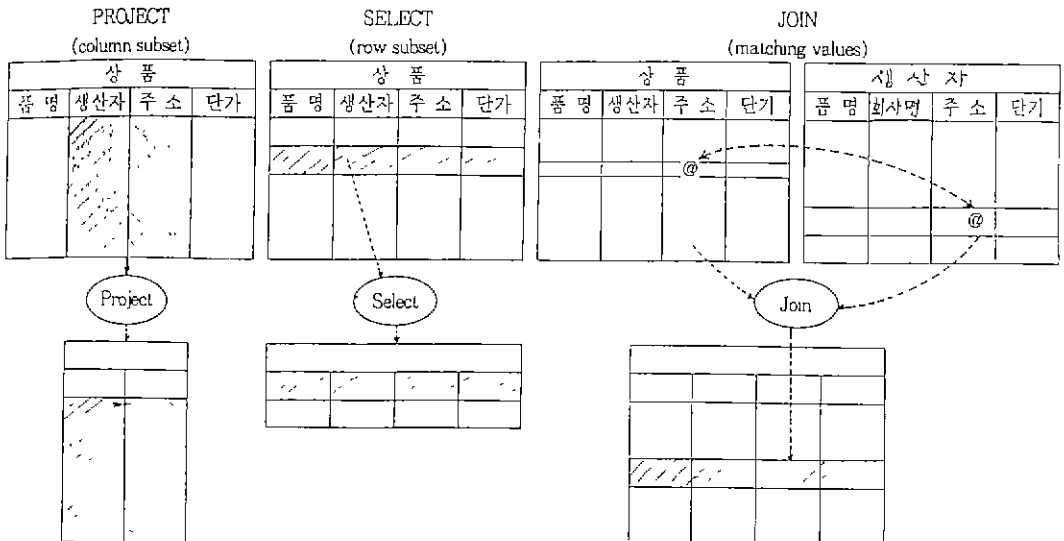


그림 3-12 RDBMS의 세가지 기본 관계연산자

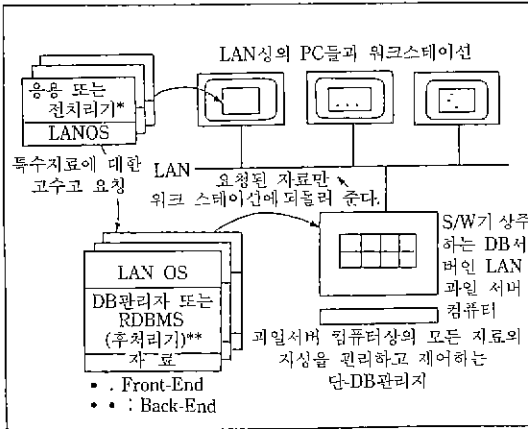


그림 3-13 DB서버로서의 워크스테이션과 PC들의 관계 [16]

시스템이고, 광역적(전역적)으로는 전체적, 종합적인 시스템으로 정의된다.

자료의 독립성은 자료가 저장된 위치(Location)와 수시로 발생하는 트랜잭션(Transaction), 자료의 단편(Fragment), 지역적 또는 국부적 자료들의 중복(Replicate) 등에 투명적임을 의미한다. 나아가서 질의의 최적화(Query Optimization)와 중요한 관련성을 갖는다. 즉, 종래 파일 시스템 처럼 자료의 어느 특정 부분을 검색하기 위하여 파일 전체를 가져오는 것이 아니라 자료 가운데서 꼭 필요한 부분만을 검색함에 있어 자료의 독립성은 매우 중요하다.

분산 데이터베이스에서의 중요 문제점들은 다음과 같은 것들을 들 수 있다.

- 분산 질의 처리
- 분산 거래 처리
- 응답
- 지역 독립성
- 이기종접속의 문제
- 자료흐름 프로토콜
- IPC²⁾/망 프로토콜
- 시스템목록(System Catalogs), SQL구성은 논법
- API(Application Program Interface)문제 또한 분산환경하에서 관계형 데이터베이스(RDBMS)가 가장 잘 유지 되기 위한기술은
- 분해 유연성(Decomposition Flexibility)

- 재구성 능력(Recomposition Power)
 - 전송의 절약(Economy of Transmission)
 - 의도하는 목적과 최적화의 분석가능성(Analyzability of Intent and Optimization)
 - 분산 독립(Distribution Independence)
- 등이다.

참고적으로 C. J. Date가 주장하는 분산 데이터베이스 관리 시스템(DDBMS)에 대한 12가지 규범을 열거하면 다음과 같다[23].

- 지역 자치성(Local Autonomy)
- 중앙부(Central Site)에 의지하지 않고도 사용
- 지속적인 조작
- 지역 독립
- 단편 독립(Fragment Independence)
- 중복 독립(Replicate Independence)
- 분산 질의 처리
- 분산 거래 관리
- 하드웨어 독립
- 운영체제(OS) 독립
- 망 독립(Network Independence)
- DBMS 독립

참고문헌

- [1] 필자 의역
- [2] 노중호, 다운사이즈화 정보시스템 설계방법, 시에치노시스템컨설팅, P. 10, 1992.
- [3] 4. 6., Paul E. Renaud, Introduction to Client/Server Systems, John Wiley & Sons, Inc. PP. 6~7, 1993.
- [5] Patric Smith. Client/Server Computing, Sams Publishing, P. 16, 1992.
- [7] 강인두, ITU '93 Vol. 1, SDS P. 380, 1993.
- [8] C. Parsons, ITB '93. Vol. 2, P. 723, 1993.
- [9] Ian Palmer, "Tools and Methodology", ITB '93, Vol. 3, PP. 9~12, 1993.
- [10] Dawna Travis Dewire, Client/Server Computing, McGraw-Will, Inc. P. 125, 1993.
- [11] 오혜영, "기업의 GUI 전략".ITB '93,PP. 553~587, 1993.
- [12] James J. Mc Carthy, "Migration to the Client/server Model", Proceedings of Client/Server Conference '93 at san Jose.

[13] 임만직, "신경망체제 도입을 위한 Client/Server의 전략적 활용방안", Proceedings of ITB '93 Conference, SDS, P. 86, 1993.

[14] 15. 16., S.Khoshafian and *et. al.*, A guide to Developing Client/Server SQL Applications, Morgan Kaufman Publishers, 1992.

[17] ~[22] Alex Berson, Client/Server Architecture, McGraw-Hill, PP. 36~383, 1992.

[23] C. J. Date, An Introduction to Database System, Vol. 1, 5th Ed., Addison Wesley, Inc., PP. 620~627, 1980.

[24] Andrew S. Tanenbaum, Modern Operating System, Prentice-Hill, Inc, PP. 362~393, 1992.

[25] 小野欽司外 6人, OSI プロトコル槍とき讀本, オム社, P. 34, 1990.

[26] 조국현외, 컴퓨터 네트워크[OSI], 정익사, P. 36, 1989.

[27] Microsoft社 "Designing Client-Server Application for Enterprise Database Connectivity", Technical note, Microsoft Co., Vol. 3, No. 14, PP 3~5, Jun. 1992.

[28] [29], 문송천, "Client/Server 컴퓨팅 환경의 국내 구축 방법", ITB '93, SDS, PP. 10~14, 1993.

[30] John Shirley, Guide to Writing DCE Applications, O'Reiley & Associates, PP. 1~20, 1992.

[31] John Bloomer, Power Programming with RPC, O'Reiley & Associates, PP. 1~5, 1991.

[32] [33], 서울대학교 하계강좌 자료, Client/Server 구축 방법론(TCP/IP), 자연과학종합연구소, 1993.

이 기 현



1965 성균관대학교 경상대학
경제학과 졸업
1972 성균관대학교 경영대학
원 정보처리전공(경영학
석사)
1976 총무처 행정전산계획실
전산처리관
1981 대한손해보험협회 전산
실장
1986 광운대학교 대학원 전산
학과(이학석사)

1993 광운대학교 대학원 전산학과(이학박사)
1993 ~ 현재 명지대학교 공과대학 전자계산학과 교수
연구 분야 : OS, 컴퓨터네트워크, 소프트웨어 공학
