

〈主 題〉

통신관리망(Telecommunication Management Network : TMN)의 개발동향

이 중 회

(모다정보통신 기술연구소)

□ 차 례 □

- I. TMN 망관리 개요
- II. TMN 개발환경구축
- III. 개발사례
- IV. 개발경험
- V. 맺음말

I. TMN 망관리 개요

최근의 지구촌내의 통신분야는 급격한 변화를 맞고 있다. 규제완화의 지구촌화가 그중 대표적인 것인데 규제완화는 통신분야의 새로운 사업자를 많이 창출하게 하였고 지구촌화는 지구촌내의 종합통신망관리를 통한 하나의 통합된 가상망을 꿈꾸게 되었다. 새로운 통신 사업자들이 성공적으로 살아 남기 위해서는 효율적으로 통신망을 운용하고 관리하는 것이 매우 중요하게 되었다.

지난 여러해 동안 전송 기술과 교환 기술은 반도체 기술 및 광 전송 기술의 급속한 발전에 힘입어 수 Gbps 급으로 대용량화를 실현하였고, 매체 전송 기술 또한 협대역 케이블 전송으로부터 광대역 디지털 마이크로 웨이브 및 광 전송으로의 괄목할 만한 발전을 이룩하여 왔으며 무선통신 분야 또한 셀룰라 이동통신, 개인휴대통신, CT-2등 급속한 서비스의 발전이 이루어지고 있다. 최근에는 트래픽의 급속한 증가 및 비디오 서비스 등과 같은 광대역 서비스의 등장으로 주고받는 정보의 다양성과 유무선의 복합적인 수요가 가속화될 전망이다.

이와 같은 교환 및 전송 기술의 발전에 비해, 통신 품질 및 운용 관리비용 등에 막대한 영향을 미치는 통신망의 운용, 관리, 유지보수 및 설비(operations, administrations, maintenance and provisioning : OAM&P) 분야의 기술 개발은 종합적이고 체계적으

로 수행되지를 못하고, 여러 통신망 요소별로 검출된 OAM&P 정보들이 각각의 독립적인 데이터로서만 이용되는 운용 관리 방식이 대부분이었다. 따라서, 예방적 차원의 운용 관리라기보다는 사후 장애 관리가 주를 이루고, 오늘날 사용되고 있는 대부분의 통신망 집중 운용 관리 시스템들은 주로 기종별 및 제작회사에 고유한 기술 형태 내지는 운용 관리 기능을 지원하기 위한 목적으로 기술 개발이 이루어지고 있다. 이에 따라 새로운 시스템이 개발될 때마다 그 시스템에 맞는 OAM&P 정보는 종합적인 전송망 관리를 위하여 새로운 형태의 관리 데이터로 수정되어야만 했다.

이러한 망 관리의 비효율성 및 장치 의존성을 개선하기 위해서는 교환 및 전송시스템의 물리적 실체의 다양성에 의존하지 않는 논리적 관리 객체를 추출, 정보화 하여 기종 및 제작회사와 무관한 관리 정보 즉, 종합적 관리 정보와 망 요소 사이에 개방된 인터페이스 표준이 요구되고 있다. 또한, 광역화, 이기종 접속 및 다양한 망 장치에 대한 효율적인 관리의 수단으로 망관리 방법이 전세계적으로 활발히 연구되고 있다. 이에 International Organization for Standardization (ISO)에서는 이기종간의 상호 접속 및 관리에 대한 문제점에 대해서 활발히 연구하여 왔다. 그러한 결과로, ITU-T SG4에서는 통신망과 제공되는 서비스에 대한 관리기능 및 통신을 제공하기 위해 통신관리망 (Telecommunication Management Network: TMN)에

대한 표준화 작업을 추진하였다. TMN은 단일한 인터페이스를 사용하여 다양한 OS와 통신 장치들 간에 관리정보를 교환하기 위한 표준화된 구조를 제공한다.

TMN의 기본 목표는 통신망 관리를 위한 구조를 제공하는 것이다. TMN의 주요 개념을 살펴보면 다음과 같다. TMN은 장비제공자와는 상관없이 어떠한 망의 자원도 관리자의 적용필요에 따라 관리되어야 한다. 관리되는 망의 자원과 관리자의 적용필요는 표준화된 통신수단을 이용하고 이러한 자원을 표시하는 공통 정보 모델을 공유하여야 한다. 다수의 정보제공자의 상호운용성을 증대시키기 위하여 OSI 표준을 쓴다. 이중에는 통신 프로토콜과 정보 모델의 구조가 포함된다. 이러한 일반적인 정보 모델과 표준 인터페이스를 이용하여 여러 종류의 장비에 대한 관리를 수행할 수 있다.

TMN에서는 관리 기능을 관리 목적에 따라 다섯 가지의 관리 기능 영역으로 나눈다. 이를 시스템관리 기능영역 (System Management Functional Areas: SMFA)이라고 하며 ITU-T X.700에서 다음과 같이 권고하고 있다. 이러한 관리 기능 영역은 다음과 같다.

- 장애 관리 (fault management) : 장애 발견, 격리, 회복
- 과금 관리 (accounting management) : 통신 자원의 사용에 대한 요금 조정, 감시
- 구성 관리 (configuration management) : 관리 자원의 특성 결정 및 변화
- 성능 관리 (performance management) : 성능 감시 및 조정
- 보안 관리 (security management) : 자원 접근, 조절

TMN을 통한 망 관리는 상기 다섯 가지 망 관리 기능을 제공하기 위해, 관리하려는 목적에 따라 망 자원 및 서비스를 논리적으로 특성화하여 관리 객체를 정의하고 이에 대한 제어 및 감시를 수행한다. TMN 환경 내에서, 관리자(Manager)는 사용자들의 관리 요구를 지원하기 위하여 미리 정의된 관리 객체에 대해 필요한 관리 활동을 수행한다. 이에 대응되는 관리대행자(Agent)는 관리자의 요청에 따라 로컬 환경 내에 있는 관리 객체들을 통해 실제 자원을 관리하고, 실제 자원에서 발생한 사건 보고를 필요시 지역 관리 객체를 통해 관리자에게 통지하는 기능을 수행한다. 또한, 이러한 관리자와 관리대행자간 전면

은 표준화된 공통관리정보 프로토콜(Common Management Information Protocol : CMIP)을 사용하여 이루어진다.

예로서, ATM Agent 시스템은 광대역 종합 통신망 서비스에 필요한 ATM 교환기의 운용 관리 및 유지 보수 기능이 ATM교환기에 종속되지 않고, 이웃의 교환기나 전송시스템, 또는 상위의 운용 시스템간에 표준화된 정보를 상호 교환하거나, 접속할 수 있도록 지원함으로써, 보다 좋은 품질의 서비스를 사용자에게 제공할 수 있다.

다음 장에서는 TMN 개발을 위한 환경구축의 요소들을 개관하고 몇 개의 개발 사례를 보고 개발경험을 통한 개발시 유의 사항을 정리하여 보겠다.

II. TMN 개발 환경 구축

망 요소간 상호운용성이 제공되기 위해서는 망 요소들 사이의 관리 정보의 교환을 위해 표준화된 프로토콜 및 메시지를 가진 운용접면이 필요하다. ITU-TS나 ISO등 국제 표준화 기구에서는 개방 환경하에서 통신망 운용 관리를 위해 OSI(Open System Interconnection) 관리 기법을 표준화하였다. OSI 관리의 표준화 대상은 시스템 관리 기능과 관련된 표준화, 관리 객체(Managed Object)의 정의와 관련된 표준화, 그리고 관리 기능과 관련된 정보를 교환하기 위한 응용계층 서비스 및 프로토콜의 표준화 등으로 구분된다.

TMN을 개발하기 위하여는 몇 가지의 고려사항이 있다. TMN은 표준화를 통하여 구성되었으며 표준을 최대한으로 활용하여 그 장점을 계속 유지하여야 한다. 객체지향형 프로그램이나 모듈화한 설계를 통하여 프로그램의 효율성을 증대시키고 차후에 유사한 용도에 최대한으로 재활용 할 수 있게 한다. 기존의 통신시스템은 표준화된 데이터 통신망을 이용하여 관리될 수 있도록 설계되어야 한다. 위와 같은 개발 환경이 구축되도록 하기 위하여 OAM &P 기능을 위한 개발 도구의 준비, 하드웨어 접면의 정의, 필요한 자원의 안배, OSI 프로토콜 스택과 운영체제 포팅, 검사방법 설정등의 준비가 필요하다.

2.1 GDMO 관리대행자 개발

TMN의 모델은 망요소(Network Element), 망요소 관리(Network Element Management), 망관리(Network Management), 서비스 관리(Service

Management), 비즈니스 관리(Business Management)의 다섯 계위로 나뉜다. 망요소가 가장 낮은 계위이고 비즈니스 관리가 가장 높은 계위이며 각각의 계위는 바로 아래쪽 계위에서 생기는 일들을 다중으로 관리한다.

이러한 관리 기능은 관리대행자와 관리자와의 상호 요청 및 응답에 의해 이루어진다. GDMO(Guidelines for the Definition of Managed Object)관리대행자의 기능은 관리자에 의해서 보내진 CMIP 요청에 의하여 지역 또는 원격지에 있는 관리 객체(Managed Object: MO)와 연결하여 결과를 얻고 그 결과를 관리자에게 응답하는 것이다. 그림1은 관리자, 관리대행자, MO의 관계를 보여주고 있다. 관리 객체는 망 요소 내에 있는 물리적 또는 논리적 자원이나 망을 관리하는 데 필요한 기능을 추상화한 것으로 그 객체가 가지는 성질들을 각각 속성(Attribute), 동작행위(Behavior), 행동(Action), 통지(Notification)등으로 표현한다. 관리 객체를 정의하기 위하여 X.722의 GDMO에서 권고한 정형화된 템플리트를 이용해야 한다.

실제 GDMO 관리대행자 구현은 다음 세 가지 단계를 통해서 개발 노력을 감소시킬 수 있다. 첫 번째 단계로, 모든 관리대행자에 공통으로 들어가는 기능들을 구현해 놓은 일반적인 관리대행자를 작성·구현하고, 이를 검사할 수 있는 일반적인 관리자를 구현한다. 일반적인 관리대행자가 갖게되는 기능은 CMIP

메세지 처리, MO Class Tree, MO Instance Tree(MIT)의 조작, MO의 생성 및 삭제, MO에 메시지 전달 기능 등이다. 그림2는 일반적인 관리대행자로서 가능한 구성을 나타낸 것이다. 두 번째 단계에서 개발자는 사용자의 응용 프로그램에서 요구되는 모든 MO 클래스를 지원하는 관리대행자 프로토타입을 만든다. 이 관리대행자를 실행시키고 시험하여, 개발자는 그의 응용 프로그램이 요구하는 각각의 MO 클래스 특성을 분석할 수 있다. 이 관리대행자를 구현하기 위해서 개발자는 일반적인 관리대행자를 위한 GDMO와 ASN.1 정의 파일들에 관리대행자 프로토타입을 위한 정의들을 덧붙이고 다시 컴파일 해야 한다. 세 번째 단계에서 개발자는 관리대행자 응용 제품을 만들기 위하여 개발자에게 적합한 코드를 삽입함으로써 두 번째 단계에서 만들어진 관리대행자를 맞춤화할 수 있다. 이러한 접근 방법은 사용자 관리대행자의 실행 버전이 설치된 후 짧은 시간 안에 만들어질 수 있는 장점이 있다. 또한, 첫 번째 단계에서의 일반적인 관리대행자를 계속해서 재 사용할 수 있으며, 각 단계별로 점증적인 개발이 가능하다.

이와 같이 사용자 규정의 관리대행자와 MO 구축에 대한 공통 빌딩 블록으로써 일반적인 관리대행자 및 MO 라이브러리 구축한다면, 개발자는 단지 GDMO 규정 파일 및 공통 관리대행자를 구축하기 위한 초기화 코드, 각 관리대행자별로 필요한 맞춤코드를 작성하면 된다.

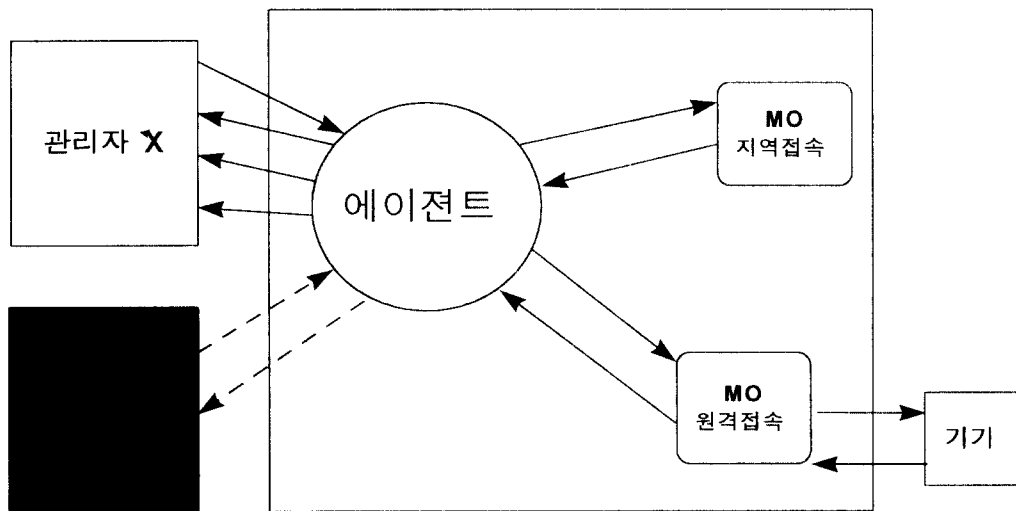


그림 1. 관리자, 관리대행자, MO의 관계

일반적으로, GDMO 표준은 일반성을 바탕으로 설계되어졌으며, MO 추상화는 실세계의 어떤 것이든 모델로 적용될 수 있다. 그러나, 이러한 융통성은 효과적인 GDMO/CMIP 관리대행자 또는 관리자의 구현에 있어서 많은 복잡성을 내포한다. 이러한 복잡성을 줄이기 위해 GDMO 관리대행자와 사용자 정의 MO를 구축하기 위한 기반을 제공하는 각종 개발 도구들을 사용하게 된다.

ASN.1은 특정 프로그래밍 언어와 컴파일러 및 하드웨어 플랫폼에 무관한 독립적인 데이터 타입을 정의하는 데이터 정의 언어이며, 일반적으로 ASN.1 정의를 처리하기 위하여 C 언어로 변환하여 사용한다. 그러나, 많은 ASN.1 타입들은 C 의 복잡한 내부 데이터 구조에서 수행되어야 하고, 대부분의 경우에 이들 ASN.1 타입을 표현하기 위하여 포인터가 무분별하게 사용된다. 이러한 문제를 해결한 새로운 C 프로그래머를 위한 차세대 ASN.1 프로그램 개발도구도 있다.

일단 관리대행자가 구현되면 다음 단계는 구현된 관리대행자가 정상적으로 동작을 하는지를 테스트할 수 있는 CMIP 메시지 기능을 수행할 수 있도록 한다. 즉, 개발한 관리대행자에 대한 관리자 기능을 시

험할 수 있어야 한다. 관리대행자 시험에서는 개발자는 자신의 환경에 맞추도록 초기화를 수행하거나 혹은, 특정 시험 시나리오에서 적용되는 설정 코드를 실행한다.

GDMO 관리대행자 구현시 필수 사항으로는 다음과 같은 항목들을 들 수 있다. CMIP과 ASN.1 데이터를 다루는 코드를 작성하여야 하고 관리대행자와 MO 간의 통신이 효율적이고 동기화가 되어야 하며 관리되는 자원들과는 접면의 표준화가 되어야 하고 보편 타당성 있는 관리대행자가 구축되어야 하며 MO의 구현이 자동화되어 수많은 MO 구현에 어려움이 없어야 하며 관리대행자를 시험할 수 있는 관리자가 구축되어야 한다. 관리대행자를 구현하는 데 있어서 위의 대부분의 사항은 공통적으로 적용되고 또한 한번 개발을 마친 후에는 다른 관리대행자 개발 시에도 재사용될 수 있으나 개발자가 적용사례에 따라 구체적으로 정해야 되는 일들도 있다. MO의 동작행위나 행동이나 통지 등은 사용자가 정의하여야 할 사항이고 에러처리나 관리되는 자원과의 접면 사용자 코드까지의 비동기 접면들도 사용자가 직접 정의하여야 할 사항이다.

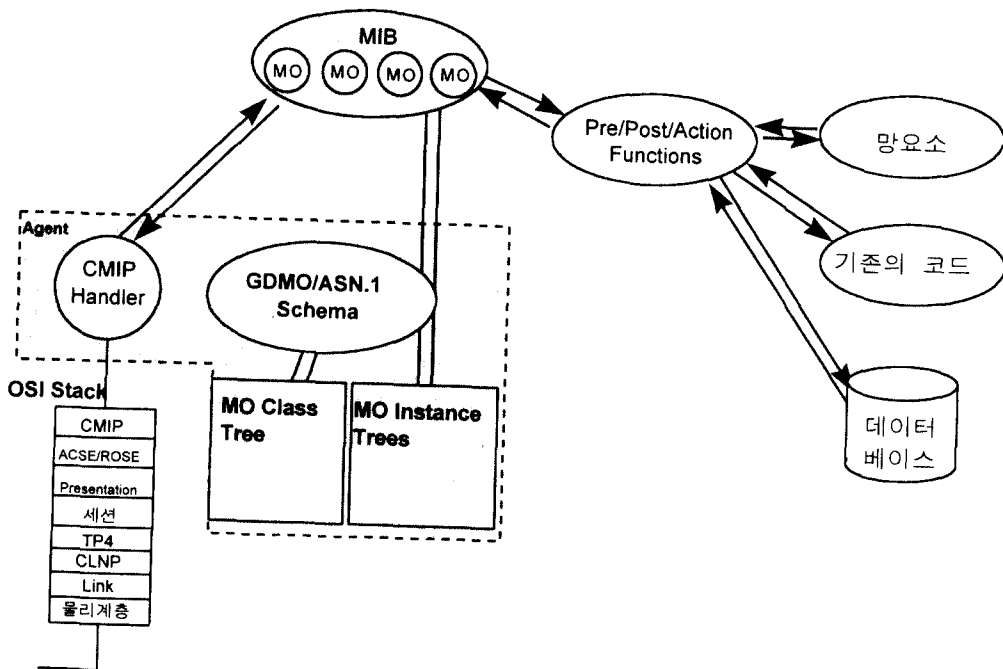


그림 2. 관리대행자의 구성

2.2 OSI 프로토콜 스택

OSI 프로토콜은 수없이 다양한 컴퓨터 하드웨어와 네트워크 구성에 상관없이 상호연결을 가능하게 하기 위한 개방형 시스템을 위한 프로토콜이다. OSI 프로토콜은 물리계층(physical layer), 데이터링크계층(data link layer), 네트워크계층(network layer), 전송계층(transport layer), 세션계층(session layer), 표현계층(presentation layer), 응용계층(application layer) 7개의 계층으로 이루어져 있다. 물리계층은 양 시스템을 물리적으로 연결하기 위해 쓰이는 플러그와 핀의 구성, 비트의 교환 방법, 0과 1에 대한 전자적 표현에 대한 규정이며, 비트들의 전송과 수신 등을 수행한다. 데이터링크계층은 전송 에러의 검출과 복구 흐름 제어를 수행하며, 네트워크계층은 종단 시스템(end system)간의 연결의 설립, 유지, 종결을 수행한다. 전송 계층은 유실 패킷, 잘못된 패킷, 중복된 패킷 등 여러 에러 처리를 수행한다. 이 네 가지 계층은 주로 양 종단시스템(end system)간의 연결을 확립하고 데이터를 보내는 등 연결성(connectivity)을 제공하기 위한 것이다.

세션계층은 양 종단 시스템이 적절한 dialogue 모드에 합의하고, 아주 큰 데이터를 전송하는 경우, 송신 측이 복구시 사용할 수 있는 체크포인트를 삽입하는 등의 데이터 교환을 관리하는 메카니즘을 제공한다. 표현계층은 양 종단 시스템이 서로 다른 데이터 표현 방식을 쓰더라도 교환되는 정보의 유형과 값은 유지 되도록 한다. 마지막으로 응용계층은 가장 상위 계층으로서 하위 계층에서 제공되는 서비스 상에서 다양한 응용 서비스를 제공하며 응용 프로그램의 기본적인 구성 블록이 된다. 세션 계층, 표현계층, 응용계층은 하위 네 계층에서 제공되는 연결성(connectivity) 위에서 연결이 확립된 응용 프로그램간에 상호운용성(interoperability)을 제공하기 위한 것이다.

TMN의 응용서비스인 관리대행자나 관리자 기능서비스는 OSI Model의 7계위인 응용계층 서비스에 해당하고 이 응용 계층 서비스가 나머지 OSI 계위와 잘 연결될 수 있도록 6계위와 7계위 사이의 연결코리가 제공되어야하며 물리계층의 선택에 따라 5계위 이하의 적절한 OSI 프로토콜 스택이 선정되어야 한다.

그림 2의 왼쪽 아래부분은 OSI 관리에 사용되는 OSI 프로토콜 스택의 예를 나타낸다. OSI 프로토콜 스택은 망 요소간의 관리 메시지를 전송하기 위한 하부구조이다.

2.3. Managed Information Base (MIB) 구축

관리정보 베이스(MIB)는 관리 시스템인 망관리를 할 수 있도록 관리 자원들의 정보를 구조화 한 관리 객체의 집합이다.

Managed information base(MIB)에는 ITU 표준에서 정의되어 있는 관리 객체 인스턴스들이 NAME BINDING 패키지 정의에 따른 포함 관계를 갖는 트리 형식으로 구성이 되며, 관리자 또는 관리대행자에 의해 생성, 삭제된다. 구현된 MIB의 루트는 시스템 관리 객체 인스턴스이다.

각 MO 인스턴스에 대해서 생성, 삭제 및 속성 값을 변경할 경우 사용자는 해당 MO 인스턴스 및 속성(attribute)에 대해 사용자 정의 함수를 추가할 수 있다. 이 함수들은 사용자 개발 환경에 적합하게 관리대행자를 맞추기 위해 사용된다.

이제 TMN은 시대적인 흐름이 되었고, 앞으로의 통신시스템은 이에 대한 요구를 당연시하고 있다. 이러한 흐름에 따라서 망요소 또한 그 자체의 하드웨어 뿐만 아니라 그 위에 탑재되는 운용보전기능의 관리대행자는 한 층 더 중요성을 더하게 된다.

Ⅲ. 개발 사례

3.1 국내 개발 동향

국내에서는 한국통신 연구개발 본부에서 광가입자 장치와 ATM MAN 교환장비를 개발하면서 TMN 소프트웨어 개발이 시작되었다. 관리 객체 추출에 있어서 일반적인 망관리 기능은 X.721에서 권고한 관리 정보 모델, M.3100에서 권고한 일반적인 망 정보 모델, TR836 및 TA836에서 정의한 망요소에 대한 전송 구성 및 감시를 위한 정보 모델 등을 참조하고 개발 시스템에 고유하게 요구되는 기능은 추가하고 필요하지 않은 기능은 제거하는 과정을 통하였다. 초기에는 관리 객체에 많은 시간을 투자하였으며 개발환경에 여러 가지 시행착오가 있었으나 이제는 핵심기술을 확보하게 되었고 한국통신과 관련된 개발업체들도 TMN개발 기반 기술을 보유하게 되었다. 한국전 자동신 연구소 또한 일찍부터 TMN에 관한 연구가 계속되어 오다가 TDX 교환기, 새로운 전송장비 등에 적용 검토 및 개발 중이다. 차후로는 각종 무선통신의 운용분야에 TMN 기술이 집중적으로 사용되리라 예측된다. 또한 각 대학 및 기업 연구기관에서도 활발한 연구개발이 계속되고 있다. 관리 정보 모델의 적합성 시험은 표준화가 이루어져 있지 않은 상태이

나 OSI 프로토콜 스택 구현에 관한 부분은 적합성 명세가 권고된 상태이고 이에 따른 적합성 여부가 가능한 한 시험되고 있다.

3.2 유럽의 GSM망 응용 예

1988년부터 유럽 통신 표준기구(European Telecommunication Standard Institutes : ETSI)는 GSM 표준을 제정하기 시작하였고 유지관리 보수책임은 맡은 분과위원회(SMG 6)에서는 TMN 표준을 최선을 다해 따르도록 결정하였다. GSM 시스템의 배경은 유럽전체에 통용될 수 있는 공통된 디지털 이동전화 시스템의 보급에 있었기 때문에 망을 구성하는데 있어서 다중의 장비 제공자가 지원할 수 있어야 하고 그러기 위해서는 기능과 각종 망 접면의 표준화로서만 가능하였다. 유지보수측면에서는 관리접면과 관리기능의 표준화를 의미하였다. SMG6 분과위원회는 GSM TMN 구조를 수립하였고 OS와 망요소간의 TMN Q3 접면을 정의하였다. 이 위원회는 5개의 관리기능 분야에서 각 분야마다 규격 안을 마련하였고 Q3 접면에 이어 기지제어부(BSC)와 기지전송부(BTS)간의 QX 프로토콜을 규격화하였다. 규격을 정하는 데 있어서 시간의 제약과 GSM 시스템 고유성을 감안하여 ITU-T가 추천하는 Top-down 방식과 새로운 시스템을 구현하면서 얻은 Bottom-up 방식이 혼용되었다. GSM TMN 구조에서는 100개 이상의 관리객체 클래스(Managed Object Classes)가 500개 가량의 속성과 함께 정의되었다. 개발중인 대규모의 GSM 시스템에 TMN 방식을 도입하여 실용화한 것이 큰 의의라 하겠다. 관리모델에 주요관리 분야를 모두 상세하게 정의하였고 이 모델은 하나의 기기 관리 객체를 제공하고 기능과 물리적인 관점의 링크를 제공함으로써 시스템의 고유기능을 표준화하려 노력하였다.

IV. 개발경험

4.1 고려 사항과 유의 사항

TMN 소프트웨어 개발은 망운용 시스템의 핵심적인 소프트웨어 개발로 자리잡아 가고 있다. 최근 국내에서 개발중인 몇 개의 TMN 소프트웨어 개발에 직접, 간접으로 참여하게 되면서 프로젝트 시작 시대수롭지 않게 여겼던 부분들이 후에 프로젝트 수행도중 여러 가지 문제를 일으키는 것을 보게 되었다. TMN의 설치를 선호하는 이유들을 보면 ITU-T TMN표준이 성숙 단계에 이르렀고 객체지향형 소프

트웨어나 소프트웨어 재사용이 새로운 소프트웨어의 개념으로 정착하게 되었고 장비제공자들이 점차로 OS 플랫폼과 장비들을 제공하며 상호운용성을 위한 개방형 시스템들이 제공되는 것 등이다. 하지만 아직도 ITU-T에서는 TMN의 구체적인 구현 방법을 제시하기보다는 장비와 서비스 제공자의 구현에 의존하고 7계위의 OSI 프로토콜 스택을 사용하는 것이 복잡하고 성능저하가 종종 생기며 기존의 운용 시스템과의 상호 운용문제, 상당한 초기 투자 등이 걸림돌이 된다. 이러한 경험을 토대로 추후 프로젝트 운용에는 프로젝트에 예상되는 문제점들을 조기에 발견하고 고려 사항들을 프로젝트 진행 단계마다 적절히 배합하여 프로젝트가 원만하게 진행될 수 있도록 기여하고자 한다.

일반적으로 프로젝트의 진행 과정을 보면 그 프로젝트가 필요한 지의 여부를 따지는 평가단계와 개발을 할 수 있도록 사양을 전하는 정의단계와 프로젝트의 중심이 되는 개발단계와 개발 후에 각종 시험을 거치며 제품을 완성하고 배포하는 배포단계를 나눌 수 있다. TMN 소프트웨어의 개발은 이와 같이 평가, 정의, 개발, 배포 단계로 나눌 수 있다.

4.2 평가단계

프로젝트의 생성과정을 보면, 초기 평가단계에 있어서는 고객의 요구를 참작하여 판매, 마케팅, 기술 부서 또는 경영 층에서 초기의 제안을 하게 된다. 이때에 마케팅 부서에서는 적극적인 참여와 제품(TMN S/W)을 기획하고 시장의 요구 사항을 파악하여 경영 층의 제품 개발에 대한 결심을 얻어내야 한다.

제품 기획 단계에서의 마케팅부서의 적극적인 참여 결여는 추후에 자칫 주인 없는 제품을 만들거나 고객의 요구와는 상관없는 제품을 만들게 될 염려가 있다. 기존의 망운용 시스템을 새로 만드는 망운용 시스템에 어떻게 접목시킬 지와 초기투자의 타당성 검토, 기존 관련제품의 유무 등이 조사되어야 할 것이다.

이 단계에서의 TMN S/W 제품 개발에 대한 평가는 이 제품이 기존의 다른 망운용 제품과는 어떻게 다른지, 제품이 개발되면 관련 고객들이 어떻게 도움을 받는 것인지 등의 고객의 요구 사항이 가능한 한 많이 파악되어야 할 것이다.

4.3 정의단계

두 번째 단계는 프로젝트의 정의 단계이다. 제품의 정의는 누가 제품의 소유자가 되며 어떠한 팀이 구성되어 개발하는가를 결정하고, 제품의 기능 사양을 결정하며 제품 계획을 마무리하고 개발 주체와 제품의 소유자간에 계약을 체결하며 프로젝트 개시를 위한 최종 검토를 실행한다. 이 단계에서 경영 층이 OK를 하여야만 제품 개발이 시작될 수 있다.

대부분의 프로젝트 정의단계를 보면 프로젝트의 진행이 프로젝트의 예산획득과 맞물려 있고 프로젝트의 정의단계에 필요한 각종 단계에서의 진행과 경영 층의 의사 결정 시기가 연도별 예산 획득 과정과 엇나갈 때에는 프로젝트 진행과 그에 필요한 결정을 하기 보다는 다분히 예산 획득 과정에서 일어나는 회계 연도 중심의 사업 진행이 될 가능성이 많다. 이는 프로젝트 전체의 개발 주기가 짧아지고 유연한 의사 결정에 의한 경쟁력 있는 제품 개발에 지장을 준다.

프로젝트 정의단계에서는 적절한 예산과 인력이 확보되어야 하고 프로젝트 수행에 대한 경험이 적을수록 추후 프로젝트 세부 진행 사항에 대한 변경이 필연적이므로 예산 확보 및 추가 확보 방안, 전문 인력 조달 방안 등이 충분히 고려되어야 한다. 특히 프로젝트를 여러 팀이 같이 하게 될 경우는 프로젝트를 세분하여 각각의 소규모 프로젝트 진행에 대한 역할 분담이 명확하게 이루어져야 되고 소규모 프로젝트간의 연결이 자연스럽게 될 수 있도록 프로젝트 관리를 강화하여야 한다. TMN 프로젝트 진행 도중 후발주자로 개발에 참여하게 되는 경우 전체적인 프로젝트의 방향, 정의 내역 등을 몰라 개발 도중 혼선을 빚는 예가 있었다. 소프트웨어 개발 같은 경우는 개발 환경의 여하에 따라 개발 팀간의 업무분장은 물론 문제 발생시 해결이 하드웨어 개발보다 훨씬 어렵게 된다.

프로젝트 관리 팀은 각각의 개발 팀이 프로젝트 전반의 흐름을 알 수 있도록 각 팀에게 프로젝트 브리핑을 강화하여야 할 것이다. 각 팀에게 프로젝트 브리핑이 소홀할 경우에는 팀간에 모르는 일들이 진행되다가 추후 프로젝트 통합시 많은 낭비가 뒤따르게 된다. 또한 프로젝트 정의 단계에서의 전문가의 입력은 매우 중요하고 전문가의 충고를 따르지 않았을 경우 추후에 다가올 문제점들을 해결하는 데 매우 큰 어려움이 따를 것을 각오하여야 한다.

4.4 개발단계

개발단계는 실제로 전문인력이 참여하여 예정된 부

분의 디자인 개념을 구현하고 구현된 부분에 대하여는 검토를 거쳐 디자인이 목표의 달성 여부를 심사한 후 성공한 부분들을 함하여 일차 제품을 구현하게 된다. 이때 경영 층 또는 그 부분의 전문가의 검증을 거쳐야 되며 이때 이 과정을 통과하면 배포단계로 가지만 그렇지 않으면 다시 부족한 부분을 보완, 구현하여야 한다. 무리한 개발 일정이나 개발 인력의 부족, 개발 환경의 미비는 개발 팀의 사기를 저하시키고 결국은 주어진 자원을 활용하지 못하는 결과를 낳게 된다. 개발 중에는 적절한 개발도구의 설정이 필수적이며 개발자간의 공동된 개발환경이 개발시간 단축에 필수적인 요소가 된다. 제품의 구현을 마치고 검증을 서치는 단계에서는 고객의 검증 또한 빼어 놓을 수 없다.

4.5 배포단계

배포단계는 고객이 직접 시험하는 기간을 거쳐야 된다. 선행 시제품, 실용 시제품 등의 과정을 거쳐 마지막 고객의 요구사항을 반영하고 가격 조정 등을 거치면 상품으로서 고객의 필요에 부응하게 되는 것이다.

V. 맺음말

TMN(Telecommunication Management Network)은 급격한 네트워킹과 네트워크 서비스의 발전으로 더 한층 그 중요성이 강조되게 되었다. 멀티미디어, 무선통신등 일차 하드웨어 개발의 벽을 넘은 다음에는 그 보다 훨씬 큰 분량의 소프트웨어의 개발이 기다리고 있다. 통신망 운용에서 GUI상에 나타난 부분이 고객이 접할 수 있는 통신망 운용의 얼굴과 같고 피부와 같은 것이라면 TMN 소프트웨어는 통신망 운용의 뼈대가 되어 우리에게 다가온다.

통신 시장의 환경을 보면 밖으로는 규제완화와 상호운용성이 강조되고 안으로는 고객의 욕구가 끊임없이 변하고 있다. 예를 들면 고객은 단순히 더 많은 대역 폭을 원하기보다는 필요한 때에 필요한 분량만큼만 쓸 수 있는 주문형 대역 폭을 요구하게 되었고, 많은 서비스 제공자와 장비제공자가 공존하게 되었다. 이러한 다양한 환경속에서 고객의 서비스를 충족하기위하여는 통신망 운용 유지 보수등, 모든 통신망 운용 시스템 디자인이 유연성이 있어야 하고 통신운용분야에서는 어렵게 표준화한 ITU의 TMN이 그 해답의 가능성을 보여주고 있다.

자연자원이 적은 우리로서는 통신과 소프트웨어 부분이 접합된 TMN 부분을 집중투자함으로써 안으로는 통신 부분에 다양한 서비스를 제공하게 되고 밖으로는 축적된 우리의 기술을 나눌 수 있는 발판으로 사용될 수 있었으면 좋겠다.

참 고 문 헌

[1] 송주빈, 김재근, 이성경, "전송시스템의 OAM 기술", 한국통신학회지 8권 7호, pp49~pp61, 1991년 7월

[2] Thomas T. Towle, "TMN as Applied to the GSM Network", IEEE Communication Magazine, March 1995, pp 68-73

[3] Patrick McLaughlin, "TMN : Pitstops and Pitfalls for TMN Engineers" 7th World Telecommunication Forum Technology Summit Vol.1 pp473~pp477, Genova, Oct, 1995

[4] Benard S. Ka, "Experience in the Use of Telecommunications Management Network(TMN) Architecture for SONET Network" 7th World Telecommunication Forum Technology pp491~pp494, Geneva, Oct, 1995



이 증 희

- 1971년 : 서울대학교 전자공학과 (학사)
- 1980년 : 美 펜실바니아 대학원 시스템공학(석사,박사)
- 1990년 : 대영전자(주) 기술연구소장 역임
- 1991년 : 동진정보통신(주) 대표이사 역임
- 현재 : 모다정보통신(주) 대표이사에 재직중