

《主 題》

통신 프로토콜 소프트웨어의 객체지향 모형

최 병 규 · 송 주 석

(연세대학교 전산학과)

■ 차 례 ■

I. 서론	V. 통신망의 객체지향 모형
II. 통신 프로토콜 소프트웨어의 객체지향 모형	VI. 통신 서비스를 위한 객체지향 프로그래밍 언어
III. 미래의 통신 서비스에 필요한 신호기능	VII. 결론
IV. 멀티미디어 통신 프로토콜 소프트웨어의 객체지향 모형	

I. 서 론

통신망은 전달 매체 기술의 발전으로 점차 고속화 되어 여러 가지 서비스를 종합한 통합망으로 진화되고 있다. 이에 따라 현재의 PSTN에서 제공되는 호의 개념은 통합된 통신망에서 제공될 수 있는 다양한 응용 서비스 중의 한 형태로 인식되어야 한다. 이러한 통신 환경의 발전 추세에 따라 새로운 통신 서비스의 추가와 기존 서비스의 변경 등이 끊임없이 요구되고 있다. 그러나 새로운 통신 서비스의 추가시 기존 통신 프로토콜 소프트웨어의 구조를 바꾸는 것은 많은 개발 비용과 시간이 소요될 것이다. 그러므로 통신 프로토콜 소프트웨어는 미래의 통신망 환경을 고려하여 통신 응용 서비스와 전달 매체에 독립적인 구조를 가져야 할 것이다. 본 고에서는 그 방법론으로 최근 많은 연구가 진행되고 있는 객체지향 패러다임을 이용하여 통신 서비스(telecommunication service)와 통신 프로토콜 소프트웨어, 미래의 통신망인 B-ISDN에서의 멀티미디어 통신 서비스 등에 대한 모형을 연구하고, 미래의 통신망에 요구되는 신호개념 모형, 통신망의 객체 모형을 간략하게 언급한다. 마지막으로 ITU-TS에서 권고된 객체지향 언어인 Object CHILL의 특성을 간단히 살펴본다.

II. 통신 프로토콜 소프트웨어의 객체지향 모형

1. 통신 서비스의 객체 모형화 필요성

1)배 경

‘통신 서비스는 망이 망 이외의 다른 역할자에게 제공하는 망 능력의 집합’이며 다른 역할자는 최종 사용자, 망 제공자, 서비스 제공자를 의미한다[1]. 서비스란의 간섭이란 하나의 서비스 엔터티가 다른 서비스 엔터티로부터 영향(변경, 삭제 등)을 받는 것을 말한다. 간섭의 원인은 망 자원의 유한성에 의한 것과 서비스 구조(하나의 서비스가 다른 서비스를 이용하여 구현된 경우)에 의한 것으로 대별될 수 있다. 이러한 간섭이 설계자의 의도에 의한 것이라면 간섭 자체가 또하나의 서비스가 될 수 있으나 그렇지 않은 경우의 간섭은 반드시 배제되어야 한다. 통신 서비스를 개발 단계별로 생각해보면 간섭은 서비스 생성 환경이나 서비스 동작 환경에서 검출될 수 있는데 서비스 생성 환경에서는 텍스트상이나 시뮬레이션을 통하여 간섭을 검출하며 이경우 재설계에 의해서 간섭을 배제시킬 수 있다. 동작 환경에서는 간섭을 검출하기가 생성 환경에서보다 더 어려우며 때로는 불가능할 수도 있다.

2) 객체 모형 예

서비스간의 간섭을 배제시키고 새로운 서비스의 수용에 유연하게 대처하기 위한 방법론으로 객체지향 모형이 최근 많이 연구되고 있다. 아직까지 통신 서비스의 객체 모형에 표준화 된것은 없지만 최상위 개념으로 통신 서비스를 '서비스와 서비스를 구현하기 위한 통신 시스템'으로 나누면 서비스 설계자는 통신 시스템의 구현과 독립적으로 설계를 진행할 수 있는 장점이 있다. <그림 1>에는 ROSA의 객체 모형 예제가 그려져 있다. a)에서는 객체 a가 객체 b에게 서비스 인터페이스를 제공하고 있다. 객체 b는 a의 인터페이스를 알고 있으므로 언제나 서비스 A를 부를 수 있다. b)에서는 a가 b에게 인터페이스 B를 통하여 Oper1을 호출하면서 자신의 인터페이스 A를 알려준다. b는 a로부터 Oper1을 호출받고 응답으로 ResOp1을 a에 보내면서 동시에 인터페이스 A를 통하여 a에게 Oper2를 호출한다. c)에서는 a가 자신의 인스턴스 b를 생성하고 b는 자신의 인터페이스 B를 a에게 알려준다. <그림 1>의 객체 모형은 여러 객체로부터 통신

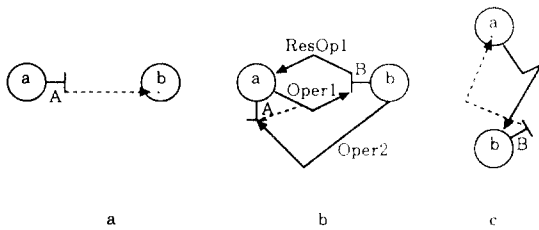


그림 1. ROSA 객체모형

서비스의 동적인 구성을 위하여 동적인 객체 생성, 동적인 인터페이스 생성, 인터페이스의 매개변수 전달 (parameter passing) 등을 채택했다. 이러한 객체 모형의 개념을 적용하여 통신 서비스를 재정의하면 '객체가 수행할 수 있는 동작'이 된다.

3) 객체 모형 적용 예

객체지향 설계는 통신 서비스 환경의 완벽한 묘사와 통신 서비스 구조 설계를 통한 복잡도의 감소로 설계 단계부터 간접작용을 배제할 수 있다. 객체 모형을 통신 서비스 설계에 적용한 예가 <그림 2>에 그려져 있다. 이와 같은 기본적인 서비스 환경을 하나의 Trader와 몇명의 User, 그에 해당하는 User Agent와 User Profile, 그리고 호 객체로 이루어진다. 여기서 Trader는 많이 제공할 수 있는 기초 서비스를 가지고 있다. User는 어느 순간의 통신 서비스 인스턴스의 최종 사용자이다. User Agent는 통신 서비스 인스턴스와 사용자간의 교량역을 담당한다. 이것은 특히 UPT의 경우에 중요하다. User Agent는 동적으로 생성되는 서비스를 관리하면서 부개의 인터페이스를 갖는다. 하나는 서비스에 대한 인터페이스로 명의 요청사항을 수행하고 다른 하나는 사용자에 대한 인터페이스로 사용자의 요구 사항에 응답한다. User Profile에는 사용자 종속적인 서비스가 등록되어 있다. 다시 말하면 '언제 어느 조건에서 사용자의 어떤 서비스가 기능되는가?'하는 사항이 등록되어 있다. 그림상에 적용하는 잠깐의 사용자간의 관계이다. User Agent는 호를 직접 관리(설정, 해제)한다.

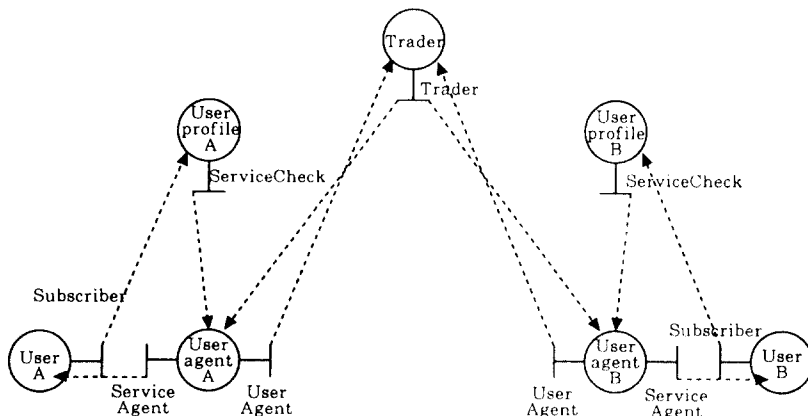


그림 2. 통신 서비스의 객체모형 예

2. 통신 프로토콜 소프트웨어의 객체지향 설계 방법

1) 객체지향 소프트웨어 개발 단계

<그림 3>에 일반적으로 생각해 볼 수 있는 통신(교환) 프로토콜 소프트웨어의 객체지향 개발 5단계가 그려져 있다[2]. 먼저 분석단계에서는 시스템 요구 사항과 필요 기능을 명세하고 시스템을 몇개의 부시스템으로 분할하여 부시스템간의 인터페이스를 정의한다. 2번째 단계인 설계 단계에서는 우선 시스템 기능 구현에 필요한 객체를 정의하는 것이 필요하다. 필요한 객체가 모두 정의되면 다음은 각 객체에 메소드(methods)와 어트리뷰트(attributes)를 할당한다. 메소드와 어트리뷰트는 각각 기존 환경의 기능 모듈(function)과 데이터에 해당한다. 다음으로 객체간의 인터페이스를 명세하고 객체간의 종속 관계를 도출한다. 객체간의 종속 관계가 정의되면 종속 관계에 따라 객체를 분류하여 클래스를 만든다. 클래스를 중심으로 객체간의 종속관계(hierarchy)를 그리면 객체간의 제어 로

직과 데이터의 할당이 상세히 이루어진다. 여기서 객체간의 종속(inheritance)성질을 정의할 수 있다. 3번째 단계는 설계 분석서를 그대로 구현하는 단계이다. 4번째 단계인 통합 단계에서는 개별자별 소프트웨어 결과(component)물을 통합하여 개인 결과물간의 인터페이스 시험을 거친다. 마지막 단계인 시스템 시험 단계에서는 개발에 참여하지 않은 시험 전담 요원이 시스템 단위의 기능 시험을 수행한다.

2) 객체지향 설계시 고려할 점

통신 프로토콜 소프트웨어의 객체지향 설계시 고려할 점을 간단히 살펴 보면 다음과 같다.

2.1) 설계조정

객체지향 설계의 가장 큰 장점중의 하나는 소프트웨어의 재사용성이다. 이 장점을 살리기 위해서는 각 설계자는 자신이 설계한 객체가 다른 소프트웨어에서도 필요시 사용될 수 있도록 해야 한다. 효율적인 설계를 위한 방법으로 설계 조정자 또는 그룹을 운용할 수 있다. 설계 조정자는 정의되는 모든 객체의 기능 중복성이나 인터페이스 문제를 사전에 검토한다.

2.2) 클래스 정의

객체 정의 단계 후에는 객체간의 인터페이스를 명세하면서 객체간의 종속관계를 그릴 수 있다. 이 시점에서 객체마다 수시로 적용되는 데이터 구조나 메소드는 일반적인(generic) 클래스로 도출할 수 있다. 결국 공통적인 데이터 구조나 기능들을 별도의 클래스로 정의할 수 있는 것이다. 이와는 달리 응용기능에 관련된 객체의 종속관계에서 정의된 클래스는 응용기능에 종속적인 클래스가 된다. 즉, 클래스의 정의를 2종류로 구분하여 진행하면 보다 더 효율적인 객체 정의를 할 수 있다.

2.3) 계층 모형

객체간의 인터페이스는 메소드만을 통하여 이루어지므로 구체적인 객체를 나열하여 응용기능의 동작을 묘사하는 것은 효율적이지 못하다. 다시 말하면 하나의 응용 기능(예 :호)의 동작을 구현 단위인 객체간의 메시지 교환으로 기술하는 것은 상세설계 단계에서 개발자에게 필요한 것이고 상위설계 단계에서는 클래스간의 동작관계를 계층화하여 도식화 하는 일이 필요하다. 즉, 클래스간의 동작이 계층으로 모형화 되면 외부나 내부의 사건(예 : 제어신호 메시지 수

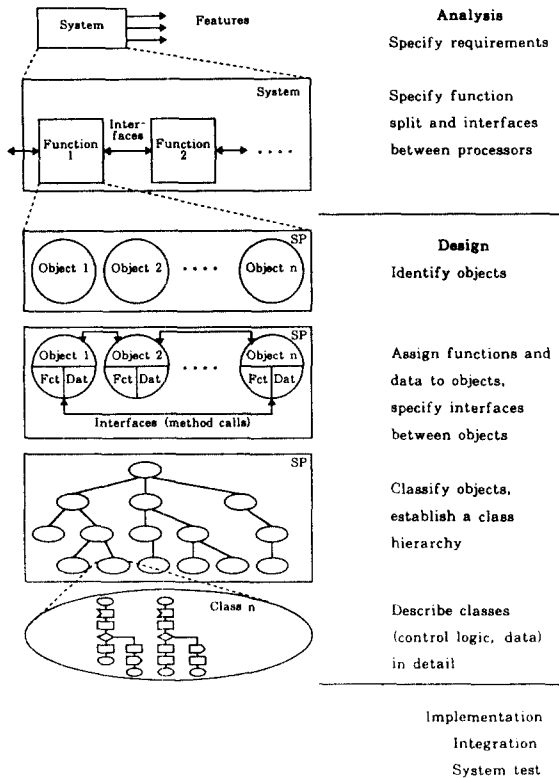


그림 3. 객체지향 소프트웨어 개발단계

신)에 대응하는 동작은 클래스간의 동작으로 분명하게 설명된다.

2.4) 결과 평가

“객체지향 설계로 구현한 통신 프로토콜 소프트웨어가 기존의 기능 중심(functional)언어로 구현된것보다 얼마나 더 우수한가?”를 측정하려면 이론적으로 똑같은 시스템 기능을 두가지 방식으로 구현하고 평가하여야 하지만 일반적인 측면을 생각해보면 생산성과 성능 그리고 유지보수를 들 수 있다. 생산성 측면에서는 객체지향 방법으로 설계된 소프트웨어에서 실제로 프로그램 행수의 감소가 예측되고 유지보수 측면에서는 당연히 객체지향 설계가 우수하다고 하겠다. 다만 객체간의 메소드 호출로 인한 잦은 메모리 액세스로 수행 속도가 기존의 소프트웨어보다 다소 느려질 것으로 예측된다. 그러나 완전한 평가는 객체지향 설계로 개발된 통신 프로토콜 소프트웨어의 생명주기(life cycle)를 평가함으로써 이루어진다고 하겠다.

3. 통신 프로토콜 소프트웨어의 객체지향 모형

사례

1) GPT사의 객체지향 모형

GPT사는 통신 서비스를 제공하는 통신 프로토콜 소프트웨어가 크게 두가지의 환경변화, 즉 통신기술의 변화와 시장의 요구 변화에 늘 대처해야 함을 밝히고 이 두가지의 변화에 대해 유연하게 반응하기 위해서 객체지향 모형을 제시하였다[3]. 먼저 여기서는 통신 서비스를 다음(그림 4)와 같이 인식한다. (그림 4)에는 통신 서비스가 응용기능층, 호제어층, 물리자원 제어층의 3 계층으로 나뉘어져 있다. 여기서 중간층인 호 제어층은 응용기능에 독립적으로 사용될 수

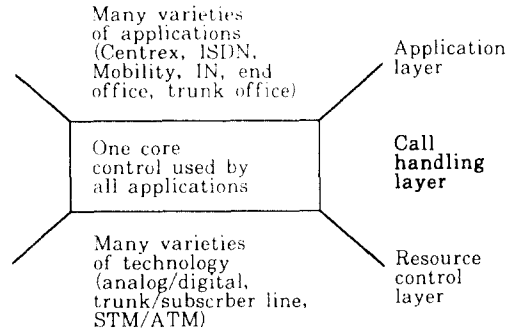


그림 4. 통신 서비스의 계층개념

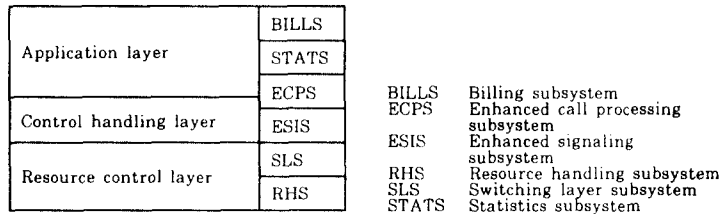
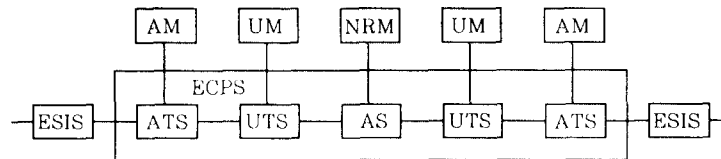


그림 5. 통신 프로토콜 소프트웨어 계층개념



AM Access manager
 AS Associator segment
 ATS Access transaction segment
 ECPS Enhanced Call processing subsystem
 ESIS Enhanced signaling subsystem
 NRM Network routing manager
 UM User manager
 UTS user transaction segment

그림 6. 호 연결모형

있도록 개념이 정리되었다. 이 모형을 계층개념으로 발전시킨 그림이 <그림 5>이다. <그림 5>는 기본 호 개념의 호 제어 서브시스템에 해당한다. 호 제어층은 다시 ECPS와 ESIS로 나뉜다. 호는 분산 객체로 정의되며 ECPS 내부에서 몇 개의 세그먼트로 구성된다. 하나의 스위치 시스템을 통한 두 사용자의 연결 모형을 세그먼트를 사용하여 <그림 6>에 제시 하였다. 각 세그먼트에는 그에 해당하는 관리자가 있다. GPT사는 이 모형을 기본으로 POTS(Plain Old Telephone Service)를 모형화하고 나아가 다중연결호, 지능망을 이용한 호, B-ISDN 호 등을 모형화하고 있다.

2) Siemens사의 객체지향 모형

Siemens사는 ISDN 교환 시스템에 필요한 호 처리 소프트웨어를 객체지향 모형을 통하여 설계, 구현하고 결과 평가를 제시하고 있다[2]. ISDN 교환 시스템은 가입자측 신호방식으로 DSS1이 사용되고 망측 신호방식으로 SS No.7이 사용된다. 교환 시스템 모형이 <그림 7>에 제시되어 있다. 교환 시스템은 크게 교환 프로세서와 운용 프로세서로 나뉘고 교환 프로세서는 DSS1, No.7 프로토콜로 외부와 인터페이스한다. 시스템 모형으로부터 호 처리 소프트웨어를 분리시켜 클래스로 부합시킨 호 처리 소프트웨어 모형이

<그림 8>에 나타나 있다. 그림에서 Line Access Control은 가입자 신호 처리를 담당하고 Call/service control은 가입자의 개인별 데이터와 과금 데이터를 관리하며 호의 설정과 해제를 담당한다. 즉, 가입자와 망을 중계하는 역할을 수행한다. Bearer connection control은 ISUP 프로토콜에 따라 link-by-link 방식으로 통신 경로 연결 절차를 담당한다. Connectivity control은 교환 시스템 내부의 스위칭 네트워크의 연결과 해제를 담당한다. 하드웨어 자원은 Connection control 클래스에 숨겨져(ecapsulation) 있다. Siemens사는 이 모형을

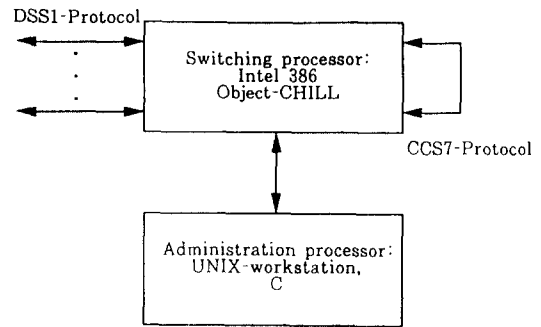


그림 7. ISDN 교환 시스템 모형

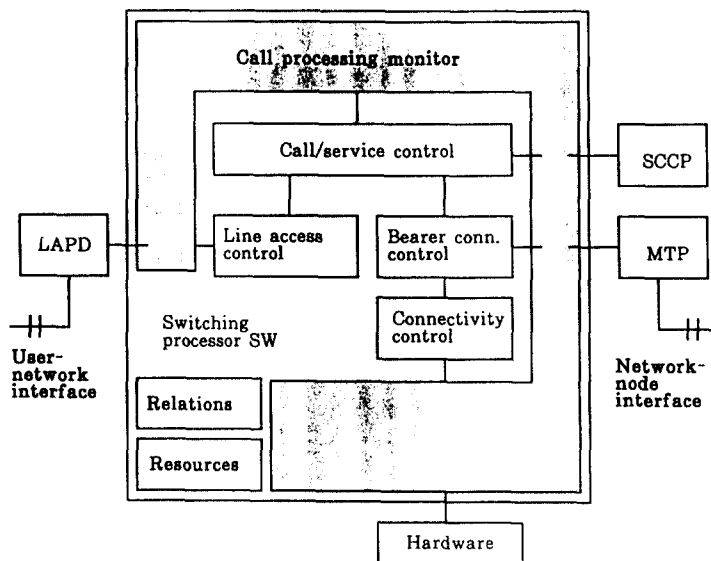


그림 8. ISDN 호 처리 소프트웨어 모형

기본으로 ISDN 호 처리 소프트웨어를 개발하고 기존의 절차 중심으로 개발된 호 처리 소프트웨어와 비교하여 프로그램의 행수와 개발기간이 감소한다는 사실을 발견하고 객체모형 방식을 통신 프로토콜 소프트웨어의 구현에 적용하면 좋은 결과를 얻을 수 있다고 지적하고 있다.

3) ETRI의 객체지향 모형

ETRI는 HAN/B-ISDN ATM 교환 시스템에 필요한 교환 소프트웨어를 성능보다는 유연성과 유지보수성에 초점을 맞추어 객체지향 모형을 제시한다[4]. 교환 시스템의 논리모형과 프로그램 모형을 일치시킨 교환 시스템 모형이 <그림 9>에 그려져 있다. 여기서의 교환 시스템은 가입자망, 지능망, N-ISDN 망, PSDN 망과 연동할 수 있는 모형으로 정의되었으며 그러기 위해서는 메타신호, Q.93B, B-ISUP 등으로 외부의 사건에 응답해야 한다. 또 MMI(Man-Machine Interface)를 통하여 시스템 운용자와 정합하고, 망 관리를 위해서는 F 및 Q 인터페이스를 통하여 TMN과도 연동해야 한다. 다시 말하면 기존의 모듈망과 연동할 수 있는 기능을 모형화 한셈이다. 그림에서 소프트웨어 플랫폼은 시스템 내부 환경을 제공하는 응용 소프트웨어 개발 환경이다. <그림 10>에는 교환 소프트웨어의 객체 모형이 제시되어 있다. 교환 시스템과 같이 실시간 처리가 요구되는 시스템에서는 객체 설정이 중요한 문제로 대두되며 제시된 모형에서는 소프트웨어를 서비스 층, 프로토콜 층, 물리자원 층의 3계층으로 나눈다. 여기서 프로토콜 층은 하부 계층과 상위 계층에 독립적으로 구성되어 기존 서비스의 변경이나 새로운 서비스의 수용 문제를 손쉽게 해결할 수

있으며, 물리적인 구성도 하부 계층 객체에 숨겨져 (information hiding) 있어 프로토콜 층은 하드웨어 장치 변화에 영향을 받지 않는다. 이와 같은 물리계층과 프로토콜 계층의 분리 및 하위 계층과 서비스 계층의 분리를 통하여 virtual hardware를 구축함으로써 시스템의 형상이 바뀌어도 소프트웨어가 영향을 받지 않도록 투명성을 보장한다. ETRI는 이 모형을 기본으로 구체적인 설계 단계에서 시스템 특성에 맞는 복합 객체, 능동객체, 병렬객체, 수동객체 등을 정의하고 있다.

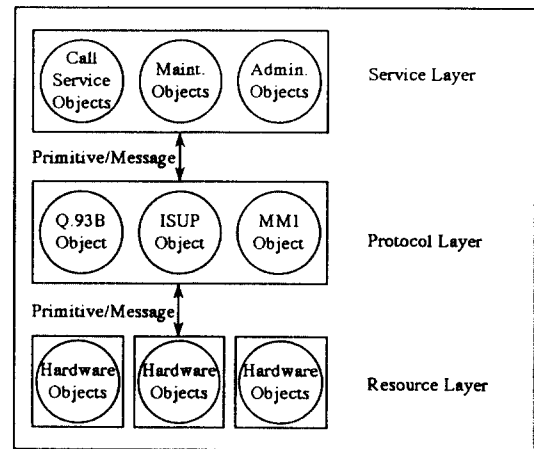


그림 10. 교환 소프트웨어 객체 모형

III. 미래의 통신 서비스에 필요한 신호기능

1. 배경

신호기능은 “전기통신망(telecommunication network)에서 연결 설정과 제어, 망 관리 등에 관한 정보의 교환”이라고 말할 수 있다[5]. 신호기능은 <그림 11>에 그려진 바와 같이 신호정보 교환의 주체에 따라서 사용자-망간 신호, 노드간 신호, 사용자-사용자간 신호로 분류할 수 있다. 서비스 사용자의 입장에서 느낄 수 있는 사용자-망간 신호기능을 바라보면 이것은 새로운 통신 서비스의 대두와 교환 시스템의 기술발전 에 따라 자식식, 공진식, 다이얼 펄스식, DTMF(Dual Tone Multi-Frequency) 방식 등으로 계속 발전하여 왔는데 이들은 모두 ‘stimulus signaling’이라고 볼 수 있으며 N-ISDN에 이르러 최초로 가입자 터미널이 지능을 갖추게 되고 ‘functional signaling’을 수행하게

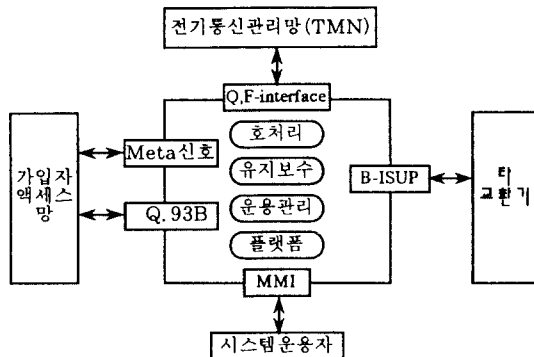


그림 9. 교환시스템 모형

된다. 그러나 통신망이 N-ISDN을 거쳐 B-ISDN으로 진화하게 되면서 <표 1>과 같은 새로운 통신 서비스가 예측되고 이들을 지원할 수 있는 새로운 신호기능에 대한 연구가 이루어지기 시작했다. B-ISDN에서 제공될 수 있는 호의 형태는 N-ISDN에서 정의되는 호의 점대점 형태를 뛰어 넘어 점대 다중점(multicast), 방송형(broadcasting), 그리고 궁극적으로 제공될 수

있는 다중점 대 다중점 멀티미디어 호가 된다. 단말의 형태도 기존의 전화기, 데이터 터미널, 그리고 ATM 호환 단말기와 멀티미디어 등 모든 단말과 함께 기존의 모든 망과도 연동이 가능해야 한다. 본 고에서는 B-ISDN 신호의 복잡한 메시지에 대하여는 언급하지 않으며, B-ISDN에 요구되는 신호기능을 추상화하여 표현하고자 한다.

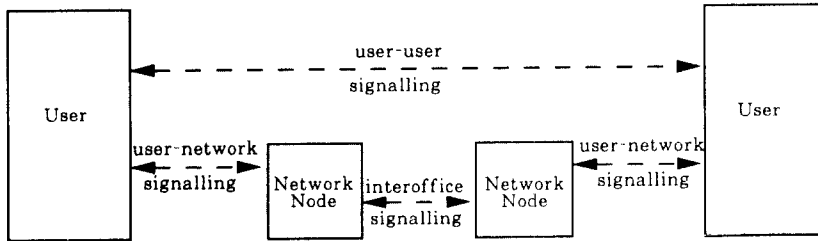


그림 11. 신호기능의 직용분류

표 1. B-ISDN 통신서비스

Possible Broadband Services	
LAN Interconnection	connecting local area networks at different locations
Remote disk server	access to a remotely located shared disk server
Desktop teleconferencing	voice/video/data teleconferencing from one's desk
Distribution video	transport for programming services such as cable television
Video-on-demand	single user controlled access to a remotely located video source
Videotelephone	voice/video telephone calls
Still picture services	access to shop-at home and other information services

2. 통신 서비스에 독립적인 신호 프로토콜 모형

신호 프로토콜을 모형화 하는 목적은 특정한 통신 서비스가 새롭게 요구될 때마다 그 응용 서비스에 종속적인(service specific) 사용자-망간 신호 프로토콜을 개발하기 위한 기본틀을 제공하기 위한 것이다. 따라서 여기서는 사용자 인터페이스나 응용기능 인터페이스는 고려하지 않는다. 우선 두 사용자의 연결을 신호 프로토콜의 관점에서 그래프로 모형화하면 <그림 12>와 같다. 여기서 Edge는 연결의 논리적 부분(segment)이다. Access edge와 Transport edge는 각각 사용자의 액세스 능력과 망내 전달능력을 의미한다. Vertex는 논리적 엔티티로서 상위층의 정보처리 기능을 담당한다. 연결에 참가한 사용자는 User Vertex로 대표된다. 각 User vertex는 결합된 Network edge와 쌍을

이루며 Access edge로 연결된다. 그림에서 그래프는 논리적인 연결성을 의미하고 물리적인 자원의 연결을 의미하지는 않는다. 망은 논리적인 연결에 필요한 자원을 할당하게 된다. 다시 말하면 그래프는 서비스의 관점에서 연결의 논리적 구성을 한눈에 조망할 수 있게 한다. 연결을 규정하는 데이터는 망의 어느 한곳에 집중될 필요가 없으며 각 노드는 자신의 입장에서 보이는 연결에 대한 데이터만 가지면 된다. 즉, 사용자나 망은 분산된 호 처리 환경에서 자신의 입장에서 바라본 데이터를 관리하면 된다. 이 모형은 N명의 참가자로 이루어진 연결로 확장될 수 있다. <그림 13>에는 4명의 참가자로 이루어진 연결형태의 두가지 예가 제시되어 있다. a)는 사용자 A가 사용자 B를 호출하고 사용자 B는 다시 사용자 C를 호출하며 C는 다시

D를 호출한 연결 모형이다. 이 경우 사용자 A는 사용자 C, D를 볼 수 없으며 가령 B와 C사이의 연결이 해제되어도 A에게는 아무런 변화도 일어나지 않게 된다. b)는 사용자 A 혼자서 B, C, D를 차례대로 호출한 경우의 연결 모형이다. 이 경우 사용자 B, C, D는 서로 알 수 없으며 가령 A와 B의 연결이 해제되어도 C와 D는 아무런 영향을 받지 않는다. <그림 13>은 연결이 하나인 경우의 모형이지만 그래프를 이용하여 다수의 연결로 이루어진 호도 모형화가 가능하고 멀티미디어인 경우 사용자는 호에 구성된 모든 연결에 다 참여할 필요는 없다. 이 그래프 모형 방법을 이용하면 B-ISDN이 제공할 수 있는 모든 형태의 호에 대처할 수 있는 신호 프로토콜을 모형화할 수 있다. 즉, 새로운 서비스의 등장이나 교환기술의 발전에 유연하게 신호 프로토콜을 만들 수 있다.

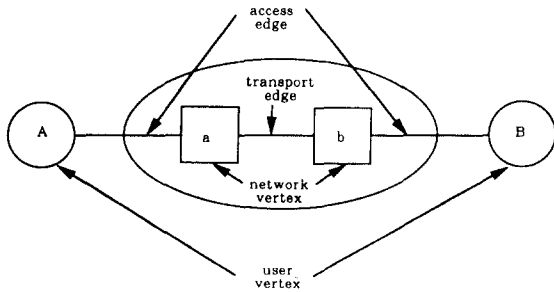


그림 12. 두 사용자의 연결 모형 그래프

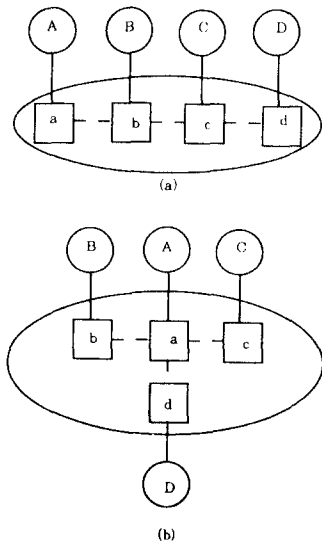


그림 13. 4명의 참가자로 이루어진 연결모형 그래프

IV. 멀티미디어 통신 프로토콜 소프트웨어의 객체지향 모형

1. 배경

현재의 통신망은 특정 서비스에 목적을 두고 구축되었으나(예: PSTN, PSDN) B-ISDN은 현재의 모든 통신 서비스를 수용하고 미래의 서비스도 별도의 망 구축없이 B-ISDN에서 제공 가능하도록 연구되고 있다[6]. 단말의 형태로 기존의 전화기, 데이터 터미널, 비디오 터미널등의 기능을 통합한 멀티미디어가 미래의 궁극적인 단말로 예견되고 있다. 멀티미디어는 사람에게 친숙한 사용자 정합기능을 제공할 뿐만 아니라 B-ISDN의 단말로써 사용자에게 통신서비스를 제공하게 된다. 현재 시점에서 기존의 워크스테이션 외형을 가진 멀티미디어를 쉽게 떠올릴 수 있겠으나 미래에는 무선망과 이동통신망을 이용하는 휴대용 소형 멀티미디어 출현이 가능한 것이다[7]. 이와 같이 미래에는 복합단말 형태인 멀티미디어가 사용자에게 가장 복잡한 통신 서비스를 제공해주는 궁극적인 대체모써 작용하게 되므로 멀티미디어를 이용한 통신 서비스 연구는 세계 각국에서 자연스럽게 추세로 활발히 진행되고 있다.

2. 멀티미디어 호 모형

멀티미디어 통신 서비스 설계의 주안점은 특정한 서비스에 종속적인 멀티미디어의 응용층과 하드웨어 장치에 종속되어 망과의 인터페이스를 담당하는 물리층과 연결 관리를 담당하는 통신 프로토콜 소프트웨어를 독립적으로 구성하는 것이다. 이 목표는 앞에서 언급되었던 통신 서비스 계층 개념(<그림 4>, <그림 10>)과 유사하다. 먼저 사용자가 멀티미디어 호를 기개(멀티미디어, 망)가 제공하는 하나의 서비스로 바라본 경우의 사용자, 멀티미디어, 망의 관계에 대한 객체모형이 <그림 14>에 그려져 있다[8]. 여기서 연결 관리 기능은 사용자 접속 기능과 데이터 전달 접속기능, 그리고 다른 많은 응용기능들로부터 호와 연결에 관련된 요구를 받아들이고 적절한 기능을 수행한다. 응용기능과 물리기능에 독립적인 연결관리를 위한 멀티미디어 호의 개념이 <그림 15>에 그려져 있다. 그림에서 하나의 멀티미디어 호는 N개의 연결과 M명의 사용자로 이루어져 있다. N과 M은 같을 필요가 없으며 연결을 나타내는 수직선과 사용자를 나타내는 수평선의 교차점은 연결과 사용자와의 관계를 의미한다. 즉, 그림에서 사용자 1은 연결 1을 통하여

상대방과 데이터를 송수신하는 반면, 사용자 2는 연결 1과 아무런 관계를 맺지 않고 있다. 사용자 M은 연결 N을 통하여 송신만을 할 수 있다. 이 모형은 N

과 M, 그리고 사용자와 연결의 관계를 주요 객체로 설정하면 어떤 연결 형태도 기술이 가능하다.

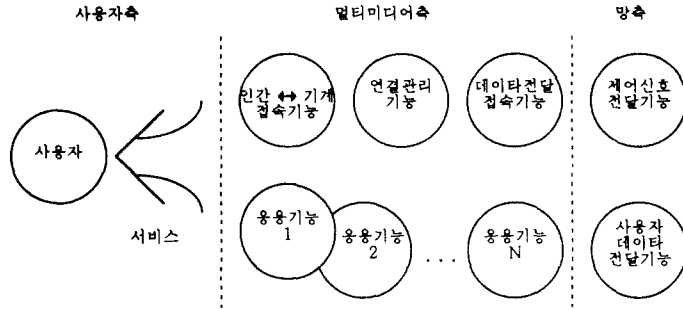


그림 14. 멀티미디어 서비스 객체 모형

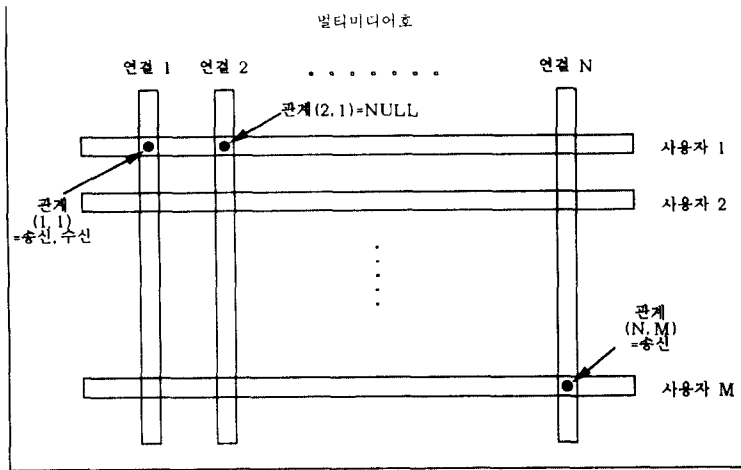


그림 15. 멀티미디어호 개념도

3. 멀티미디어 통신 서비스의 객체설정 예

1) 예 1

분산된 멀티미디어 응용기능을 위한 통신 프로토콜 소프트웨어의 객체 모형 예가 <그림 16>에 그려져 있다[9]. 그림에서 통신 프로토콜 소프트웨어 모형은 결합 객체, 동기화 객체, 연결 객체의 세가지 주요 객체로 이루어지며 이것들은 응용기능에 곧바로 액세스 될 수 있고, 망 경로 객체는 하드웨어 종속적인 객

체로 정의된다. 먼저 결합객체는 통신 엔터티의 입장에서 통신의 전체 성격을 규정한다. 이것의 어트리뷰트는 응용 엔터티의 주소, 응용 문맥(context), 연결관리 테이블, 동기화 관리 테이블 등이다. 이에 해당하는 주요 메소드는 결합의 설정과 해제, 엔터티의 추가와 삭제, 데이터 송신 등이 있다. 결합 객체의 동작은 연결 객체와 동기화 객체에 한 동작으로(atomicly) 적용된다. 응용 엔터티는 여러 개의 결합에 참가할 수

있으므로 여러 개의 결합 객체를 가질 수 있다. 다음으로 연결 객체는 특정 하드웨어 매체를 이용한 데이터의 송수신에 관련된 성격을 규정한다. 이것의 예로 리프트는 연결 확인자, 응용 엔터티의 주소, 연결 종류, 송수신 모드, 송신 품질, 송신 단위, 오류 복원 모드 등이다. 이에 해당하는 주요 예로는 연결의 설정과 해제, 연결 상대방의 추가와 삭제, 데이터 송신 등이 있다. 또 연결 객체는 제어용 데이터 송수신을 위한 제어 연결 객체와 사용자 데이터를 위한 데이터 연결 객체로 나뉜다. 그림에서 나타난 바와 같이 하나의 결합 객체는 여러 개의 연결 객체를 가지고 특정 종류의 데이터 송수신을 할 수 있다. 마지막으로 동기화 객체는 하나의 결합 객체에 속한 데이터 연결간의 동기화 요소를 규정한다. 동기화 요소로는 동기화에 관련된 데이터 연결중에서 기준이 되는 연결과 그에 대한 동기화 요구 수치, 그리고 동기화 단위 등이 있다.

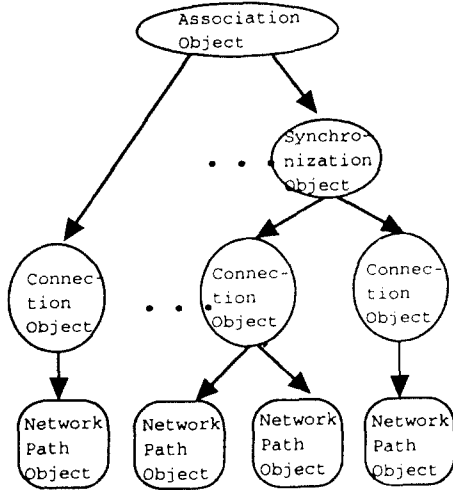


그림 16. 멀티미디어 통신 프로토콜 소프트웨어 객체 모형 예

2) 예 2

원도우 기능을 갖는 사용자간의 비디오 통신을 이용한 멀티미디어 서비스가 <그림 17>에 간단하게 모형화되어 있다[10]. 그림에는 2개의 분리된 호가 존재하며 호 1에는 사용자 A, B, C가 참여하고 호 2에는 사용자 C, Z가 참여한다. 호 1은 3자간호이지만 호 2는 단방향 호이다. 사용자 A는 3자간호에 참여하고 있으나 사용자 B만을 상대방으로 인식하고 있다. 사

용자 B는 A와 C를 상대방으로 인식하고 B의 입장에서는 Z는 보이지 않는다. 즉, A나 B는 호 2를 알지 못한다. 사용자 C는 호 1과 호 2에 모두 참여하므로 사용자 A, B, Z를 모두 인식한다. 이러한 호의 모형은 B ISDN에서 멀티미디어 통신 서비스의 일반적인 경우를 표시하고 있다. 다시 사용자의 입장에서 망측에 통신 서비스를 요구하는 상황이 <그림 18>에 그려져 있다. 그림에서 User Request Manager는 사용자 신호 엔터티와 peer to peer 통신을 하는 Network Agent이다. 이것은 사용자의 신호를 분석하고 Derived Service Manager에게 트랜잭션을 요구한다. Derived Service Manager는 서비스 요구사항을 분석하고 Transport Resource Manager가 사용하는 망 자원(스위칭, 링크)의 할당 상황을 감시한다. 여기에서 정의되는 호의 모형은 특정 통신 서비스에 독립적인 몇 가지 기본적인 객체를 이용하여 이루어진다. 기본적인 객체들은 기존의 동등과 동등한 호, 사용자(party), 연결, 채널 등을 의미하는데 단어의 혼동을 피하기 위하여 새로운 단어 'Local View'를 사용하며 Local View는 호의 기본적인 객체를 기술한 문맥이다. Local View의 인스턴스 생성은 Q.921의 TEI(Terminal Endpoint Identifier) 할당 절차에 근거한 배타신호에 관련된다. 즉, 이 절차는 사용자 신호 엔터티와 망의 신호 엔터티(User Request Manager)를 결합시켜 주고 둘 사이의 잠재적 신호를 위한 채널을 할당한다. Local View의 인스턴스 생성은 호의 기본 객체로 표현하면 <그림 19>와 같다. 객체간의 관계를 나무(tree)로 표현할 때 Local View의 기본 객체는 뿌리에 해당하며 'Local Vertex'를 정의함으로써 비롯된다. 이렇게 되면 Local View는 기본 객체의 상태와 기본 객체의 동작 기준, 그리고 기본 객체에 등록된 서비스 요구로 정의될 수 있다. 기본 객체는 다시 'Local Object, Confirmed Object, Virtual Object'으로 나누어지는데 다음 객체 모형과

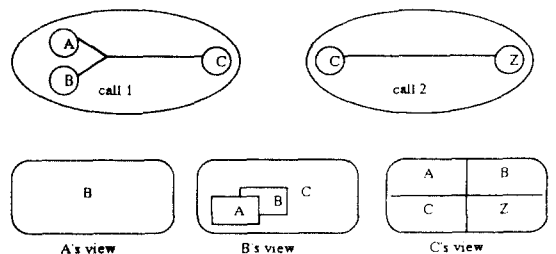


그림 17. 비디오통신을 이용한 멀티미디어 서비스 모형

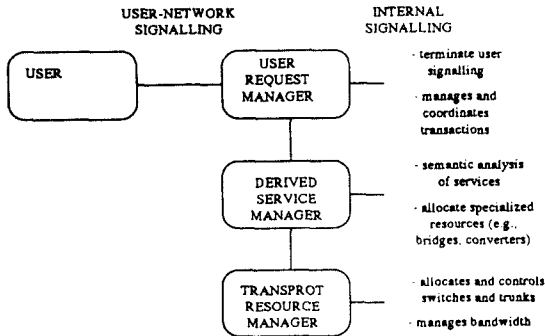


그림 18. 통신서비스 요구 상황 모형

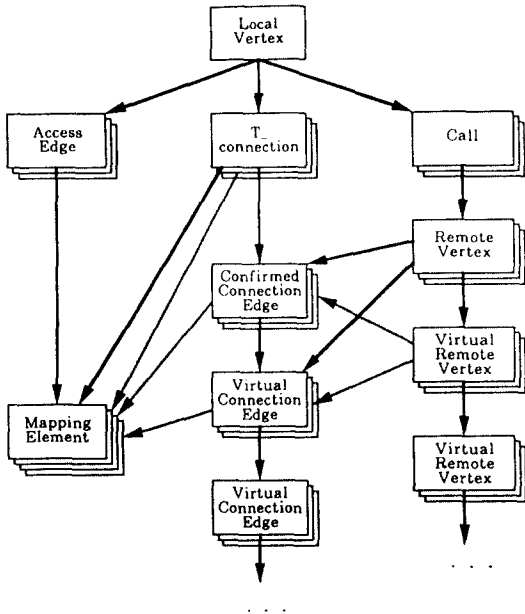


그림 19. 호의 기본 객체 모형

다른 점으로는 기본 객체중에서 Call, T_connection, Access Edge가 종속 관계가 아닌 수평 관계로 규정된다는 점을 들 수 있다. <그림 15>와 <그림 16>에서는 호가 다른 객체의 상위 객체로 정의되어 호의 나무 모형에서 뿌리에 위치하는데 <그림 19>에서는 그렇지 않다는 점이다. 실제로 통신 프로토콜 소프트웨어의 차원에서 본다면 소프트웨어의 재사용성과 유지보수성 등이 설계의 관건이 될 수 있고, 통신망의 성능 차원에서는 실시간 처리가 주요한 설계 요소가 될 수 있다. 따라서 설계자는 통신 프로토콜 소프트웨어

의 성능과 유지보수성 및 재사용성을 함께 고려하여 객체를 설정해야 하며 앞의 예제와 같이 기본적인 객체를 설정하고 통신 프로토콜 소프트웨어의 전개를 통하여 어떤 객체 설정이 가장 적절한가를 연구해야 할 것이다.

V. 통신망의 객체지향 모형

지금까지의 객체지향 모형은 통신 서비스와 그의 구현을 위한 통신 프로토콜 소프트웨어에 관한 것이었다. 여기서는 통신 서비스를 '통신망에 분산된 객체간의 메시지 교환'이라는 측면에서 분산 환경의 통신 서비스를 위한 플랫폼 개념을 도입한 통신망의 모형을 소개한다. <그림 20>에 통신망의 객체지향 모형이 그려져 있다[11]. 그림에서 각 노드는 네가지 층으로 구분되어 있다. 맨위층은 응용층이고 두번째 층은 응용 기능에 공통된 서버기능을 담당한다. 세번째 층은 Kernel층이고 마지막 층은 물리전달 층이다. 이 그림은 병렬수행(concurrent) 객체 모형에 기반하며 객체는 능동 객체와 수동 객체로 구분된다. 능동 객체들은 동시에 수행되면서 메시지를 비동기로 주고받으며 수동 객체는 능동 객체의 메시지를 받았을 때에만 작동된다. 이 그림은 새로운 통신 서비스를 계속적으로 수용하기 위한 요구 사항을 객체간의 메시지 교환 모

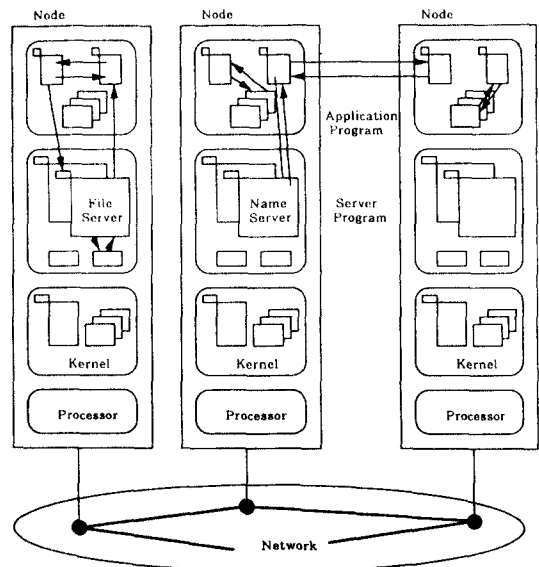


그림 20. 통신망의 객체지향 모형

형으로 제시한다. 다시 말해서 객체간의 메시지 전달 (passing) 기능에 요구되는 높은 성능(throughput)과 최소의 지연시간, 그리고 높은 신뢰도 등을 노드내의 메시지 전달과 노드간 메시지 전달로 나누어 방법론을 제시한다. 또한, 이 모형이 B-ISDN의 호 처리에 요구되는 호와 베어러의 분리제어와 UPT 서비스의 경우에도 효과적으로 적용될 수 있음을 예시한다.

VI. 통신 서비스를 위한 객체지향 프로그래밍 언어

통신 서비스 응용기능과 같은 대형 통신 프로토콜 소프트웨어의 구조연구에 객체지향 프로그램 패러다임은 최근 많은 관심속에서 활발히 연구, 적용되고 있다. 국내에서는 CCITT가 권고한 언어인 CHILL이 TDX 10 개발부터 적용되어 친숙한 상태이고 ETRI에서는 교환 시스템 소프트웨어 개발환경의 목표로 Object CHILL의 제공을 계획하고 있다[12]. 또 다른 객체지향 언어의 적용으로서 객체지향 패러다임을 이용한 시스템 명세 언어인 TOSS를 이동 전화 시스템에 적용한 사례도 있다[13]. 여기서는 Siemens사에서 ISDN 교환 시스템 개발에 사용된 Object CHILL 적용사례를 간략하게 살펴본다[14]. Object CHILL은 CHILL의 기능에 객체지향 성질을 추가한 것으로 대규모 명령수행 소프트웨어 시스템에 적용하기 위한 것이다. Object CHILL의 설계 사상에는 Ada, C++, Eiffel 등이 참고되었으며 Object CHILL과 CHILL의 관계는 대략 다음 등식으로 표현될 수 있다.

$$\begin{aligned} \text{Object-CHILL} &= \text{CHILL} \\ &+ \text{Classes} \\ &+ \text{Inheritance} \\ &+ \text{Genericity} \\ &+ \text{Assertions} \end{aligned}$$

Object CHILL의 클래스는 CHILL의 변수나 상수, 이들을 관리하는 모드와 메소드로 구성된다. 메소드는 CHILL의 procedure와 유사하며, 클래스의 정의는 specification과 body로 나뉜다. specification은 class의 인터페이스를 export, import, internal로 나누어 정의하며 body는 메소드의 구현 부분이다. 아직까지 Siemens사의 Object CHILL은 다중 상속(multiple inheritance) 지원하지 않는다. CHILL의 명령수행성은 계속 지원되므로 같은 클래스의 객체가 동시에 수행될 수도 있다. Genericity는 Ada의 generic unit 개념을

이용하여 구현되었다. 이 성질은 'parametric polymorphism'이라고도 불리우며 프로그램 구성품(component)을 매개변수로 전달하는 (parameter passing) 기능으로써 여기서는 CHILL의 모드를 매개변수화 하였다. Assertion은 CHILL의 기능을 확장하여 클래스에 관련된 invariant, 메소드에 관련된 precondition, postcondition 등의 불리언 표현이 추가되었다.

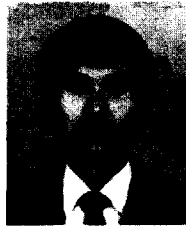
VII. 결 론

지금까지 통신 서비스와 통신 프로토콜 소프트웨어의 객체지향 모형에 대하여 대략 살펴보았다. 미래의 통신망에 요구되는 신호 기능의 모형과 통신망의 객체지향 모형도 간략히 언급되었다. 그러나 본 고에서 예시된 모형들은 일반적인 방법론적 시각에서의 묘사라고 볼 수 있다. 예를 들어, 통신 서비스의 모형화에도 본 고에서 예시된 모형외에 얼마든지 많은 모형이 있을 수 있고, 어떤 모형이 우수한가는 어느 모형이 대상 통신망과 통신 서비스를 기능의 중복없이 완벽히 묘사하는가를 검증하면 알 수 있다. 통신 프로토콜 소프트웨어와 같은 대규모 소프트웨어에 객체지향 모형의 적용이 최종점이 활발히 진행되고 있는 이유는 무엇보다도 객체지향 모형의 소프트웨어 재사용성과 유지보수성이 우수하기 때문이라고 하겠다. 그러나 통신 서비스나 통신 프로토콜 소프트웨어에 표준화된 객체지향 모형은 없으며 특히 멀티미디어 서비스 분야에는 아직 많은 연구가 필요하다. 앞에서도 언급된 바와 같이 실시간 처리가 요구되는 통신망 환경에서는 객체 설정 방법에 따른 메시지 흐름이 망 전체의 성능에 영향을 미칠 수 있으므로 통신 프로토콜 소프트웨어의 설계시 재사용성 및 유지보수성, 성능의 두 측면을 함께 고려하여야 할 것이다.

참 고 문 헌

1. John Microp, Stefan Tax, Ronald Janmaat, "Service Interaction in an object-oriented Environment," IEEE Communications Magazine, Aug. 1993
2. Wolfgang Guenther, Gerd Wackerbarth, "Object Oriented Design of ISDN Call-Processing Software," IEEE Communications Magazine, Apr. 1993
3. Hamish M. Blair, "Attacking product complexity: broadband call control for Vision O·N·E," Special ISS '92

4. 김한경, "ATM 교환 소프트웨어 기술" 전자공학회지 1993.12
5. Steven E. Minzer, Dan R. Spears, "New Directions in Signaling for Broadband ISDN," IEEE Communications Magazine, Feb. 1989
6. Jonathan S. Turner, "New Directions in Communications," IEEE Comm. Mag. Oct. 1986
7. Katrin Braun, et al, "Universal Personal Networking," IEEE ICUPC "The Conference on PCS," Oct. 1993, Ottawa Canada
8. 최병규, 강명호, 송주석, "멀티미디어호의 연결관리에 필요한 객체 설정 연구," 정보과학회 1994년도 춘계 학술발표회 제출논문
9. Seungchul PARK, Yanghee CHOI, "An End-to-End Connection Management Protocol for Multimedia Multipoint Communications," GLOBECOM 1993
10. Steven Minzer, "A Signaling Protocol for Complex Multimedia Services," IEEE Journal on Selected Areas in Communications, Dec. 1991
11. Satoshi Tanaka, Minoru Kubota, Katsumi Maruyama, Nicolsa Raguideau, "Efficient Message Passing for Network-wide Distributed Object-oriented Processing," APCC(Asia-Pacific Conference on Communications) Aug. 1993, Taejon Korea
12. 최완, 송영기, 김영시, "CCITT 언어들에 기반을 둔 소프트웨어 개발환경," 제2회 통신정보합동 학술대회 논문집, 1992년 4월
13. Soo-Dong Kim, "An Object-oriented Specification of Mobile Telephone Systems," 제2회 통신정보합동 학술대회 논문집, 1992년 4월
14. Jurgen F.H Winkler, George Diebl, "Object CHILL-An Object Oriented Language for Telecom Applications," ISS 1992



최 병 규

- 1962년 6월 8일 생
- 1985년 2월 : 연세대학교 전자공학과 학사
- 1992년 8월 ~ 현재 : 연세대학교 전산학과 석사과정 재학중
- 1985년 1월 ~ 현재 : 금성정보통신(주) 선임 연구원



송 주 석

- 1953년 3월 2일 생
- 1976년 2월 : 서울대학교 전기공학과 학사
- 1979년 2월 : 한국과학원 전기전자공학과 석사
- 1988년 8월 : Univ. of California at Berkeley 전산학 박사
- 1979년 3월 ~ 1982년 2월 : 한국 전자통신 연구소 선임 연구원
- 1989년 3월 ~ 현재 : 연세대학교 전산학과 부교수