컴포넌트 기반의 망관리 시스템 개발에 관한 연구

김 행 곤† · 김 지 영††

요 약

인터넷과 웹의 확산으로 네트워크 기반의 분산 환경은 응용의 표준 아키텍처로 인식되고 있다. 또한 네트워크의 효율성과 최상의 서비스 제어와 공유를 위해 복잡한 네트워크 구성 자원들을 관리하는 망 관리 시스템이 요구되고, 이는 응용의 표준 하부 지원 시스템으로서 뿐 아니라 독립적인 상업적 응용으로서 수요와 기대가 점차 증가하고 있다. 하지만 특정 프로토콜이나 특정적인 벤더의 장치에 의존적으로 분산되어져 있고, 이기종의 분산형 네트워크 자체의 성질 때문에 이들 시스템들을 통합하고 일관성 있게 관리할 수 있는 표준화된 망관리의 필요성이 요구되고 있다. 또한 소프트웨어 개발 방법 측면에서는 패턴과 컴포넌트에 의한 조립, 확장을 중심으로 하는 소프트웨어 재사용이 소프트웨어 생산의 기대치를 현실화할 수 있는 최상의 접근 방법으로 인정된다. 이에 따라 잘 정의된 인터페이스를 통해 좀더 쉽고 빠른 응용을 개발 가능하게 하는 컴포넌트 기반 개발 방법론을 기반으로 컴포넌트를 구축, 선택, 조립함으로써 높은 품질과 생산성을 보장한다. 따라서 본 논문에서는 분산 망관리 시스템 개발을 위해 망관리 아키텍처를 정의하고, 망관리 설계패턴과 컴포넌트들을 식별, 정의하여 컴포넌트 아키텍처에 매핑한다. 또한 이를 통해 컴포넌트 개발과 유통, 사용을 위한 컴포넌트를 명세하고, 컴포넌트 설계를 통해 이를 구현하였으며, 구현된 컴포넌트들은 등록, 검색 및 이해할 수 있는 컴포넌트 저장소 시스템으로 적용하고, 미리 구현된 컴포넌트를 통해 전체 망관리 시스템을 분석/설계, 구현하였다.

A Study on Development of Network Management Systems base on Component

Haeng-Kon Kim† · Ji-Young Kim††

ABSTRACT

With growing population of internet and web applications, distributed environment is considered to be the standard architecture of application. A network management systems(NMS) is necessary to control and monitor the complex network resources for providing and sharing the best quality service. We recognize the NMS as a standard infrastructure for supporting efficient networking and a separate commercial applications. We believe every resource including software, hardware and environment for the network management should be separated from special protocols, vendors and applications. Therefore, We need a standard network management system that is efficient and consistent because of the heterogeous network features. In regards to software development, software reuse through assembling and extending the reusable elements such as patterns and components assures to realize the best productivity and quality. The component based development(CBD) methodology that can assemble black box though well defined interfaces makes it possible to develop easer and quicker applications and is proved as the best software development solution involved in construction, selection and assembly of components. In this thesis, we describe the architecture for the network management and identify, define and design the components through analysis and design in the network management domain and identified components mapped to the component architecture. We also specify the component development and design and implement the component for developing the network management. Implemented components apply to the component repository system that register, retrieve and understand the components. We analyze, design and implement the entire network management system based on configuration, connection, performance and fault management through the pre-developed components.

키워드 : 망 관리 시스템(Network Management System), 컴포넌트 기반 개발(Component Based Development), 컴포넌트 아키텍처(Component Architecture), 컴포넌트 명세(Component Specification), 컴포넌트 저장소(Component Repository)

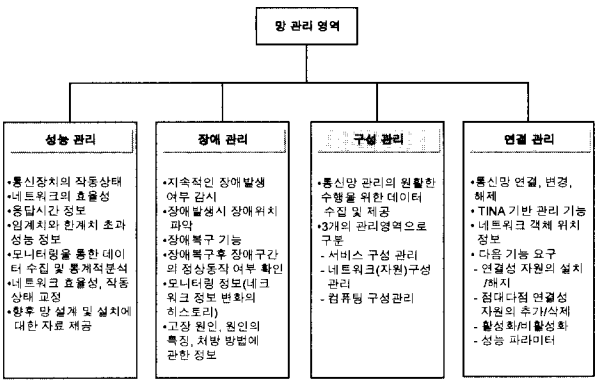
1. 서 론

최근 네트워크 환경은 기업의 규모와 조직체계가 증가됨에 따라 네트워크의 환경 또한 다양화, 거대화 및 추상화의 양상을 띠고 있다. 컴퓨터 산업의 빠른 발전으로 인하여 인터넷 사용자가 기하급수적으로 증가하고, 웹 기술을 기반으로 하는 다양한 인터넷 응용 서비스들이 지원됨에 따라 네트워크 기반의 분산 환경이 애플리케이션의 개발과 활용을 위한 표준 아키텍처로 자리 잡아가고 있다. 따라서 망관리는 개별 애플리케이션으로서 뿐만 아니라 대부분의 애플리케이션의 하부계층 지원 서비스로 요구되는 주요한 도메인이 되고 있다[1]. 또한, 망 기반의 분산 환경이 대부분의 비

† 종신회원 : 대구가톨릭대학교 컴퓨터공학과 교수
†† 준 회 원 : 대구가톨릭대학교 대학원 전산통계학과
논문접수 : 2003년 12월 26일, 심사완료 : 2004년 3월 29일

즈니스 영역에서 많은 비중을 차지하고 있고 보급이 확대됨에 따라 망의 문제로 인한 손실이 높아지게 되면서, 이를 효율적이고 일관성 있게 관리하기 위한 개방형 네트워크 구조에 맞는 표준화된 네트워크 관리가 필요하다. 하지만, 통신망의 운영관리 기술분야에서 국내의 기술 수준은 외국에서 개발된 망관리 플랫폼을 사용하여 Telecommunication Management Network(TMN)/Common Management Information Protocol(CMIP) 체계 또는 Simple Network Management Network(SNMP) 체계를 기반으로 기능을 구현하는 정도이며, 복잡한 망관리 기능의 구현은 외국 전문회사에 의뢰하는 경우가 대부분이다. 분산 망의 효율성과 최상의 서비스를 제공하기 위해 이종의 네트워크 시스템 환경들이 통합된 인터네트워킹 상태에서 망관리 시스템을 통해 네트워크 상태를 확인하고 모니터함으로써 망의 효율성과 생산성을 극대화 할 수 있는 표준화되고 통합되어진 망관리 시스템의 필요성이 요구되고 있다[2]. 소프트웨어 개발 방법 측면에서 볼 때, 소프트웨어의 규모가 점차 복잡해지고 대형화됨에 따라 사용자 요구 사항의 빠른 변화를 만족시키기 위한 새로운 소프트웨어 개발 방법이 필요하게 되었다. 특히 하루가 다르게 요구사항이 급변하는 현재의 추세로 볼 때 적시에 최적의 소프트웨어를 제공하는 일이 무엇보다 중요해지고 있고 언제, 어디서나, 누구나 필요한 정보를 쉽게 얻을 수 있는 인터넷 환경이 보편화되고, 이기종 컴퓨터간 연동기술의 발전으로 인터넷상의 다양한 소프트웨어 부품을 기종에 상관없이 'plug & play' 형태로 조립하여 사용하려는 노력이 급증함으로써 새로운 소프트웨어 개발 방법론이 절실히 요구된다. 따라서, 소프트웨어의 품질을 보증하고 재사용을 통한 소프트웨어 개발 생산성을 향상시키기 위한 방법으로 소프트웨어를 부품화하고 이를 조립, 합성하여 애플리케이션을 개발하는 컴포넌트 기반 개발(Component Based Development : CBD)이 최근에 각광받고 있다. 컴포넌트는 특정 기능을 수행하기 위해 독립적으로 개발되고 잘 정의된 인터페이스를 가지고 다른 컴포넌트와 조립되어 시스템을 구축하기 위한 소프트웨어 단위를 의미한다. CBD는 컴포넌트의 개발은 물론 최적의 컴포넌트를 선택, 조립하여 시스템을 생성하는 새로운 소프트웨어 개발 방법론으로서 적은 비용으로 빠르게 애플리케이션을 개발하는 최선의 방법으로 간주되며, 기존 컴포넌트의 재사용을 통하여 소프트웨어 개발 생산성을 향상 뿐만 아니라 소프트웨어의 품질 또한 보증할 수 있다. 이 방법론으로 개발된 애플리케이션은 산업계에서 빠르게 부각되고 있는 최신 분산객체 기술과 자연스럽게 조화될 수 있다[3, 4]

본 논문에서는 컴포넌트 저장소를 기반으로 망관리 시스템 개발에 CBD 적용을 위한 기술적 접근 방법들과 망관리 시스템 개발을 위해 분산 컴퓨팅 기반의 솔루션을 위한 모델인 ABCD(Architecture/Base/Common/Domain) 컴포넌트 아키텍처를 참조하여 망관리 시스템 구축을 위한 선행 작업으로 아키텍처를 정의하며, 컴포넌트를 식별, 분류하고 컴포넌트를 명세, 구현한다. 구현된 컴포넌트들은 등록, 검색 및 이해할 수 컴포넌트 저장소 시스템을 통해 미리 구현된 컴포넌트를 통합함으로써 CBD 기반의 망관리 시스템을 개발하고자 한다.



(그림 1) 망관리 영역별 기능

2. 관련 연구

2.1 망관리 시스템

망관리 시스템은 네트워크의 전반적인 상황을 체크하고, 문제점을 알려주는 기능을 수행하는 하드웨어 또는 소프트웨어를 통칭한다. 인터넷을 비롯한 비즈니스 전반의 데이터 네트워크는 다양한 사용자층, 다양한 목적과 용도로 확산되는 추세이며, 보급이 확대됨에 따라 많은 비용의 발생과 문제점들이 도출되고 있다. 이에 따라 사용자들에게는 성능대 가격비에 적절하고 유용한 기능을 보장해 주며, 관리자에게는 망의 안정성과 유연성을 최대한 지원할 수 있는 표준화된 관리가 필요하며 이에 대한 해답이 바로 Network Management System(NMS)이다. NMS는 네트워크를 구성하는 각종 요소로부터 구성, 장애, 연결, 성능 등에 관한 데이터를 수집, 분석하여 중단 없는 네트워크 서비스를 보장하고 효율적인 네트워크 제어를 지원해준다. 일반적으로 OSI 구조에 기반한 장애, 구성, 계정, 성능, 보안 관리의 영역이 제공된다(그림 1). 본 논문에서는 구성, 연결, 성능, 장애 관리를 기반으로 망관리 컴포넌트를 식별하고 설계, 구현한다[5-9].

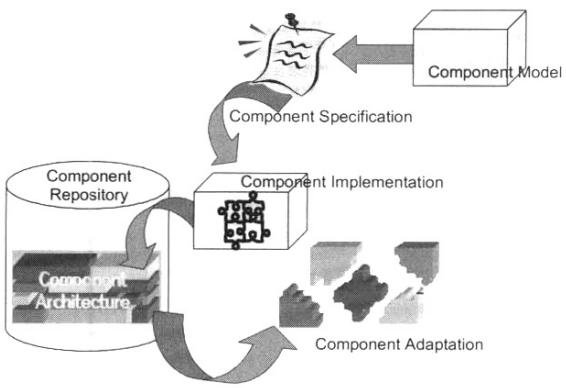
2.2 설계 패턴과 컴포넌트 기반 개발

설계 패턴은 특정 문제에 대한 해결책을 하나의 패턴으로 추상화시킴으로써 같은 문제에 대한 경험을 재사용할 수 있도록 제안 되었고, 객체지향 설계들을 생성하는데 유용한 일반적인 설계 구조를 추상화함으로써 설계단계의 재사용 방법을 도입하였다. 따라서 구현 단계의 재사용에서

플랫폼 환경과 애플리케이션 도메인의 한정성에 대한 문제는 설계 패턴을 통해 극복할 수 있다. 설계 문제의 추상화와 특정 영역에 대한 공통적인 해결책을 구성 요소간의 관련성을 정의함으로써 공통 도메인의 애플리케이션 구축에서 아키텍처의 자동 생성을 가능하게 하는 설계 패턴은 애플리케이션 도메인에 대한 신뢰성 있는 설계 지식을 확보하면서 개발자 나름대로의 구현 특성을 제공할 수 있다[10]. 설계 패턴 라이브러리는 다중 애플리케이션 도메인에 적용될 수 있는 반복적으로 적용되는 규칙이나 관련성의 집합인 패턴들을 UI 패턴, 이벤트 핸들러, 테이터베이스 모델 등의 형태로 이용하는 것이다. 즉, 패턴 라이브러리로부터 적절한 패턴을 식별하고 수정하여 완벽한 모델을 작성하고 기존의 도구를 이용하여 자동적으로 코드를 생성하고 애플리케이션을 인스톨한다. 패턴은 경험있는 다양한 그룹들이 인터넷을 통하여 공유할 때 더욱 가치가 있으며 최근에는 패턴의 조립에 초점을 두고 있다[11, 12].

컴포넌트 기반 개발은 컴포넌트를 개발하는 방법뿐만 아니라, 최적의 컴포넌트를 선택하여 이를 조립함으로써 새로운 애플리케이션을 생성하는 새로운 소프트웨어 개발 방법론으로 높은 융통성과 유지보수성을 제공하므로 소프트웨어 개발 생산성을 향상시키고 소프트웨어의 품질 또한 보증할 수 있다. 또한 CBD 방법론으로 개발된 애플리케이션은 산업계에서 빠르게 부각되고 있는 분산객체 기술과도 자연스럽게 조화될 수 있다. 다음은 두 가지 관점에서 CBD 프로세스를 기술한 것이다[13].

- 컴포넌트 개발(Reuse for component) : 도메인 분석에서 시작하여 컴포넌트 식별, 추출, 개발에 관련된일련의 행위들로 고수준의 재사용성을 가진 실행 가능한 컴포넌트를 생성하는 과정
- 컴포넌트 기반 개발(Reuse with component) : 이미존재하는 컴포넌트와 결과물들의 활용을 극대화함으로써 비즈니스 요구에 대한 솔루션 시스템의 구축을 지원하는 과정

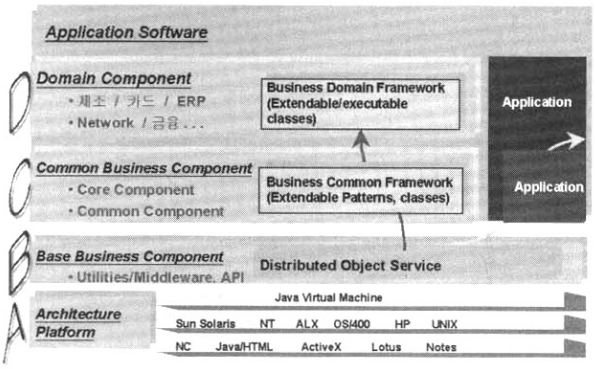


(그림 2) CBD 구성 요소

이러한 프로세스는 컴포넌트 저장소를 중심으로 컴포넌트와 관련된 기본적인 행위들 즉, 컴포넌트 아키텍처, 명세, 검색, 적용 및 조립에 의한 시스템 전개라는 CBD 행위들을 말한다. (그림 2)는 CBD 전개를 위한 구성 요소들을 나타낸다[14, 15].

2.3 ABCD 컴포넌트 아키텍처

컴포넌트 기반의 비즈니스 솔루션을 위한 표준 모델로서 ABCD 컴포넌트 아키텍처는 멀티 벤더/멀티 솔루션의 통합을 위해 컴포넌트의 스코프와 추상성, 입자성을 기준으로 계층적 분류되었다. ABCD 아키텍처 정의를 위한 원칙은 비즈니스에 필요한 컴포넌트와 일반적인 시스템 요구 컴포넌트로 구분하고, 기반이 되는 필수 컴포넌트와 부가적인 선택적인 컴포넌트로 구분한다. 또한 완성되어 수정할 수 없는 컴포넌트와 패턴 형식의 커스터마이징이 가능한 컴포넌트로 분류하며 컴포넌트가 제공하는 서비스의 범주에 따라 계층적 관계를 형성하는 군을 정의한다. 그리고 기존에 정의된 분산 서비스의 컴포넌트를 계층에 포함시킨다. (그림 3)은 ABCD 컴포넌트 아키텍처이다. 전체 4개의 계층으로 구성되며 각 계층의 특징은 <표 1>과 같이 정의된다. 본 논문에서는 컴포넌트 기반의 소프트웨어 프로세스 정의를 위한 선행 작업으로 본 연구실에서 이전에 연구되어진 컴포넌트 참조 아키텍처 모델을 기반한다[16].



(그림 3) ABCD 컴포넌트 아키텍처

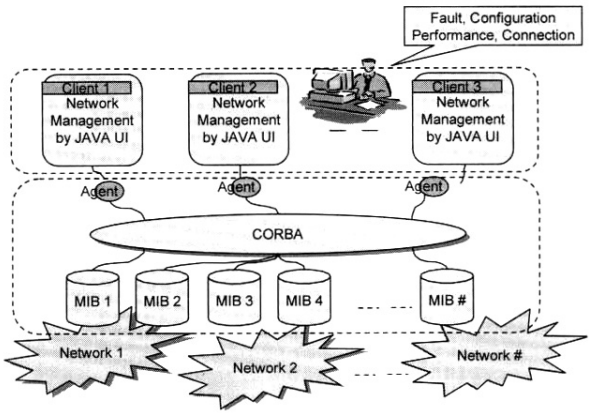
〈표 1〉 컴포넌트 아키텍처의 각 계층들

계 층	이 름	특 징
1	Architecture Platform	◦ 분산 컴퓨팅 환경에서 멀티 벤더/응용 구축을 위한 하부적인 물리적 플랫폼
2	Base Application Component	◦ 분산 컴퓨팅을 위한 미들웨어적인 통합API 및 기존의 분산 객체 서비스 ◦ 모든 응용에 하부적인 기능 수행에 필요한 공통적인 컴포넌트 ◦ 공통적인 기능성의 멀티 응용 컴포넌트
3	Common Business Component	◦ 비즈니스 응용을 위한 실행성의 기능성 컴포넌트
4	Domain Component	◦ 각 비즈니스 영역별 필요 컴포넌트

3. 망관리 시스템에 CBD 적용을 위한 기술적 접근

3.1 망관리 시스템 아키텍처

망관리자는 인터네트워킹 상에서 클라이언트의 단일 인터페이스 시스템을 통해 망의 상태를 확인, 모니터링할 수 있다. 이를 위해 망관리 시스템은 이질적인 클라이언트/서버의 연동을 위해 미들웨어 CORBA를 이용하여 통합된 (그림 4)와 같은 아키텍처를 제시한다. 분산된 다수의 망관리자는 인터네트워킹 상에서 JAVA로 구현된 단일 관리 인터페이스 시스템을 통해 망 상태를 감시하고 정보를 엑세스하며 동적으로 모니터링 한다. 즉, 분산처리 환경에서의 통신 망관리와 다양한 멀티미디어 서비스를 지원하는 TINA (Telecommunications Information Networking Architecture) 구조를 기반으로 TINA의 관리 영역을 기초로 구현된 관리 시스템들은 CORBA 기반의 통합 네트워킹 환경 하에서 클라이언트 JAVA 애플리케이션을 통해 분산된 MIB(Management Information Base)로부터의 관리 정보 액세스를 위해 표준화된 인터페이스를 이용함으로써 멀티 플랫폼/장비/응용 자원을 관리한다.



(그림 4) 망관리 시스템 아키텍처

3.2 망관리 설계 패턴 및 컴포넌트 식별

본 논문에서는 망관리 시스템 개발에 필요한 시스템 자원들의 패턴화 및 컴포넌트화를 위해 망관리 컴포넌트간의 관련성을 정의, 일반화하며 실행성의 컴포넌트를 식별함으로써 관리를 위한 표준 자원을 제공하고, CBD를 위한 풍부한 조립 자원을 제공할 수 있도록 한다.

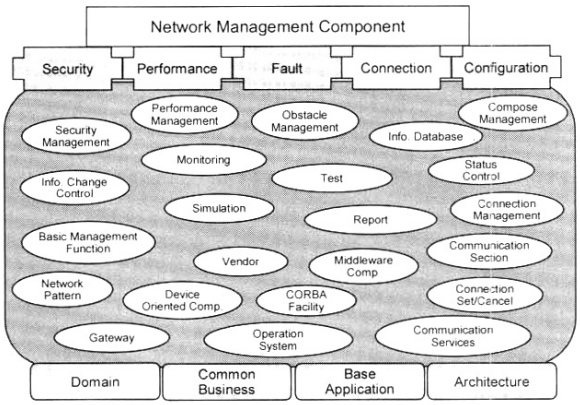
3.2.1 망관리 설계 패턴

도메인 분석을 통해 모델링 한 후 이 정보를 바탕으로 망관리 패턴을 생성할 수 있다. 예로, 웹 기반의 JAVA API를 통한 망관리를 위해 클라이언트는 웹 서버로부터 망관리 UI를 다운로드하고 CORBA IIOP를 통해 서버로 관리 정보를 요청하고 서버는 해당 관리 시스템 정보를 클라이언트에게 디스플레이 한다. <표 2>는 식별된 망관리 패턴들이다.

3.2.2 망관리 컴포넌트

TINA의 관리 영역은 크게 장애, 구성, 연결, 과금, 성능, 보안 관리로 나누어지며, 이 중에서 연결 관리는 TMN 체계에서의 구성 관리에 속하는 통신망 연결 설정, 변경, 해제 기능을 연결 관리라는 독립적인 하나의 관리 기능으로 정립한 것이다. 망관리 영역별 기능들을 토대로 망관리 구조는 장비별, 지역별, 특성별로 상이한 망 구성 요소들을 통합한 관리로 일관성 있는 망관리 및 유지보수 체제를 구축하고, 망 구성 요소들의 장애 발생시 신속한 복구를 통한 시스템의 안정성을 확보한다. 또한 트래픽 병목 회선을 주기적으로 모니터링하고 분석하여 사용자에게는 네트워크 서비스의 향상된 품질을, 관리자에게는 효율적인 망 설계를 지원한다. 이렇게 수집되고 분석된 데이터들은 망관리에 필요한 통계 데이터의 전산화를 통해 최적의 시스템 설계를 지원하여 저비용으로 고효율의 망을 관리할 수 있다.

본 논문에서는 TINA 구조의 구성, 연결, 성능, 장애 관리 영역만을 주 범위로 하였으며, 각 영역별 컴포넌트는 (그림 5)와 같이 식별할 수 있다. 그 중 연결 관리 컴포넌트를 대상으로 하여 컴포넌트 기반 개발의 핵심 선행 작업인 영역별 컴포넌트 식별과 분석 후 명세하여 재사용 가능한 컴포넌트를 추출한다.



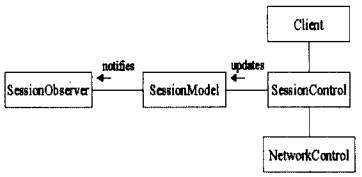
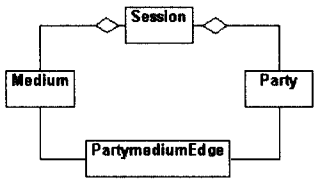
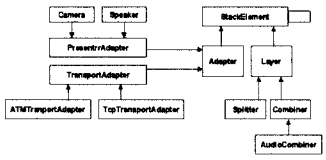
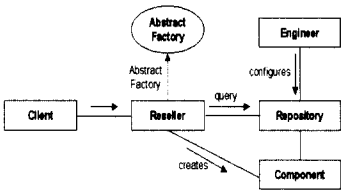
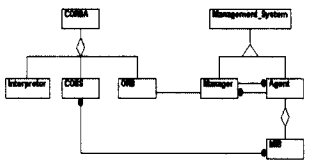
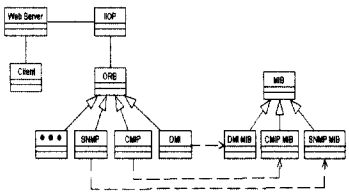
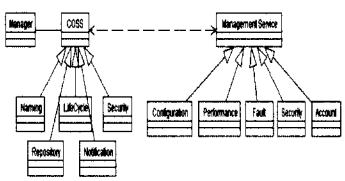
(그림 5) 식별된 망관리 컴포넌트

3.3 컴포넌트 기반의 망관리 시스템 아키텍처

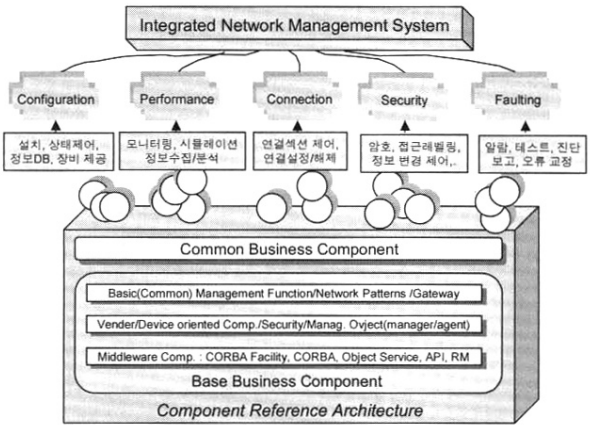
망관리 시스템 구축에서 요구되는 서비스 모듈들과 앞에서 제시한 ABCD 아키텍처에 기반하여 망관리 영역에서 요구되는 컴포넌트 계층을 정의하였다. (그림 6)은 망관리 도메인에서 식별된 컴포넌트들의 계층이며 현재 웹상에서 유통되는 컴포넌트를 검색하여 본 논문에서 제시한 아키텍처에 따라 재배치하면 (그림 7)과 같이 매핑될 수 있다.

Architecture Platform 계층의 컴포넌트들은 분산 컴퓨팅

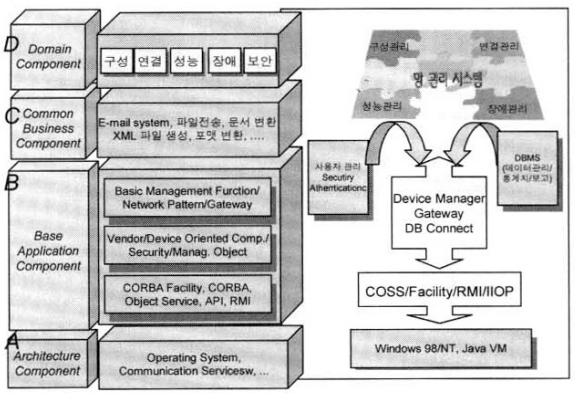
〈표 2〉 네트워크 도메인 패턴

도 메 인	사 용 예		패 턴 구 조 도
Session Management	Session Control & Observer	◦ 섹션 상태를 제어, 유지, 노티파이 ◦ N-ISDN과 B-ISDN과 같은 시그날링 프로토콜은 사용자 프로토콜(Cplane)로부터 제어 프로토콜(Uplanes)을 분해 ◦ C plane는 U plane 프로토콜의 특성과 교섭 ◦ 세 가지 클래스, SessionModel은 상태를 유지하고 SessionObserver에게 상태 변화를 알림. SessionControl은 Client입장에 Network -Control을 사용하여 네트워크 한정적인 제어프로시쥬어를 실행하고 SessionControl은 즉각적으로 SessionModel을 반영	
	Parties & Media as first class citizen	◦ Session Control & Observation 패턴 내에서 SessionModel의 역할의 주요 구조를 형성하기 위해 적용 ◦ 다중 parties와 연결을 수반하는 섹션을 구축하는 클라이언트는 복잡한 상태 정보를 유지해야 하는데 섹션이 진행 되어감에 따라 새로운 연결의 추가와 parties의 제거 등의 다양한 방법으로 수정 ◦ 각 세션은 수반되어지는 Parties와 Medium을 유지하는 Party Mediaum-Edge를 media와 연관되어지는 각 Party를 위한 상태를 유지하기 위해 사용하고 Parties와 Media는 섹션 내에서 first class citize	
Multimedia Realization	Layers	◦ Buschman에 의해 정의 ◦ 네트워크 프로토콜의 시스템의 분리를 요구 ◦ 프로토콜 컴포넌트가 어떻게 모델링 되는지를 해결하기 위해 시스템을 상위에서의 최상 수준의 프로토콜과 하위에서 최저 수준의 프로토콜을 가지는 적절한 개수의 프로토콜 계층으로 구조화	
Application Engineering	Pluggable Factory	◦ (C++)프레임워크의 재사용 가능한 확장성을 제공 ◦ 프레임워크는 실체화되어질 수 없는 기본 클래스들에 대해 알고 있음에도 불구하고 실체화 시키기 위해 Factory Method는 기본 클래스 객체를 반환하는 시그네처를 갖는 추상 인터페이스 상에 기초한 해결책을 제시 ◦ 사용자들은 반환되어지는 상세 클래스들의 인스턴스들의 상속을 통해 이들 인터페이스들의 구현을 커스터마이즈	
Network Management	CORBA to Management 패턴	◦ CORBA와 네트워크 관리 서비스 간의 매핑 관계 ◦ CORBA 객체 서비스와 CORBA Facility는 네트워크 관리 및 통신을 위한 기본 서비스를 제공 ◦ 중복되고 시스템 한정적인 서비스의 호환성과 추상화를 제공 ◦ CORBA 서비스는 게이트웨이로써 다양한 관리 시스템과 동적으로 매핑 가능	
	Manager-Server 패턴	◦ 웹 기반의 Java API를 통한 네트워크 관리 ◦ 클라이언트는 IIOP 프로토콜을 통해서 바로 관리 정보를 요청 ◦ Server는 ORB를 이용해 해당 관리 시스템의 서비스 결과를 Java 가상 기계를 통해 획득, 클라이언트는 이를 Applet의 형태로 다운로드 ◦ 웹 브라우저가 있는 어느 곳에서도 관리가 수행 가능 ◦ JMAPI의 지원으로 확장성 있는 시스템 운영 가능	
	COSS-Management Service 패턴	◦ COSS와 관리 서비스의 투명한 수행 ◦ CORBA를 통한 COSS와 CORBA 객체 지원 서비스들은 표준 네트워크 관리 서비스 수행의 추상성이 극대화된 네트워크 통신 프레임워크를 제공 ◦ 이질적인 네트워크 환경에서도 단일화 가능한 미들웨어 메카니즘을 제공	

환경에서 다양한 애플리케이션 구축을 위한 하부적인 물리적 플랫폼들로 구성되고, Base Application Component 계층의 컴포넌트들은 네트워킹을 위한 하부 컴포넌트로 통신과 데이터베이스를 위한 기본적인 서비스를 제공하기 위한 것들이다. Common Business Component 계층의 컴포넌트는 망관리 도메인 업무 수행을 위한 로직 프로세스로 Base Application Component 계층의 컴포넌트의 조합에 의해 형성되어 사용될 수 있다. Domain Component는 망관리영역별 컴포넌트들을 나타내고 이들 망관리를 위한 컴포넌트들은 하나의 도메인으로서 뿐만 아니라 분산 컴퓨팅을 위한 지원 컴포넌트로도 이용될 수 있다. 또한, 이 과정에서 식별된 컴포넌트는 다양한 의미로 다른 애플리케이션으로 조립될 수 있으며 상위의 더 큰 컴포넌트를 생성하기 위해 결합될 수도 있다. 이들 컴포넌트들은 실제적인 망관리 시스템 구축을 위해 계층별로 조립, 확장될 수 있다.



(그림 6) 망관리를 위한 컴포넌트 계층



(그림 7) 망관리 컴포넌트 계층화

3.4 망관리 컴포넌트 명세

컴포넌트는 단독으로 수행되는 작은 애플리케이션이라기보다는 조립되고 커스터마이징 되어짐으로써 비즈니스 로직을 수행하는 부품이다. 따라서 오직 정규 포맷을 따르는 인터페이스로 명세화된 컴포넌트만이 활용 가능하다. 현재 활용되고 있는 컴포넌트 명세는 인터페이스 서술을 간과한 채 개략적인 기능적 서술과 사용 환경과 같은 항목으로만 사용자에게 컴포넌트를 선택하도록 한다. 따라서 시스템으로의 통합이나 컴포넌트간의 조립을 위해 정확한 컴포넌트의 식별이 불가능하다. 그러므로 컴포넌트는 다양한 플러깅을 지원하고 분류된 컴포넌트 군으로 비즈니스 로직을 일관적으로 수행하기 위해 새롭게 정의된 명세 특성과 요소들을 가질 필요가 있다.

본 논문에서는 기존에 제시된 컴포넌트 명세 기법에 새롭게 요구되는 명세 특성들을 포함하여 새로운 명세 방법을 제시한다. ABCD 컴포넌트 아키텍처에 기반한 특징을 요약하면 <표 3>과 같다.

〈표 3〉 컴포넌트 명세 방법

① 컴포넌트 명세를 상세함의 수준에 따라 사용 명세와 개발 명세 두 가지로 정의
② 각 인터페이스 항목의 의미를 명확히 하기 위해 비정규적인 문서를 이용하여 명세 항목마다 '비고'란을 작성하고 선/후 조건을 통해 적용 범위를 명확히 정의
③ 컴포넌트 내 구성 요소들 간의 관련성 표현을 위해 순차도(Sequence Diagram)와 클래스도(Class Diagram)를 작성
④ 컴포넌트 사용의 신뢰성 보장을 위해 사용 시나리오를 작성하고 카테고리의 내에 시나리오 컴포넌트 설정
⑤ 컴포넌트 개발 환경과 컴포넌트 자체의 가변성/불변성(variant /invariant)을 명시

3.4.1 컴포넌트 명세 항목

① 컴포넌트 사용 명세

컴포넌트의 개요적인 이해와 비즈니스 카테고리 상에서의 위치 파악을 위한 사용 명세로서 본 논문에서는 <표 4>와 같이 개략 명세 항목을 정의했다.

〈표 4〉 컴포넌트 사용 명세 항목

항 목	의 미
Name	컴포넌트 식별 이름
Main Goal	(불변의)핵심적인 기능성
Description	전체적인 개요(기능, 요소, 기대 효과)
Families	컴포넌트의 비즈니스 카테고리
Related Component	상호 관계가 있는 컴포넌트
Provider	컴포넌트 제공자(업체)
Version	컴포넌트 버전
Environment	호환 가능한 플랫폼(시스템 환경)

② 컴포넌트 개발 명세

컴포넌트 개발을 위한 상세 명세로서 애플리케이션으로 전개할 때 조립하기 위한 명확한 의미적인 플러깅 지점을 확보하고 비즈니스 프로세스의 계층적 실현을 위해 <표 5>와 같은 명세 항목을 결정한다. 명세시 각 항목의 비고란을 통해 상세한 부가 설명이 포함된다. 컴포넌트 다이어

그램 항목에는 불변성(invariant)과 가변성(variant), 예외상황(exception)으로 인터페이스를 명확히 한다. 이용 시나리오 항목은 카테고리 내의 컴포넌트들의 이용 절차를 예시한 것으로 조립을 위한 명확한 가이드라인으로서 이용한다. 품질 요소(qualities Attribute)에는 컴포넌트의 성능과 보완 등의 플랫폼 관점에서 준수해야 할 요소들을 나열한다.

〈표 5〉 컴포넌트 개발 명세 항목

항 목	의 미
카테고리	비즈니스 도메인의 세부 기능 분류 따른 컴포넌트 군
컴포넌트 상관도	카테고리 내 컴포넌트들 간의 관계
컴포넌트 이름	비즈니스 관점의 서술적인 컴포넌트의 이름
분류 코드	ABCD 아키텍처에 기반한 컴포넌트의 계층적인 분류 코드
개요	컴포넌트 전반에 대한 기능, 동기, 동작 과정, 제약 사항
용어 사전	컴포넌트 명세에 사용된 용어의 의미 설명
Component Context Diagram	컴포넌트의 주요 기능을 서술
Component Interaction Diagram	컴포넌트 실행시 관련된 컴포넌트들 간의 관련성을 표현
Component Sequence Diagram	컴포넌트 자체의 행위적 순서 표현
Component Diagram	컴포넌트가 제공하고 제공받는 인터페이스 표현
Component Class Diagram	컴포넌트의 정적인 관점을 가시화하고 구현을 위한 속성과 오퍼레이션 및 상호 연관성 표현
Interface Spec.	공급하고 요구하는 인터페이스의 의미적서술, 선/후 조건, 입력/출력 결과 명시
Usage Scenario	컴포넌트 사용을 위한 인증된 시나리오
Quality Attribute	비기능적인 컴포넌트 (품질) 특성

3.4.2 망관리 컴포넌트 명세 사례

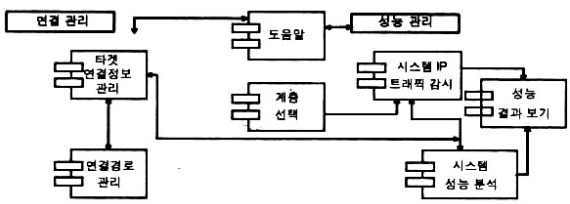
본 논문에 기술한 명세는 망관리 시스템 중에서 연결관리에 관한 컴포넌트 명세이다. 해당영역은 망관리를 위한 필수적인 영역으로 이 컴포넌트의 가용성과 적용 범위가 광대하므로 공통 컴포넌트로서 큰 가치를 가진다. 또한 기술된 명세는 컴포넌트 식별을 위한 도메인 분석에 의한 순공학적 접근을 시도하였다.

<표 6>은 망관리 도메인에서 식별된 연결관리 영역의 타겟 연결정보 관리의 명세이다.

〈표 6〉 타겟 연결 관리 컴포넌트 명세

(1) 망관리 컴포넌트(D-NM)	
High Component	Element Component
연결 관리 (01)	**타겟 연결정보 관리(01), 연결경로 관리 (02)**
구성 관리(02)	계층 뷰 선택(01), 토폴로지 뷰 관리(02), 노트 트리 뷰(03), 노드 정보 관리(04)
성능 관리(03)	시스템 IP 트래픽 감시(01), 시스템 성능 체크(02)
장애 관리(04)	알람 로그 관리(01), 알람 필터 정보 검색(02), 필터 정보 보기(03)

(2) 망관리 내 컴포넌트 상관도



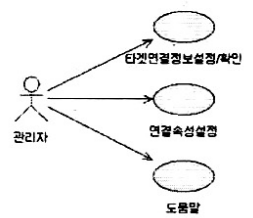
(3) 이름 : 타겟 연결정보 관리	(4) 분류 코드 : D-CM-01-0

(5) 개요 : 망의 연결 상태에 따른 활성화와 비활성화를 표현하는 기능으로 목적지 호스트의 주소를 입력하면 그 호스트까지의 지연시간, 손실률, 에러메시지를 결과로 표시한다. 또한 속성 설정을 통해 데이터의 크기를 바이트 단위로 설정하고, 타임아웃 시간과 ICMP Echo 메시지의 TTL 값을 설정할 수 있다.

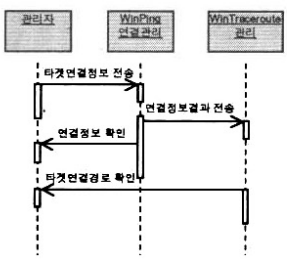
(6) 용어사전

- 연결관리 : 통신망의 연결, 변경, 해제에 대한 기능으로 점대다점 연결성 자원의 추가와 삭제, 그리고 망의 연결 상태에 따른 활성화와 비활성화를 표현하고 저장하는 기능
- 호스트 : 인터넷을 통해 다른 컴퓨터들과 양방향 통신이 가능한 컴퓨터
- ICMP : Internet Control Message Protocol
- TTL(time-to-live) : 패킷이 네트워크 내에 너무 오래 있어서 버려져야 하는지의 여부를 라우터에게 알림 기능

(7) Component Context Diagram

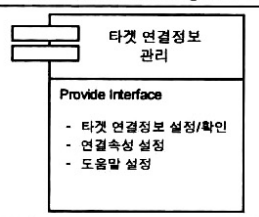


(8) Component Interaction Diagram

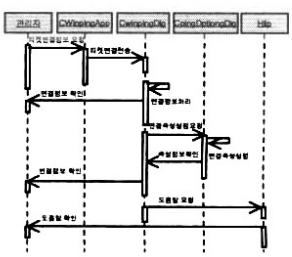


비고 : 연결관리는 IP 계층에서만 적용된다

(9) Component Diagram

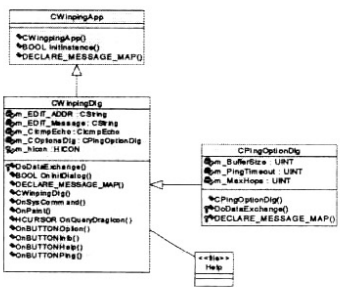


(10) Component Sequence Diagram

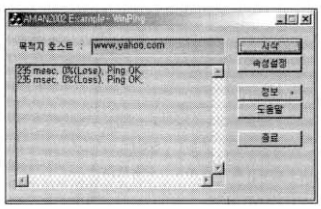


비고
Invariant : Ping Attribute
Variability : Attribute Value

(11) Component Class Diagram



(12) 인터페이스 명세
- 타겟 연결정보 설정/확인 Provide Interface타겟 연결정보 설정/확인 [to] 사용자 Description : 타겟 호스트의 연결 상태를 확인하기 위해 사용자에게 제공하는 interface이다. [Preconditions :] [Postconditions : 메시지 확인] Input : 타겟 호스트의 주소 입력 Output : 호스트까지의 지연시간, 손실률, 에러메시지출력. - 연결속성 설정 Provide Interface 연결속성 설정 [to] 사용자 Description : 타겟 호스트까지의 연결 속성 정보를 설정하거나 변경하는 interface이다. [Preconditions : 타켓 호스트와의 연결이 확립되어 있어야 한다.] [Postconditions : 연결 결과 메시지 확인] Input : 연결 속성 값 Output : 연결 속성값의 수정
(13) Usage Scenario
[필수 기능] 타겟 연결정보 설정/확인 - 사용자는 "목적지 호스트"에 목적지 주소를 넣고, "시작" 버튼을 누르면 목적지 호스트까지의 지연시간과 손실률, 에러메시지가 결과창에 보여진다. 연결속성 설정 - 일단 연결이 확립되면 연결 옵션으로 연결 "속성설정"에서 버퍼크기, 타임아웃, 최대 홉수를 설정할 수 있다. [부가적 기능]

(14) Quality Attribute	(15) Component 실행 화면
-비기능적 속성, 보완, 성능, 신뢰성에 대한 요소 Environment : Implementation Choice Type : Java Class Language : Java Tool : JDK 1.2 Container : 망관리 시스템	

3.5 망관리 컴포넌트 개발 방법

컴포넌트는 고수준의 재사용성과 생산성을 갖춘 비즈니스 컴포넌트들로 개발되고 상품화되어 개발자들이 선택할 수 있도록 제공되어야 한다. 이를 위해 비즈니스 영역의 전문가들은 가장 보편적인 비즈니스 모듈을 식별하여 컴포넌트화 한다. 이 과정은 객체지향 방법론에 기초한 순공학적인 컴포넌트 생성 방법이다.

기존의 방법론들은 분석, 설계의 많은 과정을 거친 후 애플리케이션 시스템을 대신할 컴포넌트를 식별, 추출, 생성한다. 즉, 컴포넌트를 생성하는 목적이 애플리케이션 시스템 개발을 대체할 부품화된 서브시스템을 제공하는 것이다. 따라서 특수한 비즈니스 요구나 제약 사항에 대해 적절한 해결책을 제시할 수는 있으나 많은 시간과 인력을 필요로 한다. 하지만, 본 논문에서는 컴포넌트의 공용 저장소를 통해 망관리 도메인 영역에 적용할 수 있는 제품화된 컴포넌트의 생성을 추구한다. 다시 말하면 동일하거나 유사 도메인의 애플리케이션에서 요구되는 공통 컴포넌트를 저장소를 통해 검색, 이해함으로써 개발 시스템에 적용하고자 하는 것이다.

3.5.1 컴포넌트 설계

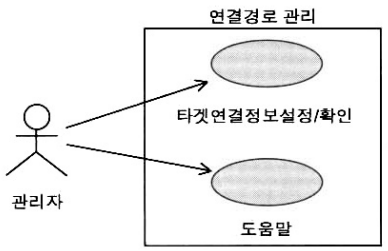
실제 컴포넌트를 개발하기 위한 설계는 UML에 기초하여 <표 7>의 4단계로 분류하여 수행한다. 본 논문에서는 망관리 영역 중 특히 연결관리 영역에서 연결경로 관리 컴포넌트를 분석, 설계하는 과정을 작성한다.

① 컴포넌트 컨텍스트 모델링

◦ 망관리 도메인을 분석하여 도메인에 존재하는 시스템 기능을 찾은 후 그들의 상호 연관 관계를 표현한다(그림 8).

〈표 7〉 컴포넌트 설계 단계

단 계	수 행 내 용
컴포넌트 컨텍스트 모델링	도메인 분석을 위한 모델링으로 시스템 기능 및 사용자와 시스템 간의 상호작용을 표현한다.
인터페이스 추출 모델링	컴포넌트들의 인터페이스를 식별하고, 그들의 상호작용을 통해 Provide/Require 인터페이스를 확인할 수 있다.
인터페이스 명세	물리적 단계의 컴포넌트를 결정함으로써 의미, 선/후 조건, 입출력 결과를 기술한다.
컴포넌트 모델링	컴포넌트의 구현을 위한 속성과 오퍼레이션 및 그들의 연관 관계를 표현하고, Provide/Require 인터페이스를 표현한다.



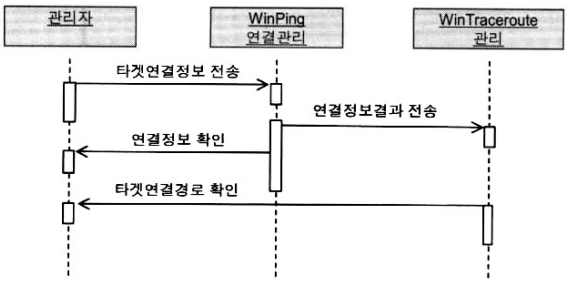
(그림 8) 컴포넌트 컨텍스트 모델

② 인터페이스 추출 모델링

◦ 망관리 컴포넌트들의 인터페이스를 식별하고 컴포넌트들의 상호작용을 표현한 것으로 연결관리 영역에서 제공하는 인터페이스로는 타겟 연결경로 설정/확인, 도움말 설정으로 정의되고, (그림 9)는 연결관리 영역의 컴포넌트가 제공하는 인터페이스와 그들의 상호작용을 표현한 것이다.

• 타겟 연결경로 설정/확인

연결관리 영역의 연결경로 관리 컴포넌트가 제공하는 인터페이스로 목적지 호스트에 목적지 주소를 설정하고 시작하면, 목적지 호스트까지 가기 위한 경로를 순서와 지연시간, IP 주소별로 나타낸다. 또한 모든 결과를 받을 때까지 실행 중단과 같은 명령을 받아들이지 않는다.



(그림 9) 인터페이스 추출 모델

③ 인터페이스 명세

◦ 연결경로 관리 컴포넌트의 서술과 선/후 조건 및 입출력 결과를 기술한다.

연결경로 관리 컴포넌트 인터페이스

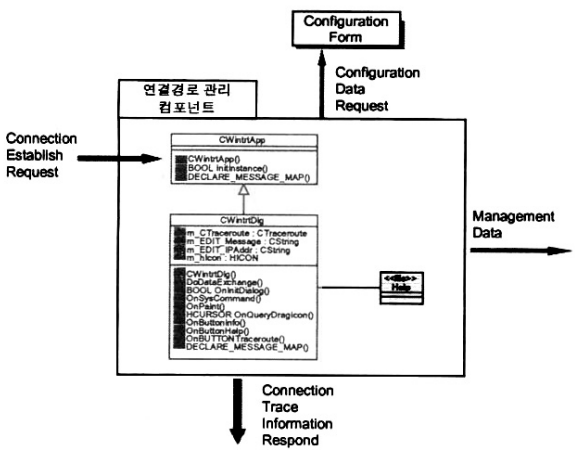
◦ 타겟 연결 경로 설정/확인
Provide Interface 타겟 연결 정보 설정 [to] 사용자
Description : 타겟 호스트의 연결 경로를 확인하기 위해 사용자에게 제공하는 interface이다.
[Preconditions : 타겟 호스트 연결 확립]
[Postconditions : 메시지 확인]
Input : 타겟 호스트의 주소 입력
Output : 호스트까지의 경로 순서와 지연시간, IP 주소를 출력

◦ 도움말 설정
Provide Interface 도움말 설정[to] 사용자
Description : 컴포넌트의 도움말 인터페이스이다.
[Preconditions : 컴포넌트가 실행 중이어야 한다.]
[Postconditions : 요청 도움말 메시지 확인]
Input : 도움말 요청 정보 입력
Output : 도움말 정보 출력

(그림 10) 인터페이스 명세

④ 컴포넌트 모델링

◦ 앞 단계를 통해 얻은 분석/설계를 이용하여 작성한 연결경로 관리 컴포넌트의 설계도를 (그림 11)과 같이 나타낸다.

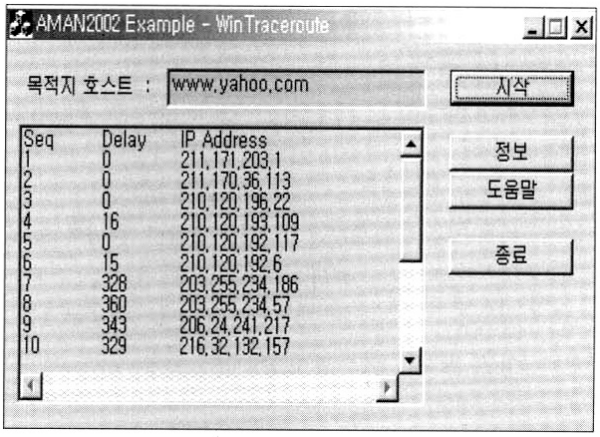


(그림 11) 컴포넌트 모델

3.5.2 컴포넌트 구현

본 논문에서는 현재 활발하게 진행되고 있는 CBD 기반 개발을 지원하기 위해 그 핵심 구성요소이자 선결 과제인 컴포넌트 구현을 위해 앞 단계의 설계 정보를 통해서 JAVA 언어를 기반으로 하여 연결관리 영역의 타겟 연결관리 컴포넌트와 연결경로관리 컴포넌트를 구현하였다.

(그림 12)는 연결경로 관리 컴포넌트로 목적지 호스트에 목적지 주소를 넣고 시작 버튼을 누르면 목적지 호스트까지 가기 위한 경로를 순서와 지연시간, IP 주소별로 창에 보여준다.



(그림 12) 연결경로 관리 컴포넌트 실행 예

4. 망관리 시스템 개발

4.1 망관리 요구사항 분석

오늘날 대부분의 비즈니스 영역에서 데이터 네트워크가 차지하는 비중이 증가하고 그 보급이 확대됨에 따라 망 문제로 인한 손실이 높아지고, 망을 사용한 사업의 증가에 따라 그 응용이 확대되어 가고 있다. 하지만 지금까지 대부분의 관리 시스템들은 관리의 특정 목적을 위해 특정 벤더나 프로토콜, 플랫폼에 의존하고 있으므로 관리자는 비일치적인 하부 구조를 충분히 이해해야 하며 망 운영의 상세한 부분까지 파악해야 한다. 이는 분산 관리되는 망 정보의 복잡성과 불확실성으로 인해 관리자는 정보 관리의 신뢰성을 보장할 수 없게 되며 유지보수 비용도 커지게 된다. 따라서 오늘날의 망관리 시스템은 클라이언트/서버 및 분산 처리 환경의 각종 망 장비를 비롯해 멀티 벤더, 멀티 시스템들을 포함하고 있는 망 전체를 통합, 관리함으로써 안정성과 신뢰성 및 생산성을 높이는 솔루션의 제공이 요구되어진다. 또한 망관리를 위한 도전으로 많은 독점적인 망, 공통적인 기초의 부재 그리고 일치적 관점의 소실, 망과 시스템 관리자들, 다른 프로토콜과 표준에 대한 이해가 요구되고 있다.

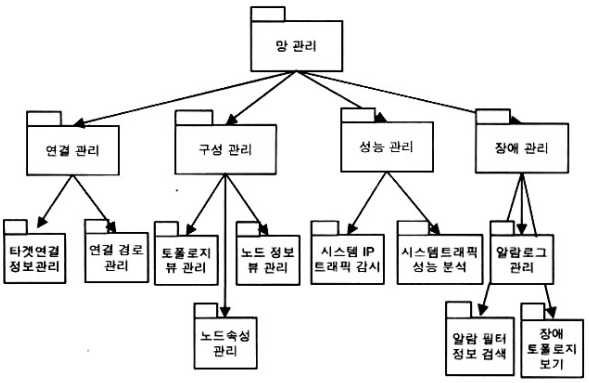
본 논문에서는 분산 처리 환경에서 네트워킹 기능을 구

현하는 것을 목표로 설계된 TINA를 기반으로 CORBA의 IIOP를 사용하여 분산 관리 객체간의 정보 전송을 제공한다. 이것은 또한 물리적 위치에 관계없이 쉽게 분산 구현할 수 있다는 장점이 있어 향후 통신망 망관리 기능 구현에 유용하게 사용될 것으로 기대하며, TINA의 관리 영역을 기반으로 망관리 시스템을 개발하고자 한다. TINA의 관리 영역에는 크게 구성, 연결, 장애, 성능, 보안, 과금 관리가 있는데 본 논문에서는 이 중에서 구성, 연결, 장애, 성능의 영역을 고려하여 시스템을 개발한다.

4.2 망관리 시스템 분석/설계

4.2.1 망관리 영역별 기능 구성도

망관리 시스템의 기능적인 관점들을 패키지 단위로 표현하면 (그림 13)과 같이 나타낼 수 있다. 각각의 패키지들은 독립적인 서브 모듈 형태의 컴포넌트로 개발 가능하며, 망관리 영역에서 연결, 구성, 성능, 장애 관리로 영역으로 구분된다. 연결관리 영역에서는 타겟 연결정보 관리와 연결경로 관리 컴포넌트를 포함하고, 구성관리 영역은 토폴로지 뷰 관리, 노드 정보 관리와 노드 속성 관리 컴포넌트을 제공한다. 성능 관리는 시스템 IP 트래픽 감시 컴포넌트와 트래픽 성능 분석을 위한 컴포넌트를 지원하고, 장애 관리 영역은 알람 로그 관리와 기존 알람 정보를 검색하기 위한 컴포넌트 및 알람 정보를 토폴로지 상에서 확인할 수 있는 컴포넌트들을 지원한다.



(그림 13) 망관리 시스템의 기능 구성도

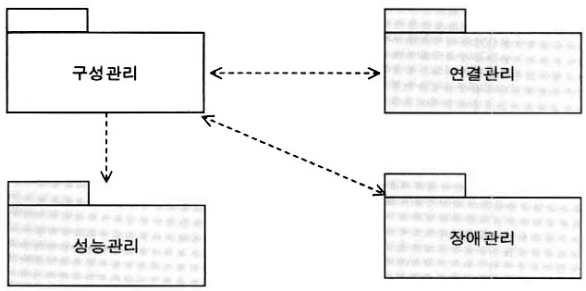
4.2.2 망관리 시스템 모델링

본 논문에서는 시스템을 시각화하고, 구축하는데 필요한 산출물들을 문서화하기 위한 모델링 언어로 UML을 사용하여 시스템과 사용자의 요구사항을 파악하고 이를 기반으로 사용자와 유스케이스를 추출하고 연관짓는 유스케이스 다이어그램과 각 유스케이스에 해당하는 시나리오를 중심으로 유스케이스 보고서를 작성하고 또한 객체와 그들의 메시지 흐름을 나타내기 위한 시퀀스 다이어그램을 기술하고 이들 정보를 바탕으로 클래스 다이어그램과 컴포넌트 다이어그램을 작성한다.

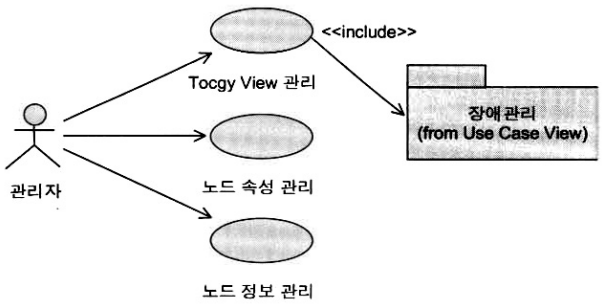
① 유스케이스 다이어그램과 유스케이스 보고서

유스케이스 다이어그램은 사용자 요구사항으로부터 시스템 영역과 기능의 정확한 파악을 목적으로 액터와 유스케이스 그리고 그들 간의 관련성 표현을 통해 구축할 시스템의 기능성과 범위를 한정짓고 각각의 유스케이스에 대한 이벤트 흐름을 나타내는 유스케이스 보고서로 완성된다. 다음의 유스케이스 다이어그램과 유스케이스 보고서는 망관리 시스템 개발을 위해 작성된 모델링 정보중 일부분이다.

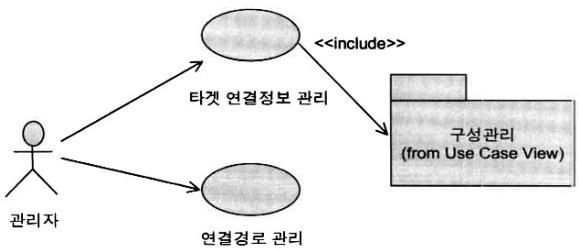
(그림 14)는 망관리 시스템의 유스케이스 패키지 다이어그램을 나타낸다. 각 패키지들은 망관리 영역별로 구분되고, 특히 컴포넌트의 재사용 측면에서 분석, 설계를 통해 컴포넌트의 식별 및 유용성에 대한 참조 정보로서 사용 가능하다. (그림 15), (그림 16), (그림 17)은 망관리 시스템의 일부를 표현한 것이다.



(그림 14) 망관리 시스템의 유스케이스 패키지 다이어그램



(그림 15) 구성관리 유스케이스 다이어그램



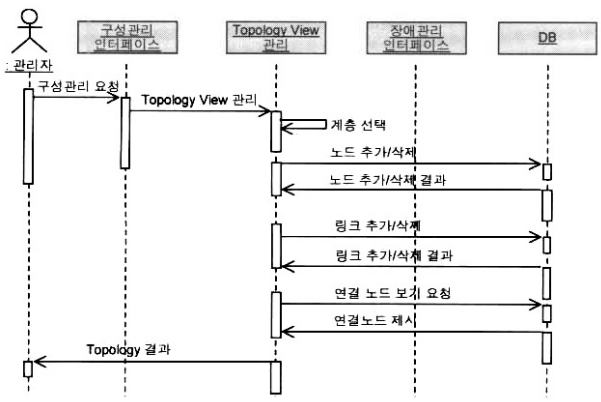
(그림 16) 연결관리 유스케이스 다이어그램

- 개략 서술
  - 이 시나리오는 관리자가 구성관리 영역의 전체 토폴로지 뷰를 봄으로써 시작된다.
    구성관리에서는 계층선택과 망의 토폴로지와 구성요소들의 정보를 추가, 갱신, 삭제할 수 있다.
  - 구성관리 영역에서 원하는 관리 영역(연결, 성능, 장애)을 선택함으로써 연결된다.
  - 연결관리는 타겟 노드에 대한 연결 정보와 연결 경로 순서에 관한 정보를 제공한다.
  - 성능관리는 망구성 자원들을 모니터하고 분석하기 위해 속성값을 설정하고 그래프 형태의 모니터링 결과를 얻을 수 있다.
  - 장애관리는 장애정보를 테이블 형태로 제공하고 장애정보 검색과 구성관리의 노드 색깔을 통해 장애 정도를 확인할 수 있다.
- 이벤트 기본 흐름
  - 망관리 시스템은 구성관리 영역을 기점으로 해서 다른 영역으로연결될 수 있다.
  - 구성관리의 토폴로지 뷰의 구성으로 시작한다.
- 이벤트 선택적 흐름
  - 계층선택 : Physical, DataLink, IP 계층 중 선택
  - 구성관리의 노드와 링크를 통해 토폴로지를 구성 및 정보를 추가, 갱신, 삭제
  - 연결관리의 특정 IP값 설정을 통해 연결정보를 확인
  - 성능관리의 특정 IP와 서비스 타입 값 설정을 통해 분석
  - 장애관리의 장애정보 리스트 확인
- 예외처리, 특별 요구 사항, 시작 조건, 종료 조건

(그림 17) 망관리 유스케이스 보고서

② 시퀀스 다이어그램

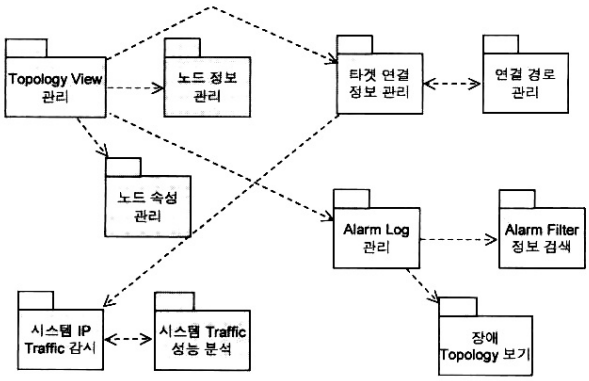
작성된 유스케이스 보고서의 시나리오 안에서 객체를 도출하고 객체와 객체들 사이의 관계 즉, 메시지 흐름을 분석할 수 있다. 시퀀스 다이어그램은 작성된 시나리오로부터 객체를 추출하고 이들 간의 메시지를 진행 순서에 따라 표현한 것이다. 다음은 망관리 시스템 개발을 위해 작성된 시퀀스 다이어그램의 일부로 (그림 18)은 관리자가 구성관리 인터페이스에서 망 구성 자원들의 토폴로지를 구성하기 위한 것이다. 토폴로지 뷰 관리에서는 관리자가 먼저 시스템이 제공하는 구성관리 인터페이스를 통해 토폴로지 뷰 화면에서 계층을 선택하고 노드와 링크들을 추가하고 노드들의 속성들을 설정하고 이들을 데이터베이스로 전달하며 처리된 결과들을 토폴로지 뷰 상에서 확인할 수 있다.



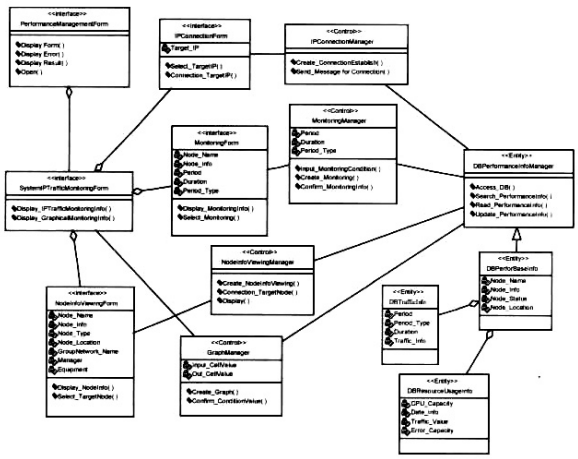
(그림 18) 구성관리의 토폴로지 뷰 관리를 위한 시퀀스 다이어그램

③ 클래스 다이어그램

앞 단계에서 분석된 망관리 시스템의 기능별 패키지 다이어그램과 그에 따른 유스케이스 다이어그램 및 보고서, 시퀀스 다이어그램을 통해 망관리 시스템의 클래스 다이어그램을 작성한다. (그림 19)는 망관리의 네 영역을 기반으로 10개의 패키지를 중심으로 전개해 나가며 논문에서는 성능관리 영역의 시스템 IP 트래픽 감시를 위한 클래스 다이어그램을 제시한다(그림 20).



(그림 19) 망관리 패키지 다이어그램



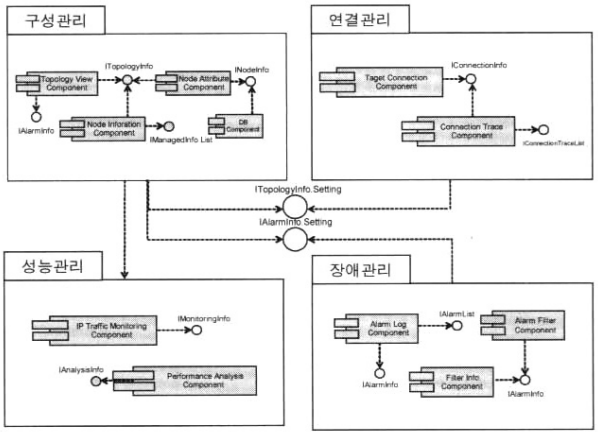
(그림 20) 성능관리의 시스템 IP 트래픽 감시를 위한 클래스 다이어그램

④ 컴포넌트 다이어그램

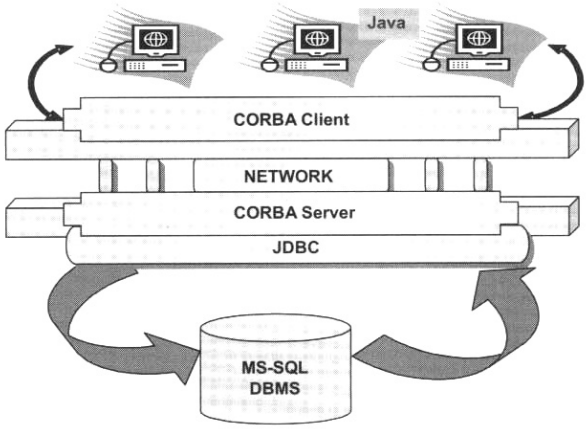
컴포넌트 다이어그램은 클래스들의 집합으로 물리적 구성단위인 컴포넌트와 그들간의 구성 및 의존 관계를 표현한 것으로 시스템의 정적인 구현 뷰를 나타낸 것이다. 인터페이스는 오퍼레이션의 집합체로 클래스와 컴포넌트의 서비스를 명세화하고 컴포넌트를 함께 묶는 수단으로 인터페이스를 사용한다. 또한 인터페이스를 통해 서비스를 호출하는 다른 컴포넌트들과 함께 그 인터페이스를 실현하는 컴포넌트들을 구성한다. (그림 21)은 망관리 시스템의 컴포넌트 다이어그램을 나타낸 것이다.

4.3 망관리 시스템 구현

앞 단계에서 분석, 설계된 정보들을 기반으로 실제 환경에서 동작할 수 있도록 최상의 구현 기술을 적용하여 망관리 시스템을 구현하였다.



(그림 21) 망관리 시스템을 위한 컴포넌트 다이어그램



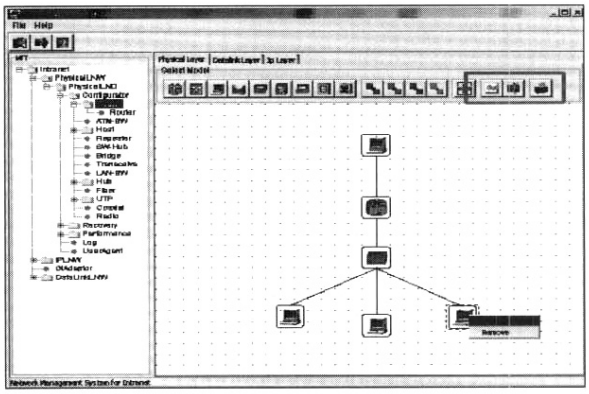
(그림 22) 망관리 시스템 개발을 위한 구현 아키텍처

망관리 시스템은 TINA가 정의하는 4가지 관리 영역으로 구분하여 제공하였고, 망관리를 위한 사용자 인터페이스는 JAVA를 사용하여 구현하였으며, CORBA를 이용하여 분산된 클라이언트들이 이질적인 MIB 서버와 투명하게 정보를 액세스 할 수 있다. 즉, 망관리 시스템의 구현 환경은 다음과 같이 요약할 수 있으며, (그림 22)는 환경적 구성을 도식화 한 것이다.

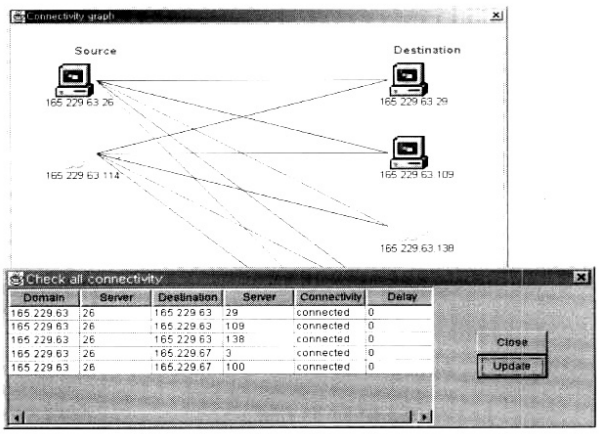
4.3.1 망관리 시스템 구현 예

(그림 23)은 구성관리 인터페이스로 통신망관리를 원할하게 수행하는데 필요한 데이터를 수집하고 제공하는 기능 및 계층별 토폴로지 정보를 그래픽적으로 표현한 것으로, 노드정보 트리(MIT)와 각 계층별 뷰(Physical, DataLink, IP), 토폴로지 뷰, 노드속성 관리로 구성되어 있다.

연결관리는 연결, 변경, 해제에 대한 기능으로 점대다점 연결성 자원의 추가와 삭제, 그리고 망의 연결 상태에 따른 활성화와 비활성화를 표현하는 기능을 테이블과 그래픽 형태로 제공하고, 또한 이들 정보를 저장하고 로드할 수 있는 기능을 제공한다. 연결관리는 IP 계층에서만 적용된다. 구성은 연결 상태를 체크하기 위한 연결성 체크와 연결을 보여주고 구성 정보를 관리할 수 있는 파일 기능을 제공한다 (그림 24).



(그림 23) 구성 관리 화면

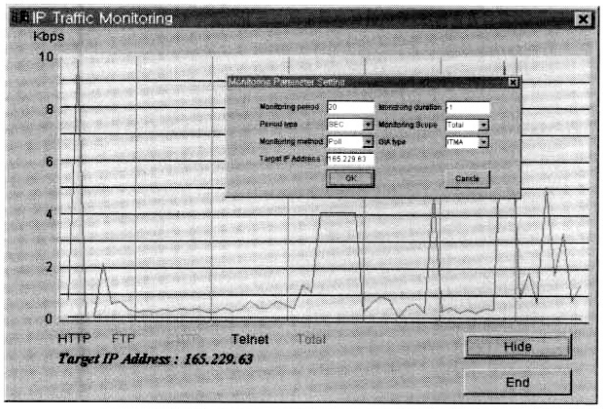


(그림 24) 연결 관리 화면

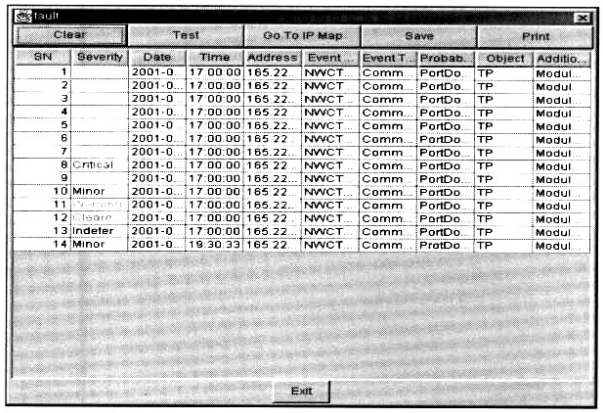
성능관리는 망을 구성하는 자원들의 성능평가를 위해 각 자원들에 대한 통계적 자료를 수집, 감시하고 시스템의 상태를 기록, 저장, 유지하며 상태에 대한 분석을 통해 망의 효율성과 작동상태에 대한 교정을 제공한다. (그림 25)는 감시, 분석, 보정을 수행하는 기능과 그 결과를 표시하는 화면으로 구성된다.

장애관리는 지속적인 장애 발생 여부를 감시함으로써 장애가 발생했을 경우 장애 위치를 파악하여 즉시 복구할 수 있는 기능을 제공하고, 장애복구 후에도 장애 구간의 정상적인 동작 여부를 확인할 수 있어야 한다. 또한 망의 고장 원인과 특징, 처방 방법에 관한 정보를 제공하는 기능을 포

함한다. 시스템은 장애 정보와 발생의 심각도를 테이블에서 다양한 색깔로 표시하고(그림 26), 장애가 발생하면 구성관리 맵에서 장애의 심각도에 따라 색깔별로 구분하여 그래픽적으로 표현한다.



(그림 25) 성능관리 화면



(그림 26) 장애 관리 화면

5. 결 론

컴포넌트 기반 개발 방법은 소프트웨어 개발을 위한 최상의 경쟁 전략으로 확산되면서 기성품 형태의 컴포넌트를 생산, 선택, 평가, 조립에 의한 애플리케이션 개발의 생산성을 높이고 유지 보수성을 보장한다.

또한 웹을 통한 컴포넌트 보급과 대중화에 따른 비즈니스 측면의 경제성 보장 및 다양한 요구사항과 이종 시스템 간의 상호호환 문제에 대한 해결책으로 제시될 수 있다. 망관리 도메인은 웹, JAVA, CORBA 등이 애플리케이션의 표준 아키텍처 플랫폼으로 자리 잡아감에 따라 분산 컴퓨팅의 기본적인 지원 서비스로 뿐만 아니라 멀티 벤더/솔루션의 표준 모델로서 그 역할이 중요하다. 즉, 하부적인 공통 기능 제공의 표준 모델과 수직적인 응용 영역으로 컴포넌트와 설계 패턴을 통한 표준 아키텍처 정의와 적용 방법에 관한 연구가 필요하다. 본 논문에서는 TINA를 기반으로 한 망관리 시스템을 개발하기 위해 시스템 아키텍처를 정의하고, 망관리 설계패턴과 컴포넌트들을 식별, 정의하여 컴포넌트 아키텍처에 매핑함으로써 CBD를 위한 공통적인 재사용 자원을 확보할 수 있었다. 또한 망관리 영역별 분석/설계를 통해 컴포넌트 개발과 유통, 사용을 위한 컴포넌트를 명세하고, 컴포넌트 설계를 통해 구현하였다. 개발된 컴포넌트들은 컴포넌트 저장소를 통해 사용자가 소프트웨어를 개발할 때 사용자가 요구하는 소프트웨어 목적과 컴포넌트 개발자의 개발 의도를 비교, 식별하여 적절한 컴포넌트를 시스템에 적용하여 개발함으로써 애플리케이션 개발과 운영을 위한 생산성을 향상 시킬 수 있었다. 그러므로 이러한 접근 방법들은 망관리를 위한 표준화된 계층을 확립하여 인터네트워킹 환경의 효율적인 관리 시스템 개발 모델을 제시함으로써 애플리케이션의 개발과 운영을 위한 품질과 생산성을 향상시킬 뿐만 아니라 망관리에서 사용되는 기존의 분산 객체 서비스들을 컴포넌트로 구성하여 실시간적인 변화에 민감하고, 다양한 사용자의 요구를 반영시킬 수 있었다.

망관리 시스템 개발에 CBD를 적용하여 얻을 수 있는 기대 효과로는 첫째, 컴포넌트 재사용을 통하여 망관리 시스템 개발 기간과 비용을 줄일 수 있고, 검증된 컴포넌트를 사용함으로써 소프트웨어 품질을 보장할 수 있다는 것이다. 둘째로 망관리 시스템 개발을 위한 표준 기술을 정립하는 계기가 될 수 있다. 컴포넌트 기반으로 시스템 모델을 정의하고, 컴포넌트를 설계, 구현하는 체계적인 방법과 컴포넌트 사용 및 유통을 위한 표준화된 명세 방법 확립을 위한 기술 향상을 기대할 수 있다. 세째, 컴포넌트 저장소 개발을 통한 컴포넌트의 보급과 획득의 대중화에 따른 경제성 향상을 제공하고, 새로운 애플리케이션으로의 적용을 통해 컴포넌트에 의한 재사용을 설명할 수 있는 통합된 접근 방식을 정립하고 실질적인 적용 기술을 획득할 수 있다. 마지막으로, IT나 EC와 같은 다양한 응용 영역을 위한 기반 기술로써 제공될 수 있다.

향후 연구로는 식별된 망관리 컴포넌트의 재사용성 및 조립, 유통의 표준화와 기존의 망관리 애플리케이션으로부터 컴포넌트로 변환하는 방법에 관한 연구가 필요하다.

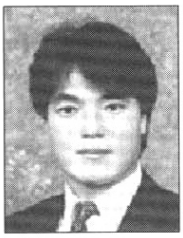
참 고 문 헌

[1] Douglas W. Stevenson, "Network Management : What it is and What it isn't," Available web server from http://netman.cit,buffalo.edu/Doc/Dstevevson, 1996.

[2] F. Dupuy, "The TINA-C : Towards Networking Telecommunications Information Services," ISS 1995 Contribution,

June, 1994.
[3] Brown A. W. and Wallnau K. C., "The Current State of CBSE," IEEE Software, pp.37-46, September/October, 1998.
[4] Voas, J. M., "The Challenges Of Using COTS Software In Component-Based Development," IEEE Computer, Vol.31, Issue 6, pp.44-45, 1998.
[5] G. Nilsson, "An Overview of the TINA," TINA Conference Proceeding, pp.1-12, February, 1995.
[6] J. Bloem, "TINA-C Connection Management Components," TINA Conference Proceeding, pp.485-494, February, 1995.
[7] Dr. Sidnie Feit, SNMP : A GUIDE TO NETWORK MANAGEMENT, McGraw-Hill, 1995.
[8] William Stallings, Networking Standards : A Guide to OSI, ISDN, LAN, and MAN Standards, Addison-Wesley, 1993.
[9] Raman, L., "OSI systems and network management," Communications Magazine, IEEE, Vol.36, Issue 3, pp.46-53, 1998.
[10] E. Gamma, R. Helm, R. Johnson and J. Vissides, Design Pattern : Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.
[11] 김정아 역, GOF의 디자인 패턴, Addison-Wesley, 2002.
[12] 김행곤, 김지영, "설계 패턴 재사용 라이브러리 구현", 한국정보과학회, 제7권 제1호, pp.48-62, 2001.
[13] Wilkes and Lawrence, Understanding Component Based Development, Addition-Wesley, 2000.
[14] Alan W. Brown, Large-Scale Component-Based Development, Prentice Hall PTR, 2000.
[15] J. Han, "An Approach to Software Component Specification," Proceedings of 1999 International Workshop on CBSE, Los Angeles, at URL : http://wwww.sei.cmu.edu/cbs/icse99/cbsewkshp.html, 1999.
[16] 김행곤, 차정은, 김지영, 신호준, "컴포넌트 저장소 형상 관리 시스템 개발", ETRI 최종보고서, 2000.
[17] Chauki Jeong, Shahsavari, M. M., "Component-based distributed network management," Southeastcon 2000, Proceedings of the IEEE, pp.460-466, 2000.
[18] 김행곤 외, "망관리 사용자 인터페이스 시스템 구축을 위한 컴포넌트 명세 및 프로토타이핑", 제5회 산학연 소프트웨어 공학 기술대회, pp.173-177, 2001.
[19] 김지영, 김행곤, "컴포넌트 기반 망관리 시스템 개발", 한국정보과학회 소프트웨어공학연구회, 제5권 제1호, pp.191-201, 2003.
[20] Grady Booch, Visual Modeling with Rational Rose 2000 and UML, Addition-Wesley, 2001.
[21] Robert Orfall, Client/Server Programming with Java and CORBA, John Willy & Sons, Inc., 1998.
[22] Ivor Horton, Beginning Java 2, 정보문화사, 2001.

김 행 곤



e-mail : hangkon@cu.ac.kr
1985년 중앙대학교 전자계산학과(공학사)
1987년 중앙대학교 대학원 전자계산학과 (공학석사)
1991년 중앙대학교 대학원 전자계산학과 (공학박사)
1978년~1979년 미 항공우주국 객원 연구원
1987년~1989년 한국전기통신공사 전임연구원
1988년~1989년 AT&T 객원 연구원
1990년~2000년 대구효성가톨릭대학교 컴퓨터공학과 부교수
2001년~2002년 Central Michigan University 교환교수
2000년~현재 대구가톨릭대학교 컴퓨터공학과 교수
관심분야 : 컴포넌트기반 소프트웨어공학, 임베디드 소프트웨어 개발 방법론 및 툴, 프로덕트라인 공학

김 지 영

e-mail : kimjy@cu.ac.kr
1997년 대구효성가톨릭대학교 전자계산학과 (이학사)
2000년 대구가톨릭대학교 전산통계학과 (이학석사)
2004년 대구가톨릭대학교 전산통계학과 (이학박사)
관심분야 : 객체지향 소프트웨어공학, 재사용, 컴포넌트기술, 프로덕트라인 공학