

이기종 분산처리환경상에서 연결관리 객체의 정보공유

正會員 신 영 석*, 오 현 주*

A Sharing Scheme for Connection Management Objects in Different Distributed Processing Environments

Young Seok Shin*, Hyeon Ju Oh* *Regular Members*

요 약

최근 객체지향(Object Oriented) 설계 개념을 도입하여 신규 멀티미디어 서비스를 신속하게 제공하기 위해서는 통신망 관리와 서비스관리가 통합된 개방형 정보통신망 구조(Open Networking Architecture)가 요구되고 있다. 통신망 관리와 서비스 관리가 통합된 정보통신망 구조는 객체지향 설계와 분산처리환경(Distributed Processing Environment) 기술을 도입하여 통신망 구성장치와 독립적인 소프트웨어의 계층적인 구조로 개방형 정보통신망 구조를 모델화하여 정립하고 있다.

본 논문에서는 개방형 정보통신망의 기반이 되는 분산처리환경에서 개방형 연결관리 구조에 따른 객체의 연산 모델링과 연결관리 객체를 효율적으로 관리하기 위한 객체그룹 모델인 빌딩블럭(Building Block) 모델을 제시하였으며, 서로다른 이기종 분산처리환경에서 연결관리 객체간 정보를 공유하는 방안을 제시하였다.

ABSTRACT

Open networking architecture is required to support new multimedia services as integrated functions of network management and service architecture.

In this paper, we propose the methodology of building block modeling using object grouping concepts and the sharing scheme of different distributed processing environments based on open networking architecture. The building block has the functions of object management, security object instance registry and object mapping in object group. It is necessary for the connection management information to be shared on the interworking between two domains. We implemented and validated connection management functions using computational object modeling and building block modeling in different distributed processing environments.

* 한국전자통신연구원 광대역통신망 연구부
論文番號: 96349-1105
接受日字: 1996年 11月 5日

I. 개 요

최근 통신망 사용자들의 다양한 멀티미디어 서비스 요구와 사용자 단말기의 대용량화 및 고기능화에 따른 멀티미디어화 추세, 통신망 사업자의 효율적인 유지 보수 및 운용의 추구하고 복수 통신사업자와 양상에 따라, 앞으로의 통신망은 점점 복잡해지고 있다. 이러한 통신망들이 복잡해질수록 망관리 및 신규 서비스의 수용은 더욱 어려워지며, 효율적인 통신망의 관리와 성능을 추구할 수 없게 되므로, 앞으로의 통신망은 새로운 서비스 제공에 따른 망관리와 서비스 구조의 기능이 통합된 정보통신망 구조로서 stakeholder(예, 통신사업자, 서비스 제공사업자, 사용자, 서비스중개자, 서비스 개발자들을 지칭) 개념으로 개방형 정보통신망 구조(Open Information Networking Architecture)의 정립이 필요하게 된다^{1,2)}.

개방형 정보통신망(이하 개방형통신망) 구조는 기존의 통신망 혹은 망구성 장치에 큰 변화없이 새로운 서비스 적용에 따라 망구성 요소장치에 크게 의존하지 않는 통신 소프트웨어 구조의 정립을 추구하고 있으며, 이와 같은 연구와 표준화는 국제 컨소시엄인 TINA-C(Telecommunication Information Networking Architecture-Consortium)에서 연구중에 있다. 개방형 통신망 구조는 기본적으로 기존의 통신망 기술을 충분히 수용하고, 앞으로 출현하게 될 다양한 멀티미디어 서비스를 쉽게 적용할 수 있는 정보통신망 구조를 위하여 분산처리환경(Distributed Processing Environment: DPE)상에서 객체(Object)나 객체그룹 단위로 망관리 객체와 서비스관리 객체를 공유하거나 접속할 수 있는 소프트웨어의 계층적 구조로 구성된다^{3,4)}. 이러한 개방형통신망은 복수의 통신사업자들이 서로 다른 분산처리환경을 구축하는 경우에, 객체들간의 정보공유나 망관리 및 서비스 제공에 별도의 객체들간의 연동이 필요하게 된다.

본 논문에서는 개방형통신망 구조로 설계되고 운용될 통신망 관리기능에서 서로 다른 이기종 분산처리 환경으로 통신망관리가 구축될 경우, 2개의 서로 다른 통신사업자 영역(Domain)간의 연결관리 객체(Connection Managed Objects)들의 정보공유를 가능하게 하는 방법을 제시한다. 이러한 객체정보 공유방법은 객체지향 설계개념에 입각하여 효율적인 객체관리와

연결관리 구조의 기능을 블럭화하기 위하여 객체그룹을 위한 빌딩블럭(Building Block) 모델이 요구되며, 이를 서로 다른 이기종의 분산처리환경에서 연결관리 연산모델(Computational Model)에 적용하여 제안된 정보공유 방안을 검증하였다. '97년말까지 기본 프레임워크(Framework)의 표준화 규격 작성을 목표로 하는 개방형통신망 구조 연구는 아직 완전한 구조가 정립되지 않은 관계로 그동안 연구되었던 기술을 바탕으로 단계적인 기술검증이 수행되고 있다. 특히 TINA 컨소시엄 회원기관과 유럽의 RACE ACTS 연구참여 기관들, EuresCOM, Alcatel, BT, NTT등의 선진 연구기관에서는 미래의 다양한 멀티미디어 서비스를 수용하기 위하여 개방형통신망 구조에 대한 정립과 기술 검증을 목적으로 분산처리환경과 객체지향 설계 개념에 의한 대표적인 멀티미디어 서비스를 선정하여 프로토타이핑을 함으로서 개방형통신망 기술을 축적하고 있다.

본 논문은 개방형통신망 구조를 기반으로 하는 통신망 구조에서 객체지향 설계 개념에 입각한 정보공유 모델과 객체그룹 모델링에 관한 내용으로서, 제2장에서는 개방형 연결관리 구조에 대한 개략적인 구조와 개념을 소개하고, 제3장에서는 제2장에 소개된 모델링 개념을 적용한 연결관리 객체의 정보공유 기능 구조에 대해 살펴본다. 제4장에서는 이에 대한 엔지니어링 모델 구현 측면에서 빌딩블럭 구조와 두 영역간의 분산처리환경에서 정보공유 구조와 연동구조의 상세 기능과 절차에 대해 살펴보고, 제5장에서는 결론을 맺는다.

II. 개방형 연결관리 구조

1. 개방형 정보통신망 구조

개방형통신망 구조는 소프트웨어의 계층적 구조를 바탕으로 망구성장치와 독립적으로 초고속전달망인 하드웨어 위에서 통신망관리와 서비스 관리 기능 모듈로 구성된다. 개방형통신망 구조는 분산시스템과 객체지향 설계기술, 통신망과 서비스관리의 통합기술로 구성됨에 따라 분산처리환경 계층, 사용자와 서비스 제공에 따르는 서비스 구조와 망관리 계층 구조로 나누어진다⁵⁾. 분산처리환경은 통신 소프트웨어 객체의 구성요소들을 분산 수용하여 기능을 수행하는 개

방형통신망의 기반 기술로서, 객체지향 설계개념을 도입하여 통신 소프트웨어 구조상에서 객체들의 모델링 개념을 적용하여 재사용성, 상호 운용성 및 분산처리환경을 설계, 구축하기 위한 컴퓨팅 플랫폼에 대한 정의 및 규정뿐만 아니라, 객체를 설계하고 구축하는데 대한 개념을 논리화하는 모델링등의 설계 개념을 정립하는 계층이다. 통신망 관리구조는 통신망의 관리 서비스와 구조에 적용되는 개념과 원칙을 정의하여, FCAPS(Fault, Configuration, Accounting, Performance and Security Management)기능에 대한 기술을 관리하는데 사용될 객체에 관련된 개념과 원칙을 규정하는 계층이다.

서비스 구조는 앞으로 출현하게 될 멀티미디어 서비스를 제공할 수 있는 일반적인 서비스 구조를 정립하는 계층으로 호(Call)와 연결(Connection)의 분리원칙과 세션(Session) 개념을 정립하여 개방형통신망을 구성하는 stakeholder에 따른 액세스 세션, 서비스 세션, 통신 세션으로 구분하여 서비스 구조를 일반화하였다. 이러한 서비스 구조의 규격, 설계, 구현과 관리

에 관련된 일련의 개념과 원칙을 규정화한 구조로서, 영상회의, 가상 쇼핑, 가상대학등의 미래에 규정될 모든 형태의 멀티미디어 서비스의 구조를 포함한다. 이와 같은 개방형통신망 기법을 기반으로 TINA 컨소시엄에서는 서비스 구조와 망관리 구조를 제시하고 있으며,^[2, 7, 11] 그림 1은 일반적인 망관리 구조를 나타내었다. 망관리 구조인 그림 1은 기본적으로 망관리 객체들간의 오퍼레이션의 접속을 위하여 분산처리환경들간의 접속 네트워크인 커널망(KTN: Kernel Transport Network)이 있으며, 분산처리환경상에서 망관리 기능의 객체들은 응용 서비스로 운용된다. 본 논문에서는 FCAPS의 여러 통신망 관리 기능중에서 가장 기본이 되는 연결관리(Connection Management: CM) 기능을 대상으로 한다.

개방형통신망의 망관리 구조에서 정의되는 연결관리 기능의 컴포넌트는 통신세션 관리자(Connection Session Manager: CSM), 연결설정 협력자(Connection Coordinator: CC), 계층망 협력자(Layer Network Coordinator: LNC), 망관리 레벨 연결실행(Network Man-

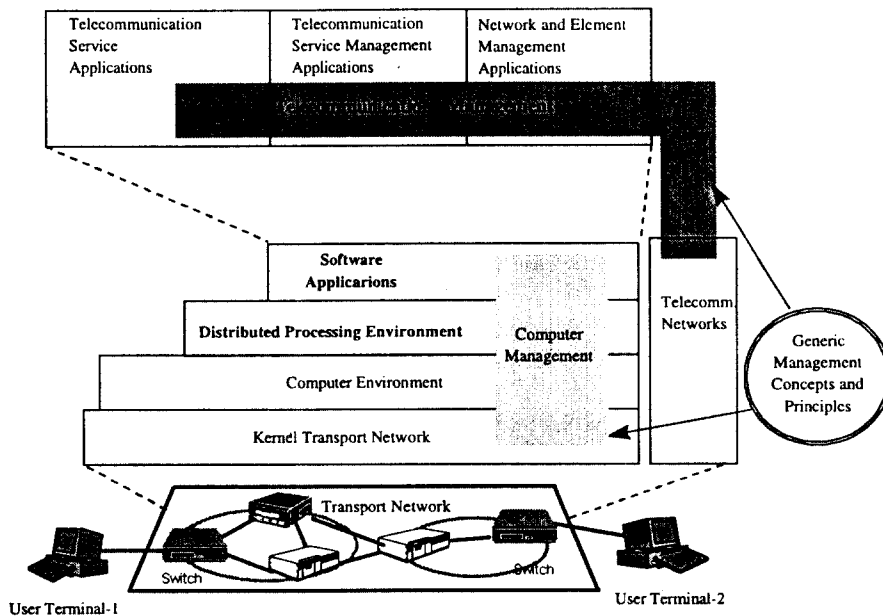


그림 1. 개방형 정보통신망의 관리 구조의 개념도
Fig. 1 Management Architecture on Open Networking Architecture

agement Level Connection Performer: NML-CP), 소자관리 레벨 연결 실행(Element Management Level Connection Performer: EML-CP)의 객체 컴포넌트로 구성된다. 이들의 연결관리 객체들은 계층적으로 분산처리환경에서 상호간 접속되며, 서로 다른 영역의 서브네트워크(SubNetWork: SNW)간에는 분산처리환경의 페더레이션(Federation) 기능을 통하여 동등한 레벨의 LNC 객체를 통하여 연결된다. 그림 2는 동일한 통신사업자 영역에서 연결관리 컴포넌트의 연산모델(Computational Model)을 나타내었다.¹¹⁾

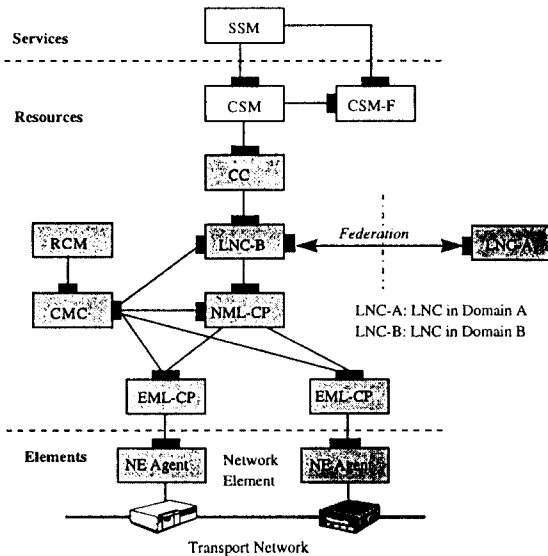


그림 2. 연결관리 컴포넌트의 연산모델
Fig. 2 Computational Model of Connection Management Component

2. 연결관리 객체 모델링

연결관리 객체 모델링은 TINA 컨소시엄에서 제안한 정보모델링(Information Modeling)과 연산모델링을 따른다. 정보모델은 연결관리 객체를 규정하는 것으로 정적인(Static) 객체와 동적인(Dynamic) 객체로 구분하여 TMN(Telecommunication Management Network)에서 많이 사용하는 GDMO(Guideline of Definition Managed Object) 표기법과 같은 형태로 quasi-GDMO + GRM과 그래픽형태의 OMT(Object Management Technique) 방법에 따라 기술하였다. 연산모

델은 TINA OD(Objct Definition Language)를 사용하여 객체 및 객체그룹들간의 인터페이스 개념을 정립하였다.

2.1 정보모델링

정보 모델링은 연결관리 객체정보 규격을 위한 프레임워크(Framework)을 제공하며, 각 정보객체들이 시스템에서 해결하여야 하는 연결관리 기능의 범위를 기술한다. 정보모델링 개념은 정보객체(엔터티) 정의, 객체 타입과 속성, 엔터티간의 관계와 객체 생성과 삭제등의 규칙으로 구성된다. 본 논문에서는 연결관리 객체를 TINA 컨소시엄에서 규정한 정적인 객체로는 LN(Layer Network), SNW(SubNetWork), NMCTP (NetWork Connection Termination Point), NWTP (NetWork Trail Termination Point), LTP(Link Termination Point)으로 모델링하며, 동적인 객체로는 Edge, Trail, SNC(SubNetwork Connection), CNX (CoNnection), TC(Tandem Connection)등으로 규정하여 연결관리 객체를 모델링하였다.

2.2 연산 모델링

연산모델 개념은 연산규격을 위한 프레임워크를 제공하며, 연산객체(Computational Object: CO)로 된 분산된 통신 응용서비스와 연산객체간의 상호 연동(Interworking)에 대한 방법을 기술한다. 연산객체는 프로그래밍과 캡슐(Capsule)화 단위이다. 객체들은 상호간에 인터페이스를 주고 받으면서 상호연동을 한다. 객체간의 접속은 각 오퍼레이션 접속(Operation Interface)과 스트림 접속(Stream Interface)으로 구분된다. 오퍼레이션 접속은 객체내에서 오퍼레이션으로 정의된 것으로서 서버 객체의 기능을 클라이언트 객체가 호출하여 사용하며, 오퍼레이션은 매개변수(객체 속성)들을 가질 수 있다. 스트림 접속은 오퍼레이션이 없는 것으로서 요구된 QoS(Quality of Service)를 수용하면서 실시간 비디오나 오디오와 같은 정보를 전달하는 채널이다.

시스템의 연산모델링은 망구성장치의 연결관리 정보모델과 일치하여야 한다. 연산모델 설계자 혹은 개발자는 연결관리 객체들이 어느 곳에 분산되어 위치하여야 하는가에 대해서 고려할 필요는 없으나, 엔지니어링 모델링 개념으로 분산처리환경에서 응용계층

구성에 대한 프레임워크를 제공한다. 이러한 개념들은 분산될 응용계층의 구조를 규정하며, 연산객체들이 어떻게 인프라(Infrastructure)에서 실행을 위해 적용 모델이 엔지니어링 모델이다. 연산객체가 적용되는 단위로서 분산처리환경을 제공하는 자원관리의 유니트인 분산처리환경 노드와 노드의 부분집합으로서 자원 할당과 캡슐화의 단위인 캡슐, 서로 같이 위치하게 되는 객체들의 그룹인 클러스터(Cluster)로 세가지 단위를 정의한다. 본 논문에서는 정보 모델링과 연산 모델링에 입각하여 소규모의 통신사업자 영역으로 서로다른 분산처리환경의 객체정보 공유 시험을 목적으로 함에 따라 엔지니어링 모델을 생략한다. 또한 연산객체와 각 연결관리 컴포넌트 구조에서 여러 연결관리 객체를 생성함에 따라, 연산모델 객체를 객체그룹의 단위인 빌딩블럭으로 정의하여 효율적인 연결관리 객체를 관리하여 상호간 객체의 정보공유가 용의하게 하였다.

2.3 객체그룹 모델링

네트워크상에서 점대점(Point-to-point)의 연결을 위해서는 연결관리 연산객체에서 정적객체와 동적인 객체들이 생성되어 연결관리 기능을 수행한다. 이러한 객체들이 연결설정 및 해제에 따라 수시로 생성 및 소멸됨에 따라, 개별된 연결관리 객체들을 엔지니어링 모델의 캡슐단위로 객체/연산객체들을 관리하는 빌딩블럭 개념을 도입하였다.¹¹⁾ 빌딩블럭은 객체들의 집합으로써, 이들을 구성하는 데에는 일정한 규칙을 가진다. 우선적으로 빌딩블럭안에 포함되는 객체들은 분산시스템의 한 노드에 같이 위치하게 되며, 빌딩블럭은 하나의 유니트 단위로 존재하며, 객체의 상태에 따라 친이된다. 모든 빌딩블럭은 빌딩블럭안에 있는 다른 객체/연산객체들을 관리하기 위한 관리객체를 그림 3과 같이 적어도 하나 이상 포함하게 된다. 빌딩블럭의 객체그룹에서 관리객체는 객체의 생성과 삭제, 콘트랙트(Contract) 관리, 콘트랙트 서비스 속성 변경 및 보안 관리, 성능 감시 등을 말한다.

본 연구에서는 위와 같은 객체그룹의 표현인 빌딩블럭을 그림 3과 같이 적용하였으며, 그림 3의 빌딩블럭은 정보객체를 연산객체로 매핑하고, 일반적으로 적용될 연산객체 템플릿(Template)을 정의하여 일반적으로 적용될 연산객체 인터페이스 정의와 빌

딩블럭내에서 객체들의 관리를 위한 객체 매핑, 보안 레지스트리(Registry), 객체 인스턴스 레지스트리 등을 정의하였다.

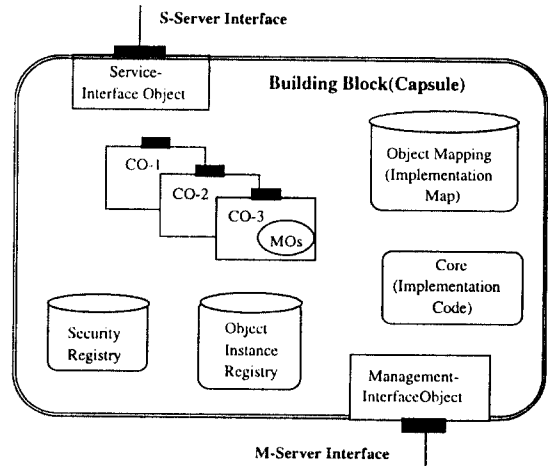


그림 3. 일반적인 빌딩블럭의 연산모델 컴포넌트
Fig. 3 Generic Computational Model of Building Block

- 서비스 인터페이스 객체는 빌딩블럭내에서 연산객체들간의 오퍼레이션 접속 서비스를 제공하며, 접속하는 외부의 객체 속성과 유일한 객체 이름을 근거로 빌딩블럭내 연산객체들과 접속된다.
- 관리 인터페이스 객체는 빌딩블럭내에서 객체 관리서비스를 제공하며, 제공되는 관리 서비스는 연산객체의 생성 삭제 검색 서비스(예 : Create, Destory, Enable, Lookup, Register, Withdraw, Add, Remove, Update, Lookup_access_rule)이며, 연결관리 객체들과 연산객체(혹은 CORBA 객체)간 정보 매핑을 관리하거나 참조하는 서비스를 제공한다. 또한 빌딩블럭내의 보안 정보에 대한 접근 제어 목록이 M-서버 인터페이스를 통하여 제어된다.
- 객체 인스턴스 레지스트리는 관리 인터페이스 객체를 통하여 생성된 객체관리 정보객체를 저장한다. 각 객체관리 정보객체는 정보객체 이름과 연산객체 레퍼런스 사이를 연결시켜주는 정보의 관리 및 연산객체 상태를 관리하며, 이때 정보객체 이름은 관리객체의 구별된 이름이다.
- 보안 레지스트리 객체는 빌딩블럭내에서 정보객체

인스턴스와 대응되는 ACL(Access Control List) 객체를 저장한다. ACL 객체는 오퍼레이션 목록으로 구성된 객체의 접속 및 오퍼레이션에 대한 권한에 관한 정보를 포함한다.

- 구현(Implementation) 맵(Map) 객체는 구현 객체 정보를 저장하며, 구현객체 정보는 빌딩블럭내에서 연산객체의 템플릿 이름과 CORBA 객체 정의사이의 매핑정보를 관리한다.

III. 연결관리의 기능 구조

1. 연결관리 객체 기능

통신세션 관리자(CSM)는 서비스 컴포넌트에게 네트워크 자원할당을 위한 통신세션 관리의 인터페이스를 제공하며, 서비스 제공에 따른 물리적인 연결을 위한 논리적 연결 그래프(Logical Connection Graph: LCG)과 물리적 연결 그래프(Physical Connection Graph: PCG)를 새로이 구성하거나 수정하는 동작으로 구성된다. CSM은 일단 LCG가 만들어지면 이것을 PCG로 바꾸며, 변환하는 과정에서 제일 중요한 작업은 연산객체들이 위치할 연산노드들과 앞으로 사용될 망접속 기준점들을 정의하는 것이다. CSM은 망 접속점사이에 연결을 설정하기 위해 CC와 상호연동을 수행하며, 서비스로부터 요구되는 대역폭 수정이나 요청 해제와 같은 작업을 수행한다.

CC는 CSM으로부터 요구되는 연결요청을 수행하며, 특정 계층망에 종속되지 않고 CSM에게 접대접 연결을 위한 주도적인 제어역할을 유도한다. CC는 CSM으로부터 대역폭과 특정 QoS를 가진 가진 연결요청을 받으면 이것에 근거하여 어떤 통신영역에 사용될 LN을 결정하여, 해당 LN에 연결요청을 전달한다.

일반적으로 전달망들은 다양한 특성과 지역적인 구성에 따라 여러개의 LN으로 구성된다. 이러한 LN은 LNC와 네트워크 레벨의 제어 계층인 CP(Connection Performer)로 제어 관리된다.

NML-CP는 한개의 LN에서 서브네트워크를 관리하는 객체로서 SNC, CNX의 연결관리 객체를 제어 및 관리하며, NML-CP에는 별도의 네트워크 계층이 존재하게 된다. 망구성장치와 물리적으로 매핑되는 객체그룹인 EML-CP는 네트워크 레벨하단의 서브네트워크에서 SNC를 관리하는 객체 기능으로 정의된다.

2. 연결관리 객체 구조

본 연구에서는 CC를 제외한 연결관리 구조 모델을 그림 4와 같이 3개의 서브네트워크와 1개의 네트워크로 구성하였다. RCM(Resource Configuration Management)에서는 연결관리 구성을 위한 네트워크 모델을 감지하여 이를 CMC(Configuration Management Coordinator)에게 공지한다. CMC에서는 RCM에서 구성한 네트워크 모델에 적응하는 연결관리 객체를 생성하여 연결관리 기능을 수행한다. 구성된 연결관리 연산객체는 연결관리 레벨의 관리객체들을 ODL로 기술하여 설계하였다. 본 논문에서는 이기종 분산처리환경에서의 연결관리 객체정보 공유를 위하여 연결관리 적용 서비스로는 1) Reserve TC, 2) Setup Trail, 3) Release Trail, 4) Undo_ReserveTC 등에 관해 객체간 정보흐름으로 객체 정보 공유를 시험하였다.

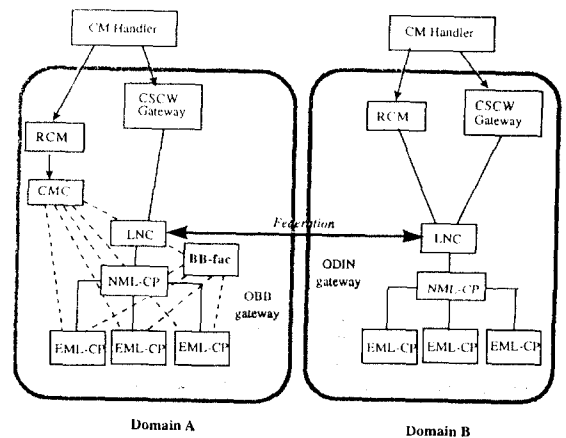


그림 4. 연결관리 객체들간의 연산모델 관계
Fig. 4 The Computational Relationship of Connection Management Objects

IV. 연결관리 객체 정보공유

1. 분산처리환경

본 연구에서는 ObjectBroker를 사용하여 객체간 인터페이스와 구현을 정의하고, 클라이언트와 서버를 구축하였으며, 이를 ANSAware 트레이더(Trader)에 등록하여 이름(Naming)서비스를 수정 보완하여 트레이더 기능을 구성하였다. ObjectBroker는 OMG(Ob-

ject Management Group) CORBA 규격을 DEC 회사에서 개발한 CORBA Compliance 제품을 사용하여 빌딩블럭 개념을 적용하여 연결관리 기능을 설계하였다. 본 분산처리환경은 ATM 통신망(Fore사의 ASX-200 ATM 스위치)를 기반으로 구축하였으며, ATM 연결은 연결관리에서 점대점의 PVC(Permanent Virtual Circuit)로 설정하였다. 이종 분산처리환경으로는 DEC사의 ObjectBroker(OBB Ver. 2.5.)와 이탈리아 CSELT에서 개발한 ANSAware기반의 ODIN(Object Distributed INterface)간에 객체정보 공유를 위한 연동 시험환경으로 설정하였다.

2. 이기종 분산처리환경에서 연동 구조

연결관리 적용 서비스를 수행시 2개의 영역에 걸쳐 있는 연결관리 객체인 경우, 페더레이션 기능이 필요하게 된다. 이것은 2개의 통신사업자 영역에 있는 분산처리환경사이에서 연결관리 객체들간의 연동이 존재하게 되며, 이기종 분산처리환경간에 연결관리 객체의 정보를 공유하는 기능이 필요하다. 그림 5은 2개의 통신사업자 영역에서 LTP1에서 LTP6간의 트레일인 점대점의 연결을 설정하는 경우, 연결관리 객체에 의한 Setup Trail 서비스의 과정을 보여주고 있다.

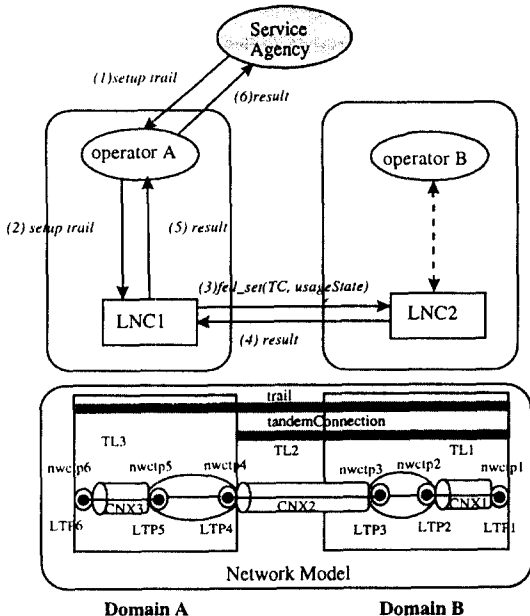


그림 5. Setup trail 서비스 흐름
Fig. 5 The Flow of Setup Trail Service Sequence

그림 5에서는 통신사업자 영역 A와 통신사업자 영역 B에 걸쳐 사용자간의 연결을 통신 서비스 에이전시에서 통신영역 A의 통신망 운용자에게 setup trail 서비스를 요청하는 경우, 2개의 영역간에 연결관리 객체의 정보공유가 요구된다. 따라서 LN 계층에서 2영역간의 정보공유를 위하여 트레일 설정에 따른 fed_set(TC, UsageState) 오퍼레이션은 영역 B의 LNC 빌딩블럭내의 TC에 대한 오퍼레이션을 호출하게 된다. 이때 위의 두 영역간 연결관리 객체간의 인터페이스를 위한 분산처리환경 레벨의 게이트웨이 기능이 필요하게 되며, 게이트웨이의 구조는 그림 6과 같으며, 두 영역에서 게이트웨이의 세부 기능은 다음과 같다.

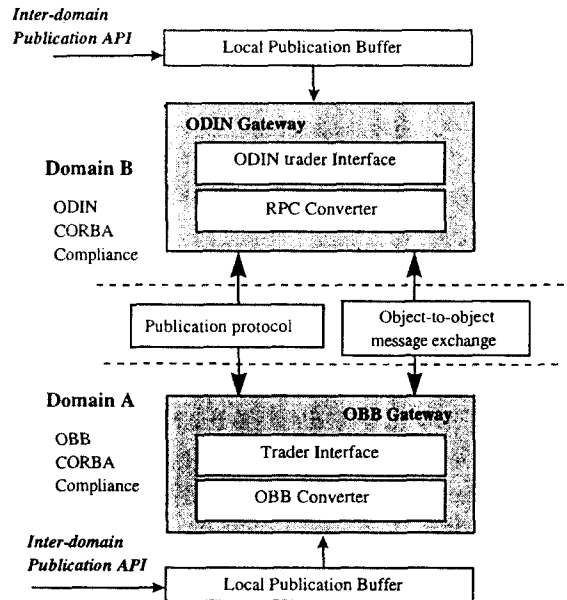


그림 6. 이기종 분산처리환경의 객체정보 공유를 위한 게이트웨이 구조
Fig. 6 The Gateway Configuration for Object Sharing in two Different DPEs

- ODIN 게이트웨이

단일 멀티쓰레드 프로세스 구조로서 공지 프로토콜 객체나 오퍼레이션이 OBB 도메인으로부터 접근가능한 ODIN 객체 인스턴스의 콘트랙트를 알려준다. 또한 동일한 프로토콜을 사용해서 OBB 게이트웨이는 OBB 객체 인스턴스의 어떤 콘트랙트가 영역 B에서 접근 가능성을 알려준다. ODIN 트레이더 인터페이스

스는 게이트웨이를 통해서 접근할 수 있는 OBB 객체의 ODIN 콘트랙트 레퍼런스를 공지(Publish)한다. RPC 변환기는 OBB 도메인으로부터 온 ODIN 객체의 트레이더에 대한 요구를 DCE RPC 호로 변환시키고, 반대로 DCE RPC로부터 온 요구들은 객체단위의 메시지 정보변환 메시지로 바뀌어 OBB 게이트웨이로 전달된다.

- OBB 게이트웨이

단일 쓰레드 프로세서 구조로써 ODIN 게이트웨이와 같은 구조 및 기능을 가진다.

- 공지 프로토콜

객체들의 콘트랙트로부터 다른 도메인으로부터 접근가능한지에 관해 서로 통신을 하기 위해 OBB 게이트웨이와 ODIN 게이트웨이간에 사용되는 프로토콜이다.

- 객체간 메시지정보 교환

RPC 컨버터와 OBB 컨버터 사이에서 연산객체들 사이의 응용 레벨 메시지를 교환하기 위해 사용되는 프로토콜이다.

- 영역간 공지 API(Application Programming Interface)

API는 2개의 영역간에 가시화하게 나타내는 객체에 대하여 직접적으로 접속이 가능하도록 사용된다.

- 로칼 공지 버퍼(Local Publication Buffer)

분산환경의 영역간 공지를 위한 API와 게이트웨이 사이에 있는 버퍼로서, 어플리케이션의 중단이나 재시작이 없이 게이트웨이의 시동과 중단 및 복구를 가능하게 한다.

3. 시험환경 구성

분산환경에서 2개 영역간 API의 오퍼레이션은 주로 *Export_contract(in bb_template_name, in federation_info, in capsule_property, out offer_id, out status)*과 *Remove_contract(in bb_template_name, in federation_info, in capsule_property, out offer_id, out status)*로 구성된다. 연결관리 객체들의 생성과 소멸에 관한 정보를 위하여 *Export_contract()*가 수행되거나, *Remove_contract()*이 수행될 때 메시지들은 직접 게이트웨이에게 전달되지 않고 로칼 공지버퍼에 보내지며, 로칼 공지버퍼는 콘트랙트의 목록을 저장한다. 로칼 공지버퍼는 *Export_contract()*과 *Remove_contract()*에 의해 생성된 모든 메시지를 관리하며, 게이트웨이가 시

작되면 이들의 메시지를 게이트웨이에게 직접 전달한다. 따라서 이러한 기법이 사용될 경우, 응용계층 서비스의 중단없이 게이트웨이를 중지시키거나 재시작할 수 있다. 그림 7에서는 이와 같은 과정을 나타내었다.

그림 7에서 OBB 게이트웨이는 ODIN 게이트웨이에게 어떤 것이 ODIN 도메인으로부터 접근가능한 OBB 객체 인스턴스의 콘트랙트 인지를 알려준다. 이때 TCP/IP 포트와 프록시 클라이언트(OBB converter)를 생성시키고 공지 프로토콜 메시지를 ODIN 게이트웨이에게 보내며, ODIN 게이트웨이는 프록시 서버(RPC converter)를 생성시킨다. 프록시 서버에 의해 생성된 새로운 콘트랙트 레퍼런스는 ODIN 트레이더에 등록된다.

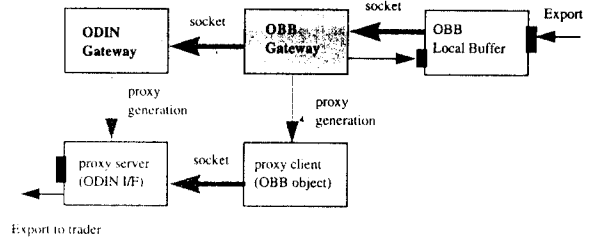


그림 7. OBB와 ODIN간의 객체정보 등록을 위한 게이트웨이 접속절차
Fig. 7 The Interface Procedure of Object Registry between OBB and ODIN Gateway

4. 시험절차 및 기능확인

이기종 분산처리환경에서 연결관리 객체 정보공유의 시험절차는 그림 5와 같이 *setup trail* 서비스를 비롯하여 트레이일을 설정, 해지하는 4개의 서비스를 선택하여 시험하였다. 따라서 그림 8은 2개의 영역간의 트레이일을 설정, 해지 서비스를 위하여 OBB 영역에서 연결관리 객체그룹인 LNC에서 점대점 트레이일 요청이 있는 경우, 동적/정적 객체를 ODIN 영역과 공유하기 위하여 객체의 레퍼런스를 상호간 게이트웨이를 통하여 트레이더에 등록하고, 상호간에 등록된 객체 레퍼런스를 검색하는 관정이다. 그림 8은 게이트웨이를 통한 연결관리 객체의 정보공유를 위한 접속절차이다.

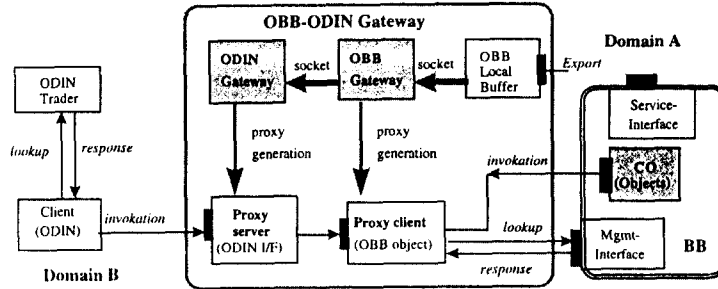


그림 8. OBB와 ODIN 영역간의 객체정보 공유를 위한 접속절차

Fig. 8 The Interface Procedure of Object Sharing between OBB and ODIN Domain

- ODIN 관리객체는 ODIN 트레이더에게 연결관리 객체 정보공유를 필요로 하는 OBB 관리객체의 콘트랙트 레퍼런스를 요구한다.
- ODIN 트레이더는 프록시 서버에 의해 생성되어 등록된 콘트랙트 레퍼런스를 제공한다.
- ODIN 관리객체는 콘트랙트 레퍼런스를 이용하여 프록시 서버에게 관리객체 공유 요구를 전달하며, 프록시 서버는 이 메시지를 객체간의 메시지 교환 프로토콜로 메시지를 변환시켜서 OBB 도메인의 프록시 클라이언트에게 전달한다.
- OBB 도메인의 프록시 클라이언트는 메시지를 받아서 OBB 영역의 동일지역(local) 오퍼레이션을 호출하게 된다.
- 동일지역 오퍼레이션의 결과에 대한 전송도 동일한 과정을 거치게 된다.

V. 결 론

개방형통신망 구조는 다양한 멀티미디어 서비스 제공과 통신망 자원관리의 설계 및 개발과 구축에 관련되는 논리적 구조와 원칙을 제시하므로써, 효율적인 망관리와 소프트웨어의 재사용으로 신속한 신규 서비스의 제공을 목표로 하고있다. 이를 위해 개방형 통신망 구조는 서비스 구조와 망관리 구조 및 분산처리환경으로 구분되어 소프트웨어의 계층적 구조로 정립되며, 객체지향 설계 개념에 따라 분산처리환경에서 새로운 객체 모델링 개념을 적용한다. 본 논문

에서는 개방형 정보통신망 구조의 기반이 되는 분산처리환경을 이용하여, 연결관리 기능에 대한 정보모델링, 연산모델링을 하였으며, 효율적인 연결관리 객체들을 관리하기 위하여 객체그룹 모델인 빌딩블럭으로 설계 구현하였다. 또한 서로 다른 이기종 분산처리환경에서 이기종간의 분산처리환경의 프로토콜 변환과 트레이딩 기능을 가지는 게이트웨이 구조로서 연결관리 객체간의 정보공유를 위한 방안을 제시하였다.

본 연구는 앞으로 개방형통신망이 구축되는 경우, 서로 다른 통신사업자가 이기종의 분산처리환경을 구축하여 서비스와 망관리가 통합된 개방형 멀티미디어 서비스 제공할 때, 서로 다른 영역간의 연결관리 객체들 사이의 연동을 위한 기능이 필요하게 된다. 따라서 이러한 연동기능을 통해 복수 통신사업자 영역에서의 연결 관리 서비스는 자동적으로 이루어지게 되며, 이를 통해 통신망 정보 및 연결관리 정보가 원활히 유통될 수 있다. 본 논문에서는 이기종의 분산처리환경을 OBB와 ODIN으로 설정하여 2개의 통신사업자 영역간에 점대점의 ATM 연결을 설정하는 연결관리 객체에 대한 정보공유를 위한 게이트웨이 구조와 객체그룹 모델인 빌딩블럭 모델을 제안하였으며, 이를 객체지향 설계와 구현으로 증명하였다.

앞으로 연구사항으로는 분산처리환경이 대규모로 구성될 때, 앞으로 출현하게될 다양한 멀티미디어 서비스를 제공하기 위하여 연결관리 객체 공유 모델과 다자간 연결 및 서비스 품질에 따른 망연결 요구로

인한 분산 소프트웨어 시스템으로서, 실시간 분산처리 기술과 객체지향 설계 관리 기능에 대한 검증과 성능분석이 요구된다. 특히 분산처리환경에서 실시간 통신망관리 기능의 객체에 대한 접속과 스트림 접속의 객체모델링은 실시간 연결관리 서비스 제공과 대규모 분산처리환경 시스템에 적용하는 객체지향 설계 및 구현과 실시간 망 구성 기술, 비연결형 서비스에 대한 연결관리 구조 모델링 등에 많은 연구가 요구된다.

감사의 글

본 연구는 TINA 컨소시엄의 WWD(World Wide Demonstration) 제안 1 그룹에 의해서 계획되어 연구되었으며, 연구결과는 스위스 제네바 Telecom '95에서 전시되었다. 본 연구와 TINA 컨소시엄 WWD 시연 및 전시회에 많은 도움을 준 한국통신, 일본 NTT, 이탈리아 CSELT 연구원에게 감사 드립니다. 또한 본 논문을 위해 내용 검토를 비롯하여 많은 조언과 기술적 문제를 협의해 주신 ETRI의 백 영식박사와 NTT 통신망 종합연구소의 Osamu Miyagishi, Sagae Chikara 연구원, DEC Japan의 Yukio Yatsudake와 CSELT의 Dr. Giovanni Matine에게 감사 드립니다.

참 고 문 헌

1. William J. Barr, Trevor Boyod and Yuji Inoue, "The TINA Initiate", IEEE Communications Magazine, March, 1993.
2. Juan Pavon, "Toward Integration of Service and Network Management in TINA", TINA-Consortium News, May, 1995.
3. Masatoshi Kawarasaki and Yuji Inoue, "Network Management toward B-ISDN", IEICE Vol. J76-B-I, No. 11, Nov., 1993.
4. Chikara S. Takita W. and Kobayashi H., "A Network Management Architecture on a Distributed Processing Environment", The proceeding of TINA'93, Sep., 1993.
5. Harvey Rubin and N. Natarajan, "A Distributed Software Architecture for Telecommunication Networks", IEEE Network, Jan/Feb., 1994.
6. N. Natarajan and Gray Slawsky, "A Framework Architecture for Information Networks", IEEE Communications Magazine, April, 1992.
7. Martin Chaoman and Stefano Montesi, "Overall Concepts and Principles of TINA", TM_MDC. 018_1.0_94, TINA-C Documents, Feb., 1995.
8. Harm Mulder and Juan Pavon, "Telecommunications Information Network Architecture", ISINM '95 Tutorial Course Notes of TINA-C, May, 1995.
9. 신영석, "Integrated On-line Management of Multi-Carrier Networks", High-Speed Network '95 Workshop 논문집, 1995. 1. 23.
10. Tatsuo Nohara, Hiroki Tanaka, Hiroshi Ishii and Makoto Yamada, "Proposal on a Cooperative Operation Support Environment over Multiple Networks", Workshop on MIS, NA and OFS, Sapor in Japan, Sep., 1995.
11. Juan Pavon, "Connection Management Architecture", TINA-C Document, March, 1995.
12. Pier Giorhio Bosco and Corrado Moiso, "A Distributed Processing Model for Telecommunications Service Management", The Proceedings of DSOM 94, Oct., 1994.
13. Hyeon Ju Oh, Young Seok Shin and Hyeong Ju Kim, "The Sharing Scheme of Managed Object between Two Different Network Management Domains", The proceedings of ISNOM '96, April, 1996.
14. 신영석, 오현주, 고병도, 김재근, "차세대 개방형 정보통신망 구조인 TINA 연구(1), (2)", 한국전자통신연구원, 주간기술동향 744/5호, 1996. 5.
15. Y.S. Shin, B.T. Kim, H.J. Oh and M.K. Choi, "Desktop Video Conference System based on Open Networking Architecture", The proceedings of PRFASIG workshop, Nov., 1996.



신 영 석(Young Seok Shin) 정회원

1958년 10월 25일생

1982년 2월: 전북대학교 전자공학과 졸업(공학사)

1984년 2월: 전북대학교 대학원 전자공학과 졸업(공학석사)

1992년 2월: 전북대학교 대학원 전자공학과 졸업(공학박사)

1993년 3월~1994년 4월: 일본 NTT 통신망연구소 객원 연구원

1984년 3월~현재: 한국전자통신연구원 선임연구원

※관심분야: ATM-LAN, 분산시스템, TINA



오 현 주(Hyeon Ju Oh) 정회원

1968년 12월 4일생

1991년 2월: 충남대학교 전자계산학과 졸업(이학사)

1996년 3월~현재: 충남대학교 컴퓨터과학과 석사과정 중

1991년 1월~현재: 한국전자통신연구원 선임연구원

※관심분야: 분산처리시스템, 이동통신, TINA