

카타르시스 방법론에 기반한 네트워크 관리 컴퍼넌트 개발

김 행곤, 차 정은, 김 지영, 박 은주¹
대구효성가톨릭대학교 컴퓨터공학과
{hangkon, jecha, kimijy, ejpark}@cuth.cataegu.ac.kr

Development of Component for Network Management Based on Catalysis

Haeng-Kon Kim, Jung-Eun Cha, Ji-Young Kim, Eun-Ju Park
Dept. of Computer Engineering, Catholic University of Taegu Hyosung

요 약

CBD(Component Based Development)는 미리 만들어진 검증된 부품을 합성, 조립하는 작업으로, 최근에 빠른 속도로 생겨난 카타르시스는 객체와 프레임워크로 CBD를 위한 완벽한 지원을 제공하는 차세대 방법론이다. 네트워크 환경과 인터넷의 보급으로 인한 인프라 구축과 다양한 방법론의 개발은 소프트웨어 부품화로서의 컴퍼넌트 기반 소프트웨어 개발에 대한 관심을 높이고 제반 여건을 조성하는 계기가 되었다. 또한 네트워크 기반의 분산 환경이 응용의 개발과 활용을 위한 표준 아키텍처로 자리잡아 가면서 네트워크 관리가 개별 응용으로서 뿐 아니라 대부분의 응용에서 하부 계층 지원 서비스로 요구되는 중요한 응용 도메인이 된다. 따라서 다수의 개발자들이 환경적 제약없이 네트워크 관리를 위한 응용 컴퍼넌트를 획득, 조립함으로써 요구되는 서비스를 구현할 수 있다. 그러므로 본 논문에서는 컴포넌트 소프트웨어 설계를 위한 방법론인 카타르시스를 사용하여 네트워크 관리 시스템(NMS)을 위한 컴포넌트를 설계하고 필요한 컴포넌트를 식별, 개발한다. 이를 통해 NMS를 위한 공용의 컴퍼넌트를 확보함으로써 체계적인 CBD 방법론에 의한 NMS 개발로의 표준화된 접근할 수 있다.

1. 서론

컴퍼넌트 기반 개발(CBD)은 컴포넌트 구축, 선택, 조립으로 구성되는 소프트웨어 개발 방법으로 개발자에게 잘 정의된 인터페이스만 제공하여 블랙 박스 형태의 조립을 통해 쉽고, 빠르게 응용을 개발할 수 있게 한다. 또한 컴포넌트 재사용은 소프트웨어 개발에서 구성 요소의 모듈화를 촉진시켜 산업화와 대량화를 위한 재사용 요소로 실현할 수 있다. 객체 프레임워크로부터의 차세대 CBD 방법론이라 할 수 있는 카타르시스는 모델과 복잡한 시스템을 서술하는데 있어 사용자의 응집된 뷰의 조립을 위한 강력한 메카니즘 상에서 잘 정의된 일관성 규칙을 제공한다. 최근 분산 컴퓨팅 환경에 대한 통합된 네트워크 관리에 관심이 높아지면서 웹, Java, CORBA가 응용의 표준 아키텍처로 자리잡고 있다. 네트워크 관리의 상호 연결된 네트워크의 서비스 사용을 위해 자원들을 관리, 제어하기 위한 것으로 개별 응용으로써 뿐 아니라 대부분의 응용에서 하부 계층 지원 서비스로 요구된다. 그러므로 네트워크 관리 도메인의 컴포넌트를 식별하고 활용함으로써 응용의 개발 측면에서 생산성을 향상시킬 수 있다. 따라서 본 논문에서는 네트워크 관리의 도메인 분석을 통해 CBD를 위한 완벽한 지원을 제공하는 차세대 방법론인 카타르시스를 사용하여 네트워크 관리

시스템(NMS)을 위한 컴포넌트를 설계하고 필요한 컴포넌트를 식별, 분류함으로써 공용의 컴포넌트를 확보함으로써 체계적인 CBD 방법론에 의한 NMS 개발로의 표준화된 접근할 수 있다.

2. 관련 연구

2.1 네트워크 관리 시스템

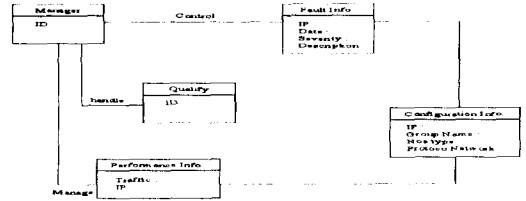
네트워크 관리 시스템은 네트워크를 관리, 제어하기 위한 도구들의 집합으로서 웹과 같은 간단한 인터페이스를 통해 모든 관리 기능을 수행해야 하며 기존 구성 자원들과 호환성 있게 동작해야 한다. 관리 체크 리스트로서의 OSI 관리 아키텍처 모델은 <표 1>과 같은 5개의 기능적 관리 영역에 따른 네트워크 관리 정보를 제시한다[1].

2.2 카타르시스 방법론

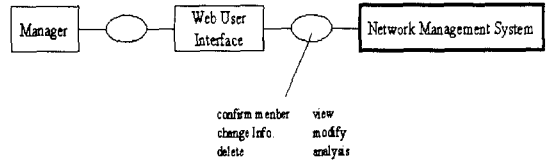
카타르시스는 CBD를 위한 완벽한 지원을 제공하는 차세대 방법론으로써 다음과 같은 몇 가지 특징을 가진다. 첫째, 카타르시스는 프레임워크를 사용한 컴퍼넌트 기반 개발이다. 둘째, 많은 문맥에서 모듈성과 융통성이 있고, 비즈니스 모델에서부터 코드까지 정제에 의한 추적 개발, 그리고 의미의 일관성 체크와 같이 명확한 개발 프로세스이다. 셋째, 카타르시스는 모든 레벨에서 프래

<표 1> 네트워크 관리 정보

OSI 영역	기능	관리 정보
Fault Management	모든 네트워크 상태를 관리, 추종 및 해결하고 보고하는 기능	모든 네트워크 활동 정보, 모니터링 정보(네트워크 정보 변화와 엑스트라), 고장 원인, 원인별 특징, 처방 방법에 관한 정보, 복구 예상할 방법 및 길의 정보, 고장 원인의 연속적 정보
Configuration Management	네트워크 장치 하드웨어, 소프트웨어 지원들에 대한 모든 정보를 수집하고 유지관리하며, 데이터베이스가 되어 네트워크 운영의 계획, 제어, 변경을 하는 기능	네트워크 구성 관리 정보, 네트워크 객체 위치 정보, 네트워크 서비스 리스트 정보, 노드 관련 설정 정보, 특징 코드 정보, 네트워크 연결 정보, 네트워크 특성 관리 정보(미래에 설정, 로딩, 디버깅, 네트워크 보물고시 상태 관리 정보)
Accounting Management	네트워크 사용자, 사용자 이용을 분석하여 이용자에게 비용을 부과하고, 특정 네트워크 사용자 행위를 제어	자원의 배정 할에 정보, 사용자들의 변경 정보, 네트워크 구성 요소의 사용 기록 정보, 비용 변화 정보, 오류 예측 정보, 통계 보고서 작성, 네트워크 상태 예상 정보(장기 구배 정보, 주기 정보), 도입 부하 정보
Performance Management	전체 네트워크 구성 장비에 대한 용량 정보를 수집, 관리하며, 최적의 성능을 유지하는 데 관리	네트워크 구성 정보, 용량 시간 정보, 현재 성능 정보, 현재 및 과거 성능 정보, 네트워크 초기화 정보, 관련 데이터 분석 정보, 네트워크 사용자에 정보
Security Management	불법적인 데이터, 유출을 예방하고, 적의 비밀 정보를 누출을 최소화하여 정보 변경을 예방	접근 정보의 사별할, 암호 정보, 세션식 정보, 사용자 접근 제어 정보



(그림 1) NMS의 Business Model



(그림 2) NMS의 System Context Model

입워크를 함성, 추적, 재사용가능한 모델링을 위한 지원 도구의 새로운 레벨을 부여한다. 넷째, 많은 다른 개발 방법론을 지원한다.

UML 표기법을 사용하여 시스템 설계자나 제작자가 쉽고 빠르게 소프트웨어에 대한 설계를 진행할 수 있도록 한다. 카타르시스 방법론에서 컴퍼넌트를 설계하는 과정은 크게 다음 5가지로 나뉜다[2,3].

(1) Domain Modeling

컴퍼넌트의 도메인을 분석하고 그것을 모델링한다.

(2) System Context Modeling

인터페이스를 통해 시스템의 기능을 표현하고, 비즈니스 타입과 시스템 인터페이스의 상호작용을 표현한다

(3) Type Interface Definition

오퍼레이션과 타입의 관계 및 오퍼레이션의 시그내처, Pre/Post Condition의 설정을 통하여 오퍼레이션의 명세를 확정한다.

(4) Component Interface Specification

실제의 물리적 단계의 컴퍼넌트를 결정한다. 도메인의 자원, 네트워크 아키텍처, 로드 밸런싱, 보안등을 고려하여 컴퍼넌트의 단위를 결정한다.

(5) Application User Interface Design

설계된 컴퍼넌트 사용을 시험하는 단계로 컴퍼넌트 구현 단계 이후의 적용 작업이다. 여기서는 UI 설계, 소프트웨어 프로토타이핑, 유용성 테스트 등이 수행된다.

3. NMS 컴퍼넌트 설계

3.1 문제 명세

네트워크 관리자는 웹 상의 사용자 인터페이스를 통해서 관리를 하게된다. 관리자는 관리 시스템을 사용하기 전에 거쳐서 네트워크 정보를 감시하고, 구성 정보를 등록, 수정, 삭제와 성능을 제어한다.

3.2 컴퍼넌트 설계

(1) Domain Modeling

: 네트워크 관리 도메인을 분석하여 도메인상에 존재하는 비즈니스 타입을 찾은 후 그 속성과 연관관계를 표시하는 비즈니스 타입 모델을 작성한다(그림 1).

(2) System Context Modeling

: 네트워크 관리 시스템의 인터페이스를 통해 시스템의 기능을 표시한다(그림 2).

(3) Type Interface Definition

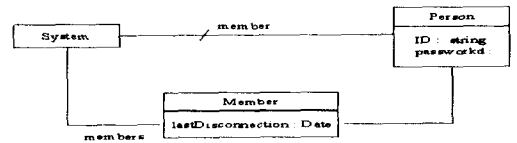
: 인터페이스의 자료형과 객체를 정의하는 단계로서, 네트워크 관리 시스템에 대한 Type Model은 member management, monitor, manage, control로 정의되고(그림 3, 4, 5, 6), 이들을 통합한 모델은 (그림 7)과 같다. 아래는 각 Type Model에 대한 서술이다.

● member management

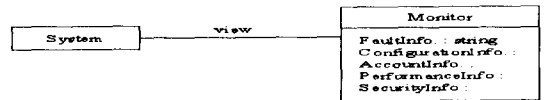
네트워크 관리 시스템은 관리자의 인증여부를 변경된 정보를 위하여 기존 정보를 삭제, 수정한 마지막 연결종료 시간에서부터 한정기간이 지나면 자에 대한 권한을 취소한다.

● monitor

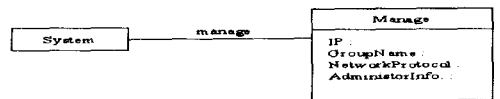
네트워크 관리 시스템에서 관리되어지는 모든 관 감시하는 기능으로, 관리되어지는 정보에는 Configuration, Accounting, Performance, Security에 대한 정보가 있다.



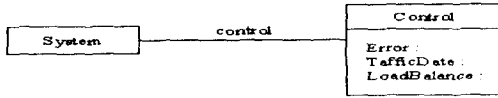
(그림 3) Type Model : member management



(그림 4) Type Model : monitor



(그림 5) Type Model : manage



(그림 6) Type Model : control

●manage

네트워크 정보를 관리하기 위하여 새로운 관리 기록하고 불필요한 정보의 삭제, 그리고 변경된 정보를 관리한다.

●control

관리 시스템에서 에러가 발생하는 경우, 관리자에게 이를 알리고 발생한 에러를 분석한다. 그리고 나서 이에 대한 적절한 해결책을 판단하여 관리자에게 제

(4) Component Interface Specification

NMS-manage Component Interface Specification

- 구성 정보 등록**

Provide Interface "구성 정보 등록" [to] "사용자"

Description: 네트워크 관리를 위한 사용자의 정보를 등록하는 interface이다.

[Preconditions: 등록 정보를 저장할 MIB가 구축되어 있어야 한다.]

[Postconditions: 등록된 정보가 MIB에 저장되어야 한다.]

Input: 사용자 구성 정보

Output: 사용자 구성 data
- 구성 정보 수정**

Provide Interface "구성 정보 수정" [to] "사용자"

Description: 등록되어 있는 사용자의 구성 정보를 수정하는 interface이다.

[Preconditions: 수정하려는 구성정보는 반드시 MIB에 등록되어 있어야 한다.]

[Postconditions: 수정된 구성정보는 MIB에 저장되어야 한다.]

Input: 등록되어 있는 사용자의 IP

Output: 변경된 사용자 data
- 구성 정보 삭제**

Provide Interface "구성 정보 삭제" [to] "사용자"

Description: 등록되어 있는 사용자의 구성 정보를 삭제하는 interface이다.

[Preconditions: 삭제하려는 구성정보는 반드시 MIB에 등록되어 있어야 한다.]

[Postconditions: MIB에서 삭제하려는 data가 삭제되어야 한다.]

Input: 삭제하려는 사용자의 IP

Output: 변경된 사용자 data
- 구성 정보 저장**

Require Interface "구성 정보 저장" [to] "MIB"

Description: 구성정보의 등록, 수정, 삭제 data를 위한 interface이다.

[Preconditions: MIB가 구축되어 있어야 한다.]

[Postconditions: data의 operation이 이루어져야 한다.]

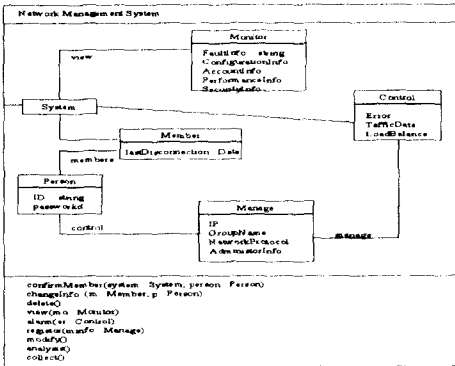
Input: data name

Output: data name

비고

- 구성 정보: 네트워크 관리 시스템에서 관리되어 있는 configuration에 관한 정보로서, IP, GroupName, NetworkProtocol, AdminInfo등 이 있다.

(5) Application User Interface Design



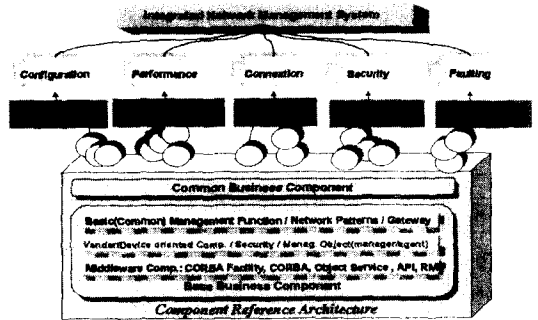
(그림 7) 통합된 NMS Type Model

4. 네트워크 관리 컴퍼넌트 식별

3장에서 설명한 카타르시스 방법론에 따라 네트워크 관리 시스템을 위한 컴퍼넌트는 <표 2>와 같이 식별될

<표 2> 식별된 네트워크 관리 컴퍼넌트

Component	
Configuration	설치, 상태제어, 정보DB, 장비 제공
Performance	모니터링, 시뮬레이션, 정보수집/분석
Connection	연결섹션 제어, 연결설정/해제
Security	암호, 접근 레벨링, 정보변경제어
Faulting	알람, 테스트, 진단 보고, 오류 교정



(그림 8) 네트워크 관리 도메인에서 식별된 컴퍼넌트 계층

수 있다. 이들 컴퍼넌트들은 분산 컴퓨팅을 위한 공통의 하부 컴퍼넌트 혹은 NM 도메인만을 위한 컴퍼넌트로 분류될 수 있다. 이들 컴퍼넌트들은 (그림 8)과 같은 NM 계층을 이룰 수 있다[4].

5. 결론 및 향후연구

최근 CBD와 네트워크 기반의 분산 환경이 응용의 개발과 활용을 위한 표준 아키텍처로 자리잡아 가면서 네트워크 관리가 중요한 응용 도메인이 된다. 따라서 다수의 개발자들이 환경적 제약없이 네트워크 관리를 위한 응용 컴포넌트를 획득, 조합함으로써 요구되는 서비스를 구현할 수 있다. 본 논문에서는 컴포넌트 소프트웨어 설계를 위한 방법론인 카타르시스를 사용하여 네트워크 관리 시스템(NMS)을 위한 컴포넌트를 설계하고 필요한 컴포넌트를 식별, 분류함으로써 공用的 컴퍼넌트를 확보할 수 있었다. 식별된 컴퍼넌트들은 체계적인 CBD 방법론에 의한 NMS 응용 개발을 위한 표준화된 접근 방식을 제공할 수 있을 것이다. 향후 의한 표준화된 컴퍼넌트 표현 정보를 이용하여 컴퍼넌트 재사용 저장소를 구축하고 이를 효율적으로 검색할 수 있는 방법론이 제시되어야 할 것이다.

6. 참고 문헌

[1] William Stallings, Networking Standards : A Guide to OSI, ISDN, LAN, and WAN Standards, Addison-Wesley, 1993

[2] D'Souza, Wills, Objects, Components, and Frameworks with UML, Addison Wesley, 1998

[3] 최중윤 외, "컴퍼넌트 인터페이스 설계와 객체지향 분석설계방안", KIPS 소프트웨어공학연구회지, 제 2권 1호, 1999

[4] 김해곤 외, "네트워크 관리를 위한 설계 패턴 및 컴퍼넌트 식별", SETC'99, Vol.2, KIPS, 1999