

통합 통신망 관리를 위한 TINA manager와 TMN agent의 연동방안 연구

정회원 김 호 철*, 김 영 탁**

Integrated Network Management with TINA manager and TMN agent

Ho-cheal Kim*, Young-tak Kim** *Regular Members*

요 약

차세대 정보통신망의 중요한 요건으로 이동성 멀티미디어 서비스의 제공이 지적되고 있으며, 이러한 통신망 자원의 효율적인 운영, 관리를 위해서는 통신망 관리시스템이 필수적이라 할 수 있다. 현재 ATM과 SDH 기반의 초고속 통신망과 IP 기반의 인터넷네트워킹 기술이 차세대 정보통신망의 주축이 될 것으로 예상되며, 정보 통신망 요소 시스템들(교환기 및 전송장비, 단말장비, 라우터 등)의 관리는 TMN(Telecommunications Management Network)과 SNMP(Simple Network Management Protocol)을 기반으로 하는 agent들이 적용될 것이다. 이와 같이 서로 다른 기술을 기반으로 하는 통신망 관리 agent를 갖는 통신망 자원들을 통합관리하고 광역 정보통신망의 효율적인 구성을 위해서는 분산환경을 기반으로 한 통신망 관리체계가 보다 효과적일 수 있다. 이러한 요건을 만족하는 통신망 구조가 TINA(Telecommunications Information Networking Architecture) 체계의 통신망 관리구조이다. 본 논문에서는 통합 통신망 관리를 위한 TINA manager와 TMN agent 사이의 연동(interworking)방안 및 기능 구조를 제안한다. 제안된 연동구조는 관리객체를 변환하는 기존의 방식과는 달리 TINA 체계의 통신망관리 계층 중 EML(Element Management Layer)의 서브네트워크 관리동작을 통신망 요소 시스템의 TMN agent 관리 동작으로 변환하는 것을 기본 개념으로 하고 있다. 이는 specification translation을 수행하는 JIDM(Joint Inter-Domain management) 구조에 비하여 비교적 간단한 구조를 가지며 시스템의 처리부담이 작은 장점을 갖는다.

ABSTRACT

In the forthcoming next generation highspeed networks, the provisioning of broadband mobile multimedia services is the most important issue; while an efficient network management architecture, which can manage the network resources efficiently, is essential. The next generation highspeed networks will be composed of the ATM and SDH-based transport network systems and the IP-based interworking systems. The management functions of these network systems are implemented various technologies, such as TMN and SNMP. In order to integrate these network resources efficiently with heterogeneous management functions, the distributed network management architecture such as TINA is most appropriate. In this pape, we propose an interworking architecture for TINA manager and TMN agent. where the TINA-based EML subnetwork management operations mapped into the related management operations of the TMN NE agent. The proposed interworking architecture is simpler than JIDM, and the processing overhead is minimized.

* 영남대학교 대학원 멀티미디어통신공학과 광대역정보통신망 연구실(gum6man@infocom.ice.yeungnam.ac.kr),

** 영남대학교 정보통신공학과 광대역정보통신망 연구실(ytkim@ynuucc.yeungnam.ac.kr)

논문번호 : 99017-0809, 접수일자 : 1999년 8월 9일

I. 서론

현재의 통신망 요소시스템들은 다양화되고 복잡해지고 있으며 서비스 제공을 위해 상호 연동되고 있는 추세이다. 또한, 차세대 정보통신망의 이동성 및 멀티미디어 정보 요구에 의해 초고속 정보통신 요소시스템 기술이 계속해서 개발 되고 있다. 이와 같이 복잡하고 다양해지는 통신망의 효율을 극대화하기 위해서는 통신망의 관리기술이 필수적이라 할 수 있다. 현재 대표적인 통신망 관리 기술로는 TMN과 SNMP가 있다. 차세대 정보통신망의 요소 시스템들은 ATM (Asynchronous Transfer Mode)과 SDH (Synchronous Digital Hierarchy)를 기반으로 하고 통신망 기술은 IP 체계의 인터넷워킹이 주축이 될 것으로 예상된다. 또한, 이들 요소시스템들의 관리기술은 대부분 TMN과 SNMP를 기반으로 하는 agent로 구현이 될 것이다. 이는 다시 말해 통신망의 관리를 위한 시스템이 이원화됨을 의미한다.

서로 다른 기종의 다양한 통신망 요소 시스템들이 차세대 정보통신망 서비스의 제공을 위하여 상호 연동되고 있는 시점에서 효율적인 통신망 관리를 위해서는 이원화된 agent들을 통신망 관점에서의 통합하는 것이 필요하다. 즉, TMN 또는 SNMP 기반의 agent를 갖는 통신망 요소 시스템들을 통합관리할 수 있는 구조의 manager가 필요하다. 이러한 manager는 이동성 및 멀티미디어 서비스 제공에 적합한 구조이어야 하는데 이에 가장 적합한 통신망 관점의 관리구조 중 하나가 TINA 이다.

TINA는 소프트웨어의 재사용, 분산처리 등과 같은 개념을 기반으로 통신 소프트웨어를 쉽게 개발하고 새로운 통신망 서비스의 개발에 필요한 시간과 비용을 줄일 수 있는 장점을 제공한다^[1]. TINA는 기존 전송망 상에 DPE(Distributed Processing Environment)를 갖는 구조로 설계되어 있으며, 모든 소프트웨어는 이 DPE를 기반으로 구현된다. DPE를 기반으로 하지 않는 TMN과 SNMP agent들을 TINA manager를 이용하여 통합하기 위해서는 직접 또는 간접적인 연동(interworking) 인터페이스를 필요로 한다.

본 논문에서는 TMN과 SNMP를 기반으로 하는 관리 agent들을 TINA manager로 통합하기 위한 방안과 구조를 제안한다. 제안된 구조는 연동구조에 의한 오버헤드, 소프트웨어 구조, 프로토콜간 변환

방식을 고려하여 설계하였으며 이를 CORBA (Common Object Request Broker Architecture) 기반의 manager와 TMN 또는 SNMP agent와의 연동 구조인 JIDM(Joint Interdomain Management)과 비교한다.

논문의 구성은 II장에서 TINA와 CORBA의 구조적인 차이점과 이들을 기반으로 한 통신망 관점의 manager 구현에 따르는 TMN agent 연동 고려사항을 비교한다. III장에서는 TINA 체계의 manager 구성에 대한 시나리오를 정의하고 IV장에서 본 논문에서 제안하는 연동구조에 대한 설명을 한 후 V장에서 간단한 구현 예를 보이고 결론을 맺는다.

II. TINA 와 CORBA manager

TINA의 DPE는 CORBA ORB(Object Request Broker)를 코어로 하여 구성된다. 따라서, ORB에 적용되는 대부분의 개념이 TINA DPE에 포함되고 할 수 있다. 그러나, TINA의 DPE와 CORBA의 ORB 사이에는 설계목적에서부터 많은 차이점들이 존재한다. 이러한 차이점들은 CORBA에 적용되는 JIDM 연동구조가 TINA에 그대로 적용될 수 없음을 의미한다.

1. TINA DPE와 CORBA ORB

TINA DPE와 CORBA ORB의 첫 번째 차이점은 사용목적이다. 우선 TINA DPE는 서로 다른 관리기능을 통신망 관점에서의 통합을 우선하므로 이 기종의 관리기능들을 하나의 통신망 관점의 관리체제로 통합하고 이를 이용한 서비스의 제공 및 새로운 서비스의 개발을 쉽게 하는 것이 목적이다. 이는 non-DPE 소프트웨어도 TINA의 구조에 포함시키고자 하는 개념에서 명확히 알 수 있다. 반면에 CORBA ORB는 프로그래머가 통신망기술에 관여하지 않고 통신 소프트웨어를 구현할 수 있도록 하기 위한 범용의 구조 제공이 목적이다. 이는 모든 응용간의 통신을 ORB가 대신함으로써 가능하다.

두 번째 차이점은 분산 구조의 차이점이다. 앞서 언급한 것과 같이 CORBA 기반의 소프트웨어는 ORB를 통해서만 가능하다. 이는 CORBA ORB가 transport network을 숨기는 역할을 함으로써 나타나는 특성인데 이는 통신망의 특성을 고려한 응용 소프트웨어의 구현이 힘들다는 것을 의미한다. 이와는 달리 TINA DPE는 KTN(Kernel Transport

Network)과 분산환경을 포함하는 새로운 transport network를 구성하고 있으며 non-DPE node를 TINA 외 전체구조에서 고려하고 있으므로 특정 통신망의 특성을 고려한 응용 소프트웨어의 구성이 가능하다.

세 번째 차이점은 소프트웨어의 요소의 설계방식이다. CORBA ORB를 이용한 소프트웨어 요소의 설계는 IDL(Interface Definition Language)을 이용하여 되어있다. IDL을 이용한 설계는 소프트웨어 요소의 서비스 인터페이스만을 정의할 수 있으므로 통신망 관리에 필요한 객체의 설계에는 부족한 면이 있다. TINA DPE에서의 소프트웨어 요소 설계는 CORBA와는 다르게 ODL(Object Definition Language)을 이용한다. ODL을 이용할 경우 통신망 관리객체의 속성, 기능 등의 표현이 가능하다.

2. JIDM 연동방식의 문제점

앞서 언급한 것과 같이 CORBA 기반의 manager는 통신망 관리기능의 통합 개념을 갖고있지 않다. 즉, CORBA를 이용한 통신망 관점(network view)의 관리기능 구현은 CORBA ORB를 사용하는 TMN manager 또는 SNMP manager를 구현함을 의미한다. CORBA 기반으로 구현된 TMN manager와 TMN agent를 연동하기 위해서는 OMG(Object Management Group)에서 채택한 JIDM(Joint Inter-Domain Management) 방식을 사용한다. JIDM 방식은 CORBA ORB를 코어로 하는 TINA DPE 기반의 manager에도 적용이 가능하나 몇 가지 측면에서 TINA 체계의 통신망 관리 manager와 TMN agent 간의 연동에 직접적으로 적용되기 힘든 문제점이 있다.

첫 번째로 JIDM 방식의 연동을 위해서는 두 가지의 변환(specification translation, Interaction translation)을 수행하여야 한다. 특히 specification translation은 TMN 기반의 NE(Network Element) agent에 존재하는 관리객체를 일대일 변환을 통하여 분산객체화 하여야함을 의미한다. 따라서, NE 계층의 관리객체가 이원화되고 이를 관리하기 위한 구조가 복잡해진다.

두 번째로 JIDM 연동방식의 경우 통신망 관리기능의 통합이 복잡하다. CORBA 기반의 manager는 통신망의 통합보다는 통신망 관리기능의 다른 구현 방식이라고 볼 수 있다. 따라서, TMN domain과 SNMP domain의 통합을 위해서는 IIMC (ISO/CCITT and Internet Management Coexistence)와 같은 또 다른 연동방식을 필요로 한다.

세 번째로 JIDM 연동방식은 CORBA ORB가 하위계층의 통신망 프로토콜과 전송망을 사용자에게 숨기기 위한 specification translation을 수행함으로써 불필요한 traffic을 발생시킨다. 이원화된 객체는 일관성 유지를 위하여 모든 상태변화에 대한 사항을 실시간으로 처리해야한다. JIDM 연동방식의 경우 이러한 일관성 유지를 위한 처리도 ORB와 게이트웨이를 통해서 수행되기 때문에 통신망 관점에서는 불필요한 traffic 오버헤드가 발생하는 것이다.

앞서 언급한 JIDM 방식의 문제점들은 CORBA 기반의 manager 구현을 목적으로 한다면 반드시 수반되는 사항이므로 JIDM 연동방식 자체의 문제점이라고는 할 수 없다. 하지만 TINA 체계의 통신망 관리기능 구현을 목적으로 한다면 JIDM 방식의 다음의 방식을 통하여 해결할 수 있기 때문에 JIDM의 직접적인 적용보다는 이를 해결하기 위한 새로운 연동방식이 필요하다.

- ① JIDM 방식의 specification translation을 제거
- ② TINA의 통합구조를 수용하는 연동 구조

본 논문에서는 JIDM 연동방식을 TINA 체계의 통신망 관리구조에 그대로 적용할 경우 발생하는 문제점을 해결하기 위한 새로운 연동방식인 abstract translation 방식을 제안한다.

III. 통신망 관리기능의 연동 시나리오

1. TINA의 관리계층 구조

TINA 체계의 통신망 관리구조는 TMN의 5계층 구조를 일반화 3계층(service, resource, element) 구조로 정의하고 있으며, 통신망 관리기능 영역은 TMN의 5가지 관리기능(장애관리, 구성관리, 성능관리, 보안관리, 과금관리)을 수용한다. 이들 관리기능 중 구성관리는 다시 NTCM (Network Topology Configuration Management)과 연결관리로 분리된다^[2]. 앞서 언급한 TINA DPE를 포함하는 통신망 관리구조 개념을 도식화하면 그림 1과 같다.

그림 1에서 TINA 체계의 관리기능은 통신망관리와 컴퓨팅관리의 두 가지 구조가 있다. TINA 체계의 통신망관리 대상인 transport network은 KTN으로 표현되는 기존의 transport network와 DPE를 포함하는 새로운 구조이다. TINA DPE는 앞서 서술하였듯이 CORBA ORB를 core로 하고 있으나 동일하지 않다. 즉, CORBA ORB 서비스를 일부 포

합하는 확장된 개념으로 TINA DPE가 설계되었으며 KTN을 전체 구조에서 숨기지 않는다.

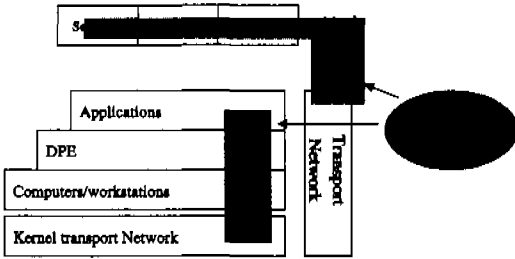


그림 1. TINA 관리구조

통신망 관리계층 중 element 계층은 통신망 관리 대상이 되는 topology resource들이 모여있는 곳이며 물리적인 자원뿐만 아니라 논리적인 자원도 관리된다. Element 계층을 TMN의 관리계층 관점에서 보면 NEL에 해당되는 계층이다. Resource 계층은 TMN의 관리계층 중 EML (Element Management Layer)과 NML (Network Management Layer)에 해당되는 계층으로 통신망 관리기능 소프트웨어 요소들과 관리정보 객체들이 모여있는 곳으로 TINA 체계의 manager를 구성한다. Service 계층은 TMN의 관리계층 중 SML(Service Management Layer)에 해당되는 소프트웨어 요소들로 구성되며 VPN (Virtual Private Network), Trouble Ticketing 등과 같은 통신망관리 서비스를 제공한다¹⁾.

2. 연동 유형

본 논문에서 제안하는 연동구조를 설명하기에 앞서 TINA 체계의 통신망관리 계층구조와 TMN 또는 SNMP 기반 관리시스템과의 연동유형을 보면 그림 2와 같이 크게 수평연동과 수직연동으로 분류할 수 있다. 그림 2에서 ①번, ②번, ③번, ④번은 수평연동 방식의 연동 위치를 보여준다. 수평연동 방식은 TINA 체계의 통신망 관리계층과 TMN 또는 SNMP 기반의 관리계층 중 동일계층을 연동한다. 이러한 수평연동 방식은 TMN 또는 SNMP 기반의 통신망관리 시스템이 이미 구축되어 있는 경우에 TINA 체계의 통신망관리 시스템과의 정보교환 목적 및 TINA 체계의 통신망관리 시스템으로의 변경을 위한 중간단계의 형태로 적절하다.

수직연동 방식은 동일계층간 연동이 아닌 상하 계층간을 연동 하는 방식으로 TINA 체계의 통신망 관리 시스템을 이용한 상이한 관리 시스템의 통합을 목적으로 한 사용에 유리하다. 그림2에서 ⑤번

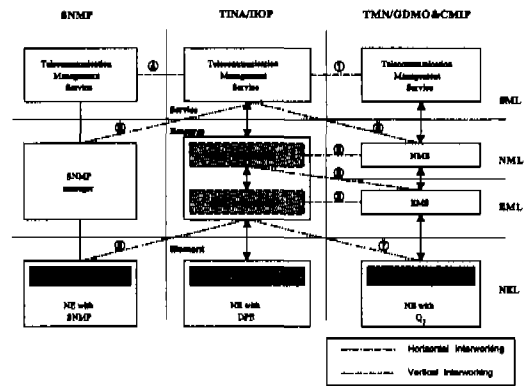


그림 2. TINA 체계의 관리시스템과 TMN, SNMP 기반 관리시스템과의 연동 유형

은 service 계층을 TINA의 분산 소프트웨어로 구성하고, 서비스 제공에 필요한 통신망 관리계층 중 NMS (Network Management Service)와 EMS (Element Management Service)를 TMN 체계의 통신망 관리기능을 이용하는 방식이다. ⑥번은 TMN 기반의 관리기능 영역인 EMS와 TINA 체계의 통신망 관리계층 중resource 계층에서 NML 기능 소프트웨어 요소를 연동 하는 방식이다. ⑦번은 TMN의 NEL을 제외한 모든 통신망 관리기능 영역을 TINA 체계의 통신망 관리시스템으로 구성하는 방식이다. ⑧번과 ⑨번은 SNMP 체계의 통신망기능과 TINA 체계의 관리기능을 연동 할 경우의 수직 연동 방식이다. ⑤번 형태의 수직 연동방식은 TMN 기반의 통신망관리 manager의 기능을 이용하여 TINA 체계의 통신망 서비스를 제공하는 형태로 TMN 기반의 manager가 구축되어 있을 경우 새로운 통신망 서비스를 효율적으로 개발 하고자 할 때 가장 현실적인 방법이다. ⑥번의 경우는 TMN 기반의 EMS 기능을 이용하는 연동방식으로 통신망 관점(network view)의 관리기능을 TINA 체계로 구성하고 NE 관리를 TMN 기반의 EMS에게 요청하는 형태이다. 하지만, 이 방식에는 몇 가지 문제점이 있다. 첫 번째 문제는 TINA 체계의 관리계층은 NML, EML로 구분되지 않고 resource 계층으로 구분되는 것으로 NML과 EML로 TINA 통신망 계층구분을 하는 것이 불필요하며 단지 소프트웨어 구성 요소의 기능에 따른 분류만 하게 된다. 따라서, ⑥번 방식의 경우 TINA 체계의 분산 소프트웨어 요소들과 TMN 기반의 EMS간 연동이 복잡한 구조적 양상을 갖게 된다. TINA 체계의 계층구조는 인위적인 분리 구현보다는 통합된 형태로의 resource

계층으로 구현하는 것이 효과적이다. 두 번째 문제는 관리기능의 수행에 있어 통신망 자원 관점으로 갈수록 단순한 정보의 흐름이 발생하고 통신망 관점으로 갈수록 복잡한 형태의 논리적인 정보가 발생함으로 NML과 EMS 사이에서 연동기능을 구현하는 것은 연동기능을 위한 TINA 소프트웨어의 구현에 어려움을 가증하게 된다. 그러므로 연동기능의 구현 시에 가장 효과적인 위치는 TINA 체계의 service 계층과 TMN 기반의 NMS 사이, 그리고 TINA 체계의 통신망관리 기능영역인 resource 계층과 TMN 기반의 NE 사이와 같이 계층구분이 일치되는 지점이다. 즉, TMN을 기반으로 하는 통신망 장비를 TINA체계의 관리 시스템으로 통합시키기 위해서는 ⑦번과 같이 TINA resource 계층의 최하단 EML 요소와 TMN 기반의 NE agent를 연동 하는 것이 가장 적합하다.

3. 연동 기본구조

TINA체계의 resource 계층 최하단 요소는 EML-CP, EML-TC, EML-FM 등과 같은 EML 소프트웨어 요소들이다^[3]. 이들 소프트웨어 요소들은 NML 소프트웨어 요소와는 유사한 관리기능을 수행 하지만 관리하는 자원의 범위가 다르다. TINA는 resource 계층의 소프트웨어 요소 구현 시 EML 요소와 NML 요소들 구분할 필요가 없지만 연동기능을 위해서는 EML 요소들 NML 요소의 기능과 구분하여 구현하는 것이 수직연동 방식에서는 유리하다. TINA 체계의 resource 계층과 element 계층 간 인터페이스는 그림 3과 같다.

TINA 체계의 통신망 관리구조에서는 resource 계층 요소들은 하위 계층의 통신망 기술이나 물리적인 장비에 독립적인 기능만을 갖도록 규정하고 있으나 element 계층이 TMN의 NE agent로 구성될 경우에는 그림 3의 구조에서 TMN 기반의 agent들 TINA 체계의 관리시스템에 유연하게 통합하기 위한 연동기능 구조가 필요하다. 앞에서 언급하였듯이 DPE와 ORB가 개념적으로 차이가 있고 manager 구현 형태와 목적이 다름으로 인해 JIDM 방식을 그대로 TINA 체계의 관리 시스템에 적용하게 되면 불필요한 오버헤드가 발생된다. 이러한 오버헤드를 줄일 수 있는 방법으로 JIDM의 specification translation을 수행하지 않는 것이다. 즉, NE agent의 관리객체를 일대일 변환하지 않고 NE agent의 기능을 그대로 이용하면서 관리정보를 제공하는 것이다.

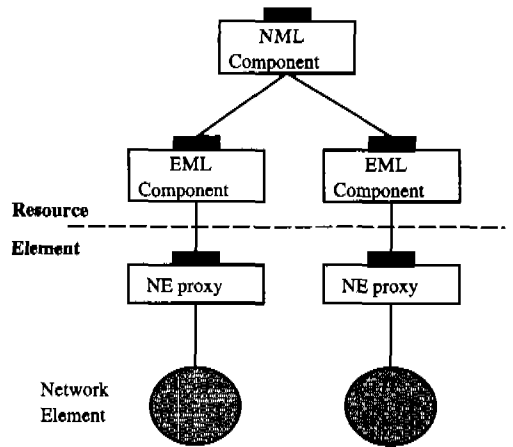


그림 3. TINA 관리구조의 resource 계층과 element 계층의 인터페이스

본 논문에서는 이러한 오버헤드를 제거하기 위해 abstract translation 방식을 제안한다. JIDM 방식은 객체와 객체간의 변환과정을 수행함으로써 관리객체의 이원화 문제를 발생시킨다. 하지만 abstract translation은 TINA EML 요소의 관리객체와 NE agent의 관리객체에서 동일 목적 또는 유사 목적의 객체를 분류하고 EML 요소의 관리기능 수행 시 계층간 관리기능을 변환함으로써 연동기능을 수행한다. 즉, TINA EML의 관리기능과 TMN NE agent의 관리기능간 변환이 abstract translation의 주안점이다.

Abstract translation 방식을 위한 연동기능형태는 우선 두 가지들 고려해 볼 수 있는데 그림 4의 (a)와 같이 TINA 체계의 resource 계층과 TMN기반의 NE agent 사이에 연동기능 수행을 위한 별도의 분산 소프트웨어 요소들 두는 구조와 (b)와 같이 TINA 체계의 EML 소프트웨어 요소에 TMN 기반의 NE agent 연동을 위한 기능을 포함시키는 구조이다. TINA 체계의 EML 소프트웨어 요소와 TMN 기반의 NE agent 사이에 별도의 소프트웨어 요소로 연동기능을 구현 할 경우 TMN기반의 NE agent는 연동기능 소프트웨어 요소에 의하여 가상의 TINA element(NE proxy)로 표현된다. 즉, 그림 3에서와 같이 TMN 기반의 NE agent가 TINA EML 소프트웨어 요소에 의해 NE proxy로 인식이 된다. 이 방식은 소프트웨어의 구조적인 측면에서는 TINA 체계의 분산 통신망 관리구조에 대한 규정을 그대로 따른다는 장점이 있다. 그러나, 통신망 관리 측면에서는 두 가지 오버헤드를 갖고 있다.

첫 번째 오버헤드는 연동기능 소프트웨어 요소가 모든 통신망 관리기능 영역에서 모든 EML 요소의 관리기능 요구에 대하여 어떠한 형태로든지 관리기능 변환을 제공할 수 있어야 하는 것이다. 이는 연동기능 소프트웨어 요소의 구조를 복잡하게 하는 주요인이 된다. JDM 방식이 이러한 구조를 갖는데 interaction translation 기능은 단순한 반면 NE agent의 모든 관리객체를 TINA의 분산객체로 일대일 변환함으로써^[4,5] 전체적인 연동구조가 복잡하게 된다.

두 번째 오버헤드는 non-DPE node인 TMN 기반의 NE agent를 DPE node인 TINA 체계의 EML 요소에 연동하기 위해 2 단계의 메시지 전송구간이 존재하고 이것에 의한 전송 오버헤드가 발생한다. 1 단계 메시지 전송은 TINA 체계의 EML 요소와 연동기능 소프트웨어 요소 사이에 발생하는 전송이고, 2단계는 연동기능 소프트웨어 요소와 NE agent 사이에서 발생하는 CMIP(Common Management Information Protocol) 전송이다. 2 단계의 메시지 전송에 사용되는 프로토콜은 모두 응용계층에서 수행되는 프로토콜이므로 메시지 서비스를 위해 추가되는 데이터 오버헤드 외에 CPU 연산 오버헤드가 상당한 비중을 차지하게 된다.

TINA 체계의 EML 요소에 TMN 기반의 NE agent 연동을 위한 직접적인 인터페이스를 포함시키는 구조는 EML 요소와 연동기능 간에 분산 메시지 전송구간을 제거함으로써 전자의 방식에 비하여 상대적으로 전송 데이터 오버헤드를 줄일 수 있고 연동기능이 해당 EML 요소의 관리기능만을 지원하면 됨으로 비교적 간단해진다. 하지만 TMN 기반의 NE agent가 TINA 체계의 분산 소프트웨어 요소 형태가 아닌 TMN 기반의 NE agent를 직접 EML 요소에 접속시키는 형태가 된다. 전자의 방식은 EML 요소가 하부의 통신망 관리 프로토콜을 고려하지 않아도 되는 EML 요소로 구성할 수 있는 반면 직접적인 인터페이스를 포함시키는 구조는 하부 통신망 관리 프로토콜에 의존적인 EML 요소로 구성된다. 이는 TINA 체계의 resource 계층 소프트웨어 요소는 하부 통신망 기술에 독립적인 기능으로 구성된다는 규정에 어긋나는 구조이다. 직접적인 인터페이스를 EML 요소에 포함시키는 방식의 연동구조는 별도의 연동기능 소프트웨어 요소 구조에 비하여 메시지 전송구간을 1단계로 줄임으로 인해 오버헤드를 줄일 수 있다는 장점이 있지만 구조적인 문제점을 내포하고 있으며, 모든 EML 요소들이

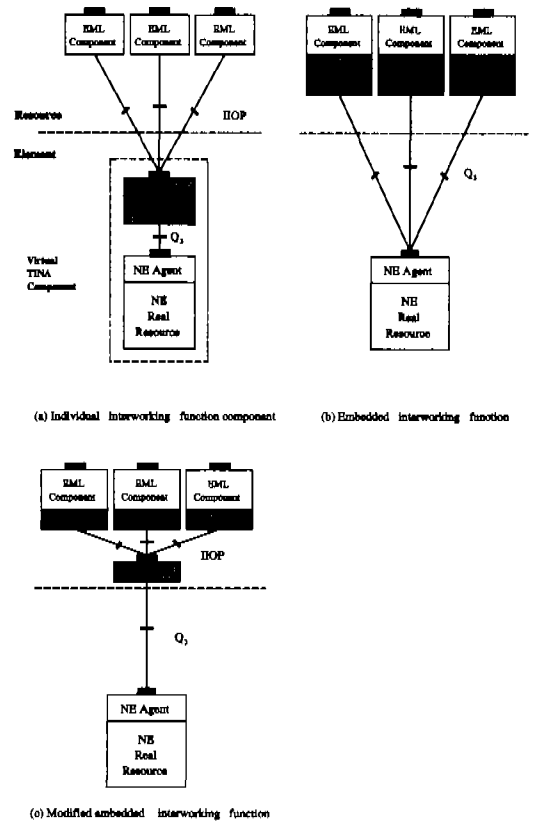


그림 4. 세 가지 연동기능의 형태

CMIP 스택을 내부에 갖고 있어야 하기 때문에 CPU 연산 오버헤드는 CMIP 스택의 비중을 생각했을 때 오히려 증가하는 문제점이 있다. 또한, TMN 기반의 NE agent의 입장에서 다중의 CMIP 연결 설정이 이루어져야 함으로 이를 지원하지 않는 agent의 경우 적용이 힘들다.

따라서, 본 논문에서는 독립된 연동기능 소프트웨어 요소 구조의 장점과 직접적인 인터페이스를 EML 요소가 갖는 구조의 장점을 모두 갖는 구조를 그림 4의 (c)와 같이 제안한다. 우선 TINA 체계의 모든 EML 요소가 CMIP 스택을 갖게되는 문제를 해결하기 위해 CMIP 스택을 EML 소프트웨어 요소에서 분리하여 별도의 소프트웨어 요소로 구성한다. 이러한 구조는 첫 번째 방식의 구조를 갖게 되어 NE agent가 분산 소프트웨어 요소로서의 장점을 갖는다. 그러나 첫 번째 방식의 구조와는 달리 관리기능간 변환이 EML 요소에서 발생한다는 것이다. 즉, 첫 번째 방식의 구조는 관리기능 변환이 연동기능 소프트웨어 요소에서 수행되는 반면 제안된 방식에서는 변환기능이 EML 요소의 기능으로

포함되어 있다. 이렇게 함으로써 변환기능이 복잡해지는 것을 피할 수 있다. 하지만 이러한 분리는 두 번째 방식의 전송 오버헤드를 줄이는 측면은 분리된 CMIP 스택 소프트웨어 요소에 의하여 상실된다.

본 논문에서 제안한 그림 4의 (c) 연동방식은 앞서 언급한 방식들 중에서 구조적, 기능적 장점을 가지는 반면 2단계 메시지 전송을 수행하기 때문에 전송 오버헤드의 감소는 기대할 수 없다. 하지만, 본산 구조인 TINA 체계의 통신망 관리시스템에서 연동기능의 2단계 메시지 전송 제거에 의한 오버헤드 감소는 성능향상에 미치는 영향이 상대적으로 미약하다.

IV. TINA manager와 CMIP agent의 연동을 위한 세부 기능구조

본 논문에서 제안한 연동구조의 기본동작은 그림 5와 같다. EML 요소의 외부로부터 EML 요소 쪽으로 인터페이스를 통해 관리기능 요청이 도착하게 되면 자신이 관리하는 관리객체에 대한 관리기능을 수행하게 된다. 만약 관리기능 수행이 하위계층의 NE agent 관리기능 수행을 필요로 하는 경우 EML의 mapping 부 계층에서 계층간 기능변환을 수행한 후 CMIP 소프트웨어 요소의 CMISE 인터페이스를 호출하여 NE agent에 관리기능 수행을 요구한다. CMIP 소프트웨어 요소는 EML로부터 전달된 메시지를 NE agent로 메시지를 변환하여 전달하게 된다.

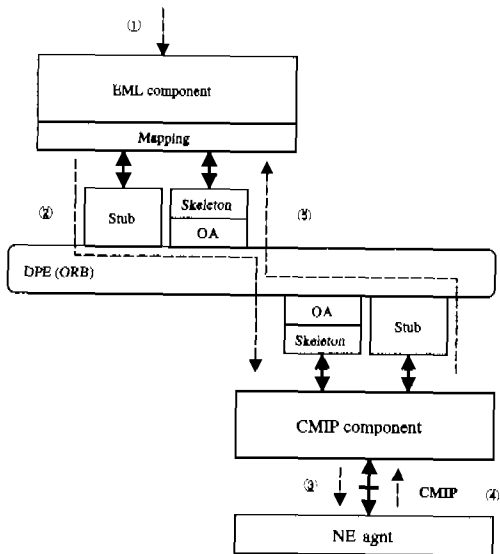


그림 5. 연동 구조의 기본동작

1. EML 소프트웨어 요소 구조

변환기능을 갖는 EML 요소는 크게 2개의 영역으로 구분하여 구성된다. 우선 TINA resource 계층의 NML 요소의 요청을 수신하기 위한 EML 인터페이스 영역과, TINA의 EML 관리기능을 TMN 기반의 NE agent 관리기능으로 변환하기 위한 메시지 변환 영역(abstract translation)으로 구성된다. 이 중 TINA에서 정의하고 있는 부분은 NML 요소들의 요청을 수신하기 위한 EML 인터페이스와 이 인터페이스가 제공하는 기능이다. 각 영역은 EML 요소들 부 계층 형태로 재구성한다. EML 요소의 부 계층 구조를 그림 6에 나타내었다.

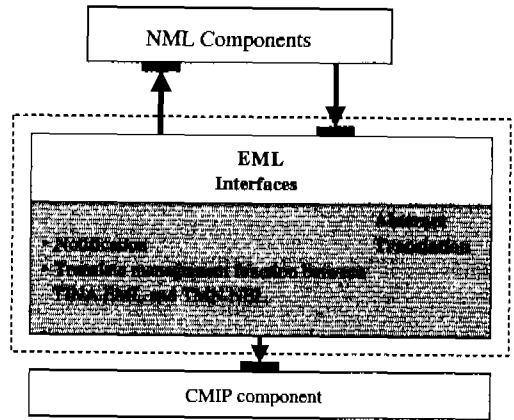
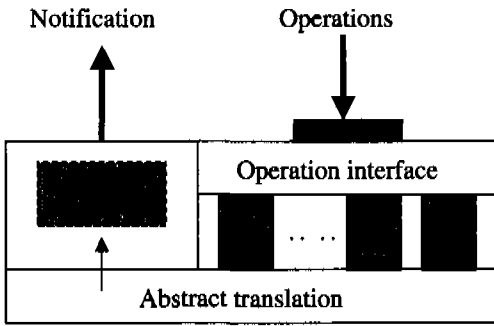


그림 6 변환 기능을 갖는 EML 소프트웨어 요소의 부 계층 구조

각 부 계층의 기능을 살펴보면 인터페이스 기능 영역은 TINA 체계의 실질적인 EML 기능을 수행한다. 이 계층은 TINA에 의해 현재 구조가 정의되고 있으나 인터페이스 기능정의 만이 정의되고 있을 뿐 내부구조는 언급하지 않고 있다. 그러므로 입력과 출력부분 TINA의 정의를 따르도록 설계함으로써 NML 요소와의 메시지 전송에 영향을 주지 않도록 하는 것이 가능하다. 이를 위해 인터페이스 기능영역을 그림 7과 같은 구조로 구성한다. EML 요소의 실질적인 관리기능을 인터페이스 기능과 분리하여 별도의 기능블록(FB)으로 설계함으로써 인터페이스에 대한 독립성을 보장하고, 다양한 변환기능 및 프로토콜 스택에 쉽게 적용하도록 하였다. FB가 수행하는 기능은 NML요소의 요청을 실질적으로 수행하는 것이다. NR 블록은 역 방향 메시지 전송을 담당하는 상위 계층으로의 참조자이다.



FB : Functional Block
NR : Notification Reference

그림 7. EML 소프트웨어 요소의 TINA 인터페이스 기능 구조

TINA 체계의 EML 요소가 수행하는 관리기능은 TMN의 관리기능 수행과는 절차적인 부분에서 다르다. 서로 다른 수행절차를 갖는 관리기능 간의 변환은 EML 요소의 기능블록의 요청에 의해 메시지 변환 부 계층에서 판단하여 수행한다. 이러한 변환의 판단을 위해 TINA 체계의 EML 요소가 수행하는 관리기능과 TMN 기반의 NE agent에 적용되는 관리기능에 대한 사전 변환작업이 수행되어야 한다. 그리고 각 관리기능간의 변환과 함께 관리객체의 상호 연관성도 사전에 파악되어야 한다. 이는 관리기능의 목적이 동일하더라도 관리하는 정보객체의 종류와 형태가 통신망 관리체계 간에 서로 다를 수 있기 때문이다. 이러한 사전 변환과정은 서로 다른 통신망 관리체계 간의 연동에 있어 가장 핵심이 되는 사항이다. 사전 변환된 내용은 연동구조 안에서 function mapper로 구현된다. 메시지 변환 부 계층은 변환과정의 효율화를 위하여 그림 8과 같은 구조로 구성되며 수행하는 기능은 다음과 같다.

- EML 요소의 관리객체를 조작하고 하위계층의 NE agent와 관리기능 변환에 의한 메시지 전송 필요성 결정 (Resource Manager)
- EML의 통신망 관점에서의 관리기능을 NEL의 망 자원 관점의 관리기능으로 변환(Function Mapper)
- EML 인터페이스 부 계층의 notification 참조자를 이용하여 notification 메시지를 전송 (Notification)
- CMIP 소프트웨어 요소와 연결 유지 (Association)

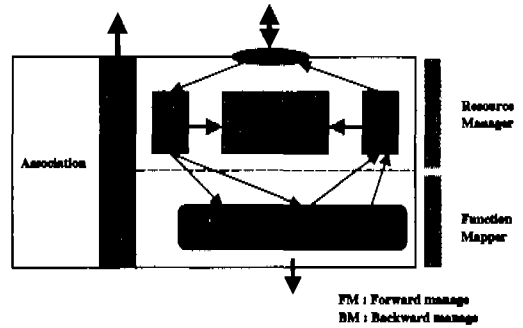
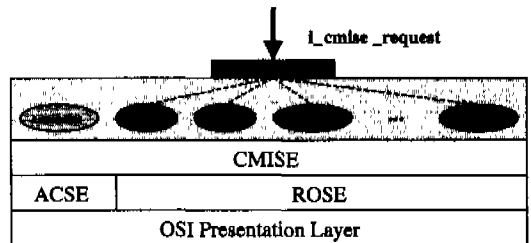


그림 8. 메시지 변환 부 계층 구조

Function mapper는 TINA 체계의 EML 관리기능과 TMN 기반의 NE agent간 기능 정합을 목적으로 하는데 이는 하나의 EML 관리기능 수행이 다수의 NE agent 기능 수행으로 표현될 수 있고 그 반대일 수도 있기 때문이다.

2. CMIP 소프트웨어 요소 구조

CMIP 소프트웨어 요소는 TMN NE agent와의 Q3 인터페이스를 제공한다. 이 소프트웨어 요소는 EML 요소의 CMISE (Common Management Information Service Element) 요청을 위한 인터페이스와 Q3 연결 설정을 위한 ACSE와 CMIP 메시지 전송을 위한 ROSE로 구성되어 있다. TMN의 관점에서 보면 CMIP 소프트웨어 요소는 NE agent에 대한 NE manager로 동작하게 된다. 즉, NE agent가 서버로 CMIP 소프트웨어 요소가 클라이언트가 된다. 그림 9는 CMIP 소프트웨어 요소의 구조이다.



ACSE : Association Control Service Element
ROSE : Remote Operation Service Element

그림 9. CMIP 소프트웨어 요소 구조

가장 상위에 위치한 부 계층이 CMISE 인터페이스 계층으로 EML 소프트웨어 요소의 메시지 변환 요청을 수신하는 TINA 체계의 인터페이스이다.

i_cmise_request 인터페이스에는 CMIP 메시지에 해당하는 멤버 함수들이 구현되어 있어서 요청의 종류에 따라 해당 메시지 변환과 함께 CMIP 스택을 통하여 메시지를 NE agent로 전송한다. EML 요소의 function mapper에 의하여 CMIP 소프트웨어 요소의 프로토콜 변환함수가 호출되면 프로토콜 변환함수는 해당하는 CMISE API (Application Programming Interface)를 사용하여 CMIP 메시지를 전송한다. 예를 들면 EML-CP는 연결설정을 위해 몇 가지 절차를 기능블록과 FM(Forward manager)에서 수행한 후 NE agent로 connect라는 M-Action 메시지를 NE agent로 전송해야 한다. 이때, function mapper가 CMIP 소프트웨어 요소의 인터페이스에서 M-Action method를 IOP(Internet Inter-ORB Protocol)을 통해 호출하면 M-Action method는 CMISE의 m-action API를 이용하여 CMIP 메시지를 전송한다.

3. JIDM 방식과의 차이점

본 논문에서 제안한 방식은 연동이라는 측면에서 JIDM과 유사한 점이 있다. 이는 TINA의 DPE가 CORBA의 ORB를 코어로 하고있는데 기인한 것이다. 하지만 앞서 언급하였듯이 JIDM 방식을 TINA 체계의 통신망 관리에 적용하기 위해서는 또 다른 연동구조를 필요로 한다. 또한 이 연동구조 역시 분산요소로서 구현되어야하는 문제점이 있다. 이렇게 될 경우 이들 소프트웨어 요소에 의한 처리 지연이 발생하게 되고 전체적인 성능에 영향을 줄 수 있다. 특히, TINA의 EML 요소는 자원하나에 하나씩 존재하므로 이러한 처리 지연의 증가는 고속의 통신망이 될수록 심각한 고려대상이 될 것이다. 따라서 본 논문에서 제안한 연동 구조는 이러한 2차적인 연동에 의해 발생하는 오버헤드를 제거하기 위하여 JIDM 방식에서 specification translation을 abstract translation으로 대체하였다.

본 논문에서 제안한 연동 방식과 JIDM과의 가장 큰 차이점은 연동의 관점이 어디에 있는가 하는 것이다. 이는 다음과 같이 설명할 수 있다. JIDM 방식의 specification translation은 통신망 관점의 관리 소프트웨어를 고려하지 않고 단지 NE agent의 객체를 CORBA의 ORB상에 어떻게 표현할 것인가에 관한 방안이다. 즉, TINA 체계의 독립적인 통신망 관리시스템은 자신의 관리방식이 아닌 하부 agent의 관리방식을 따라야 한다는 것이다. 반면에 본 논문에서 제안한 방식은 하부 agent의 객체를

변환하지 않으므로 TINA 체계의 통신망 관리시스템은 자신의 관리방식으로 관리기능을 수행할 수 있다. 이는 연동이 관리객체를 기준으로 하여 수행하는 것이 아니라 관리기능간의 상호 변환을 통해서 수행됨으로 인해 가능하다. 다시 말해 본 논문에서 제안한 방식은 TINA 체계의 통신망 관리시스템 관점에서 하부의 agent를 어떻게 수용할 것인가에 대한 해결방안을 제시한 것이다

본 논문에서 제안한 abstract translation은 JIDM 방식과는 달리 TINA 체계의 통신망관리 시스템 구성 시 또 다른 연동을 요구하지 않는다는 측면에서 보면 JIDM 방식을 TINA 체계에 적용할 때 보다 유리하다. 즉, 연동을 위한 처리 지연이 감소함으로써 차세대 정보통신망의 요구사항을 만족할 것이다.

V. 제안 연동구조를 적용한 설계

1. TMN 기반 통신망 자원관리를 위한 TINA 관리 시스템

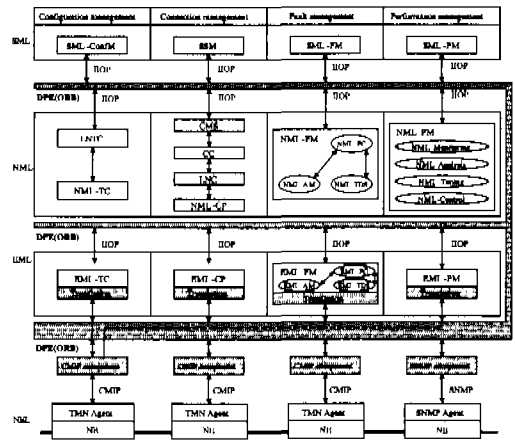


그림 10 제안된 연동구조를 갖는 TINA 통신망 관리시스템

TINA의 통신망 관리시스템을 관리기능별로 구분하면 그림 10과 같다. 그림 10은 TINA의 6가지(연결관리 포함) 관리기능 영역 중 통신망의 성과와 직접적인 관련이 있는 4가지 관리기능만을 표기하였다. TINA 체계의 통신망 관리시스템은 계층별로 구분되는 소프트웨어 요소들이 ORB를 코어로 하는 TINA DPE 상에서 분산 구현된다. 예를 들면 NTCM을 위한 관리 요소들은 계층별로 EML-TC, NML-TC, LNTC 등이다. 각 관리기능의 EML 요소에 abstract translation 기능이 내장되고 TMN

agent와 CMIP component를 이용하여 메시지 전송을 수행한다. 그림 10에서 TINA 체계의 통신망 관리시스템의 목적인 통신망 관리기능 통합의 형태를 보여주는데 우측 하단에 있는 NE agent는 TMN 기반이 아닌 SNMP를 기반으로 하는 agent이다. SNMP의 연동은 TMN의 연동에서 abstract translation의 function mapper를 SNMP 관리기능으로 변경하고 이를 SNMP 요소를 이용하여 메시지를 전송함으로써 구현된다. 이때, EML 요소의 상위 부계층은 TMN의 경우와 동일하기 때문에 통신망 관리기능 통합을 위한 또 다른 연동방법이 필요 없음을 알 수 있다.

2. EML-CP 적용 예

본 절에서는 제안된 연동구조를 TINA 체계의 연결관리 기능 요소인 EML-CP에 적용하여 ATM 스위치의 연결기능을 수행하는 예를 보인다. EML-CP의 선택 이유는 TINA의 통신망 관리기능 중 현재 EML-CP가 가장 많은 부분에 대하여 정리가 되어 있으며 TMN 기반의 연결관리와는 달리 매우 중요한 기능으로 정의되어있기 때문이다. 그림 11은 EML-CP의 관리기능 수행을 위한 관리객체의 일부이다.

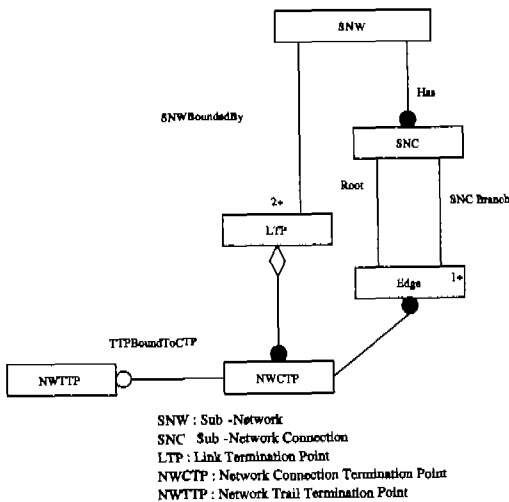


그림 11. EML-CP 관련 통신망 자원

TINA의 EML-CP는 SncServiceFactory와 Snc-Service의 2가지 인터페이스로 구성된다. SncServiceFactory는 EML-CP가 관리하는 SNW(sub-network)에 대한 접근정보를 의미하며 SNC(Sub-network Connection)를 생성하는 기능을 수행한다. SNC는

통신망 자원으로서 SncService인터페이스에 의하여 구체화되는 객체이다. SncService인터페이스로 구체화된 SNC의 역할은 sub-network 안에서의 논리적인 연결점인 Edge를 생성하고 이를 sub-network의 종단점인 NWCTP (Network Connection Termination Point)에 attach하고 detach 함으로써 실질적인 연결기능이 수행되게 하는 것이다⁶⁾.

연동기능의 구현을 위해 EML-CP 동작과 TMN 기반의 NE agent와의 동작을 비교해 보면 EML-CP의 연결동작은 먼저 NML-CP가 EML-CP의 SncServiceFactory 인터페이스를 통해 새로운 SNC의 생성을 요청을 보낸다. SncServiceFactory는 SncService instance를 생성함으로써 이 요청에 대한 서비스를 수행한다. SncService는 생성 시에 SNC 객체와 root edge를 생성하고 root edge와 지정된 NWCTP를 서로 attach 한다. 이때 NWCTP 객체가 생성되어 있지 않으면 NTCM을 통해 새로 생성하게 된다. SncService가 생성되고 나면 실질적인 연결설정을 위한 leaf edge의 생성 요청과 leaf edge에 지정된 NWCTP의 attach를 SncService를 통해 수행됨으로써 EML-CP의 연결동작을 완료하게 된다⁷⁾.

이와는 다르게 TINA 체계의 element 계층 소프트웨어 요소로서의 TMN agent 연결설정 동작은 agent의 상위계층에서 연결과 관련된 action 메시지가 agent의 객체로 전달됨으로써 수행된다. 본 논문에서는 ATM 스위치를 망 자원으로 사용하기 때문에 EML-CP는 TMN 에이전트의 atmFabric 관리객체로 M-ACTION connect 메시지를 전달함으로써 atmCrossConnection 관리객체를 생성한다. 생성된 atmCrossConnection 관리객체는 fromTermination과 toTermination을 위한 vcCTPBidirectional 관리객체를 생성하고 연결 함으로써 연결관리 동작을 완료한다⁸⁾.

위의 EML-CP와 TMN NE agent의 연결관리 기능수행 과정을 분석해 보면 TINA 체계의 element 계층 소프트웨어 요소로서의 TMN NE agent와 resources 계층의 EML-CP간에 연결설정을 위해서는 한번의 메시지 전송만이 필요함을 알 수 있다. 메시지 전송의 시점은 EML-CP의 leaf edge와 NWCTP의 attach 시에 발생한다. 연동기능에 의한 연결설정은 element 계층의 M-ACTION 메시지에 대한 완료 메시지를 EML-CP가 TMN NE agent로부터 수신함으로써 최종적인 연결설정이 완료된다. EML-CP의 통신망 자원 객체와 TMN NE agent의

GDMO 관리객체 간에 상호 연관관계가 있는 것들을 표 1에 정리하였고 연결설정 동작에 의한 두 계층간에 발생하는 메시지 전송에 대하여 표 2에 정리하였다. TMN NE agent의 GDMO는 ITU-T I.751 권고안을 따랐다.

표 1. EML-CP와 TMN agent 관리객체 매핑

TINA Quasi-GDMO		TMN GDMO	
Object Type	Key Attribute	MO	Key Attribute
Subnetwork	characteristicInfo	atmFabric	AtmFabricId
Subnetwork Connection	connectionTopology	atmCrossConnection	CrossConnectionId AdministrativeState OperationalState Directionality FromTermination ToTermination
Network Connection Termination Point	directionality	vcCTPBidirectional vpCTPBidirectional	VcCTPId VpCTPId SegmentEndPoint
Link Termination Point	directionality	atmCrossConnection	Directionality
Edge	directionality	atmCrossConnection	Directionality

표 2. EML-CP와 TMN agent 관리기능 매핑

EML-CP Operation	TMN Agent	
	MO Operation	관련 Attributes
SncServiceFactory.create_snc()		
SncService.create_edge()		
SncService.attach_edge()	atmFabric.connect()	FromTermination ToTermination
SncService.delete_edge()	atmFabric.disconnect()	ObjectInstance

EML-CP의 두 인터페이스에 의하여 구성되는 기능블록은 모두 9개로 구성되며 이 중 하나는 EML-CP에서 NML-CP로의 notification을 위한 기능 블록이다. 기능블록들은 인터페이스의 멤버함수 수와 일치한다. SncServiceFactory 인터페이스의 멤버함수로는 create_snc()와 create_setup_snc()의 두 가지가 있고, SncService 인터페이스의 멤버함수로는 delete_snc(), create_edge(), delete_edge(), attach_edge(), detach_edge() 그리고 migrate_edge()의 6가지가 있다. TMN의 통신망 자원에 대한 연동기능 구조를 갖는 EML-CP는 그림 12와 같다.

VI. 결론

본 논문에서는 기존의 CORBA 분산환경과 TMN의 연동에 관한 논문^{9,10)}들에서 언급된 구조와는 다르게 독립적인 통신망 관리구조를 갖는 TINA 체계

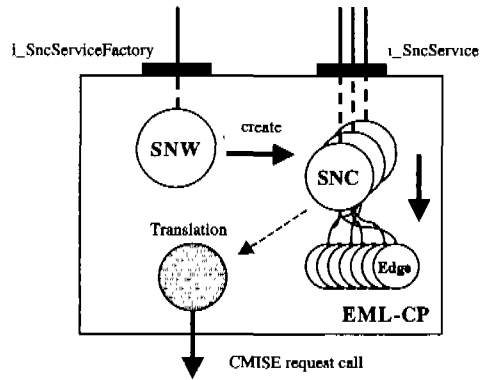


그림 12. TMN 연동기능 구조의 EML-CP

의 통신망 관리시스템에 TMN 기반의 관리기능을 연동하기 위해 통신망 관리기능 변환을 수행함으로써 단순화되고 TINA 분산 소프트웨어 요소로서의 장점을 갖는 구조를 제안하였다. 제안된 연동구조는 JIDM 방식에서 발생하는 관리객체의 변환으로 인한 관리객체 이원화 문제를 TINA 체계의 manager에 적용이 가능하도록 제거하였다. 이러한 연동구조의 단순화는 소프트웨어 요소가 실행되고 있는 하드웨어 시스템의 처리 부담을 줄임으로 인해 성능의 향상을 볼 수 있다. 이는 TINA의 DPE 및 메시지 전송 프로토콜인 IIOP이 응용계층의 프로토콜로써 다수의 EML 소프트웨어 요소가 동일 하드웨어 시스템 상에서 수행될 경우 전송 오버헤드와는 달리 CPU 연산 오버헤드는 전체 통신망 관리시스템에 심각한 영향을 줄 수 있기 때문이다.

본 논문의 향후 과제로는 TINA의 6가지 통신망 관리 영역을 본 논문에서 제안한 연동 구조를 기반으로 구성하고 이를 기반으로 한 통신망 서비스의 구현을 수행할 예정이다.

참고 문헌

- [1] TINA-C, "Overall Concepts and Principles of TINA," *TB_MDC.018_1.0_94*, Feb. 1995.
- [2] TINA-C, "Management Architecture," *TB_GN.010_2.0_94*, Dec. 1994.
- [3] TINA-C, "Network Resource Architecture version 3.0," *NRA_v3.0_97_02_10*, Feb. 1997.
- [4] OMG, "JIDM Interaction Translation," *telecom/98-08-14*, Aug. 1998.
- [5] X/Open, "Inter-domain Management: Specification Translation," *P509*, May. 1997.

[6] TINA-C, "Network Components Specification version 2.2," *NCS_DRAFT_v2.2_97_12_20*, Dec. 1997.

[7] TINA-C, "Network Resource Information Model Specification version 3.0," *NRIM_v3.0_97_12_17*, Dec. 1997.

[8] ITU-T, "Asynchronous Transfer Mode Management of The Network Element View," *Rec. I.751*, Mar. 1996.

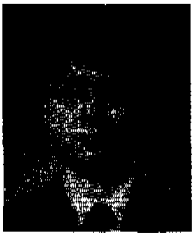
[9] Qinzhen Kong, Graham Chen, "Integrating CORBA and TMN Environments," *Proceedings NOMS 96*, 1996.

[10] Wang-Chien Lee, Gail Mitchell, "A Framework for TMN-CORBA Interoperability," *Proceedings NOMS98*, 1998.

B-ISDN, NGI, TINA, SNMP, TMN

김 호 철(Ho-Cheal Kim)

정회원



1989년 2월 : 경북대학교 전자공학과 졸업

1989년 1월~1995년 7월 : 삼성전관 종합연구소 주임연구원

1995년 8월~1996년 4월 : 삼성상용차 기술지원 대리

1999년 2월 : 영남대학교 멀티미디어통신공학과 졸업 석사

1999년 3월~현재 : 영남대학교 멀티미디어통신공학과 박사과정

<주관심 분야> TINA, SNMP, TMN, NGI

김 영 택(Young-Tak Kim)

정회원



1984년 2월 : 영남대학교 전자공학과 졸업

1986년 2월 : KAIST 전기 및 전자공학과 졸업 석사

1990년 2월 : KAIST 전기 및 전자공학과 졸업 박사

1990년 3월~1994년 8월 : 한국통신 통신망연구소 전송망구조연구실 선임연구원

1994년 9월~현재 : 영남대학교 공과대학 정보통신공학과 부교수

<주관심 분야> Broadband networking, ATM/