CORBA 기반 TINA 서비스 관리 시스템 설계 및 구현

최오훈°, 박혜숙, 백두권
고려대학교 컴퓨터학과 소프트웨어 시스템 연구실
{pens,hspark,baik}@software.korea.ac.kr

The Design and Implementation of TINA Service Management System based on CORBA

O Hoon Choi°, He Sook Park, Doo-Kwon Baik
Software System Lab. Dept. of Computer Science & Engineering,
Korea University

요 약

통신망 단말기의 멀티미디어화로 새로운 서비스에 대한 이용자 요구증대에 의해 공중망은 다양한 서비스를 개방 망 환경에서 수용하고 있다. 또한, 서비스 제공자의 신속한 새로운 서비스 제공, 망 사업자의 통신망에 대한 효율적인 유지보수 및 운용추구에 따라 신규 서비스를 능동적으로 수용하기 위한 새로운 통신망 구조가 필요하다. 본 논문에서는 분산 컴퓨팅 기술과 전기통신기술 통신망 요소기술을 근간으로 하여 분산 처리환경 기술과 객체지향 모델링 및 설계기법을 적용하여, 통신망을 소프트웨어의 계층적 모델로 구조화한 차세대 정보통신망 구조인 TINA(Telecommunication Information Network Architecture)의 서비스 관리 시스템을 구현 시 고려해야 할 몇 가지 문제점을 지적하고, 이러한 문제를 해결하게 위한 방안으로 객체지향 응용 프로그램과 데이터 저장소 사이의 투명한 인터페이스를 제공하는 객체 관리 시스템(Object Management System: OMS) 구조와 서로 다른 저장소에 저장되어 있는 객체들을 효과적으로 관리하기 위하여, 메타데이터를 이용한 OMS-metaData Registry(OMS-MDR)를 제안한다. 또한, 제안된 구조에 근거하여 TINA의 서비스 관리를 위한 객체 지향 응용 어플리케이션을 개발하는데 적용하였다.

1. 서론

새로운 서비스에 대한 이용자의 요구증대와 서비스 제공자의 신속한 서비스 제공, 망 사업자의 효율적인 유지보수 및 운용의 추구로 신규 서비스를 능동적으로 수용하기 위한 새로운 통신망 구조가 필요하다. 이러한 요구를 위해 분산처리환경 기술과 객체지향 모델링 및 설계기법을 적용하여, 통신망을 소프트웨어의 계층적 모델로 구조화한 TINA(Telecommunication Information Network Architecture)[1][2]는 진보된 통신 시스템의 개념이다. 일반적으로 TINA 서비스 관리 계층은 객체지향 기술로 구현된 5개의 서브시스템으로 구성된다. 그러나 객체지향 기법을 적용하여 TINA[1][2]의 서비스 관리 시스템을 구축하는 것에 몇 가지 문제점이 있다. 그 중 대표적인 원인은 객체지향 기법을 적용하여 구현된 응용 프로그램과 내부 객체들의 지속성을 보장하기 위한 데이터 저장소와의 인터페이스 작업이 어렵고 시간이 많이 걸린다는 점이다[3]. 일반적으로 망을 구성하는 서브시스템의 관리에 필요한 관리 객체 (Managed Object: MO)는 한곳에 집중되는 것이 아니라 지역적으로 분산된 여러 저장소에 저장된다. 그러므로 각 MO가 분산환경 하에서 이기종의 하드웨어 및 운영체제 환경에서 관리될 수 있기 때문에 객체지향 어플리케이션과 데이터 저장소 사이의 인터페이스는 플랫폼, 데이터 저장소 종류, 구현언어에 대하여 투명성을 가져야 한다. 이러한 문제점을 해결하기 위하여 본 연구에서는 효과적인 데이터 인터페이스를 구축하기 위해 관리 객체를 정의하고 이를 관리하는 객체 관리 시스템(Object Management System: OMS)을 설계 및 구현하였다. 이 시스템을 통하여 어플리케이션 계층과 실제 데이터 집속기능 미들웨어(ODBC, JDBC)를 분리시키고자 한다. 그리고 어플리케이션 계층과 OMS를 CORBA 기반의 분산 객체 미들웨어로 연결하고자 한다. 또한 서로 다른 저장소에 저장되어 있는 관리 객체들을 효과적으로 관리하기 위하여, 메타데이터를 이용한 OMS-metaData Registry (OMS-MDR)를 제안하고 구현하였다[10].

2. 관련 연구

2.1 CORBA기반의 분산 객체 관리

분산 환경에서 시스템의 이형성 및 분산성 때문에 발생하는 상호 운용성 문제를 미들웨어를 이용하여 해결하려는 연구가 매우 활발히 진행 중이다. Gravano[4]은 미리 계산된 정보를 사용해서 다양한 분산 자원들의 선

택 기법을 제안하였고, Fanchitti[5]는 기존 어플리케이션들을 효율적으로 처리하기 위한 Wrapper 기동 방법을 제시하였으며, Neches[6]는 다양한 소스 자원들에 대한 적절한 선택방법을 제안하였다. 1989년 Object Management Group(OMG)은 분산 환경에서 응용 프로그램들의 통합을 위해 개방형 객체 기반구조로 Object Management Architecture(OMA)를 제정하였다[7]. 응용 프로그램들간의 이형성 및 분산성을 투명하게 해결하고 시스템 및 어플리케이션들간의 상호 운용성 기술을 제공하려는 연구가 객체기반 기술에 근거한 CORBA 기반의 분산 컴퓨팅 구조이다. 본 논문에서는 TINA의 서비스 관리 시스템을 구축하는데 필요한 어플리케이션을 개발하는데 이러한 CORBA의 특성들을 이용한다.

2.2 TINA의 관리 구조

TINA는 발전된 통신 시스템의 유동적인 디자인을 위한 프레임워크로서 전형적인 통신 접근 방법으로 정보 기술을 통합한다[9]. TINA는 망 관리 구조를 망 요소와 망 관리 계층(NML), 서비스 관리 계층(SML) 및 사업 관리 계층(BML)으로 구분한다[2]. SML의 서비스 관리 시스템들은 액세스 관리 시스템, 통신 세션 관리 시스템, 서비스 세션 관리 시스템, 과금 관리 시스템, 가입자 관리 시스템의 서브 시스템들로 구성된다. 현재 망 관리를 위하여 TMN 관리 기술과 OSI시스템 관리 기술을 기본 구조로 적용하고 있다[8]. 이러한 관리 기술은 TMN환경에서 망 관리와 망 요소의 관리가 목적이며 이를 위해 MO를 정의하고 표준 인터페이스를 정의한다.

3. Object Management System : OMS

3.1 OMS의 설계

OMS는 3가지 소프트웨어 관리자 : Data Manager(DM), metaData Registry Manager(DRM), Data Access Manager (DAM)로 이루어진다. 그림 1은 OMS 구조를 보여준다. Interface part는 사용자와 관리자에게 보여지는 user interface부분이며, Internal system part는 실제로 프로세스가 처리되는 부분이다. Internal System part는 오퍼레이션을 요구하는 비즈니스 객체와 실제적인 프로세스를 담당하는 OMS, 분산환경하의 RDB의 부분으로 구성된다.

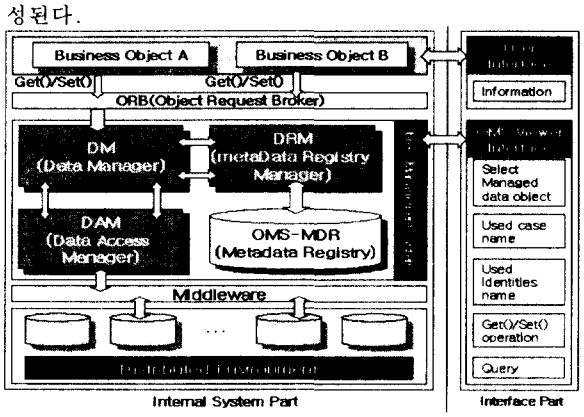

그림 1 관리 객체 시스템의 구조도

OMS에서 DM은 MO들과 직접적으로 상호작용을 한다. 이 상호 작용은 두 가지의 질의 : MO의 내용을 읽기 위한 get() 오퍼레이션과 삽입, 삭제, 갱신을 위한 set() 오퍼레이션이 있다. DM에 의해 입력받은 두 오퍼레이션의 입력 인자 값은 DAM에 의해 SQL 문장으로 변환된 질의 객체 형식이다. DAM은 분산환경하의 DB 연결과 질의 전송, 결과 반환에 관한 처리를 담당한다. DRM은 DAM에 접근하기 전에 어플리케이션에서 원하는 정보를 MDR에 저장하여 자주 검색되거나 요청되는 정보를 보다 쉽게 제공하며, DAM을 통해 처리된 질의어의 정보를 OMS-MDR에 저장하여 질의의 효율성을 높인다.

3.2 OMS-MDR 설계

OMS-MDR[10]을 구성하는 관리 객체 클래스의 스키마는 관리 객체 클래스 이름, 관리 객체 데이터 요소 스키마, 관리 객체 프로시져 스키마로 구성된다. 관리 객체 클래스 이름은 관리 객체 클래스들을 구분한다. 관리 객체 데이터 요소 스키마는 관리 객체 클래스내의 데이터 요소를 나타낸다. 데이터 요소는 자신의 요소 값을 관리 객체 데이터 요소에서 저장하지 않고, 데이터 요소의 ID만을 저장하여 각 데이터 요소 ID에 따라 필요한 정보를 획득한다. 관리 객체 프로시져 스키마는 관리 객체 프로시져 스키마내에 관리 객체 프로시져 내용을 저장하지 않고 오퍼레이션 패키지들의 참조 ID만을 저장한다. 오퍼레이션 패키지는 OMS-MDR의 삽입, 삭제, 생성, 최적화를 위한 오퍼레이션들의 집합이다. 그림 2는 관리 객체 클래스 구조를 나타낸 그림이다

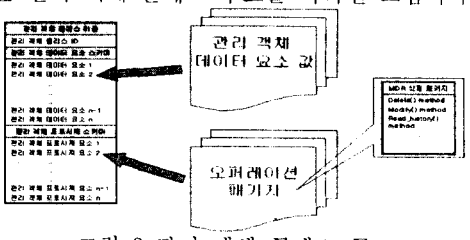

그림 2 관리 객체 클래스 구조

3.3 OMS Process

User Interface의 작업 요청을 받은 DM은 DRM을 통해서 OMS-MDR로부터 MO의 데이터 요소를 전달 받는다. DAM은 MO의 데이터 요소 값을 참조하여 데이터 저장소별로 합당한 질의코드를 생성하여 데이터 저장소와 상호작용을 한다. 데이터 저장소로부터 검색한 내용을 토대로 질의 결과 값이 어플리케이션으로 전달된다. 상호작용을 빠르게 하기 위하여 DRM은 이전에 RDB에서 회수된 MO의 내용을 OMS-MDR에 저장한다. DM과 DRM 그리고 DAM 사이의 Loose coupling은 상호 운영성을 증가한다. 그림 3은 어플리케이션의 요청에 따른 OMS의 작업을 프로세스 어셈블리 라인 다이어그램으로 표현한 것이다. 프로세스 어셈블리 라인은 모델을 프로세스로 구분하여 각 프로세스의 작업 시 필요한 패키지들을 정의하여 각 패키지들의 입출력 내용을 순차적으로 표현하는 다이어그램이다. 그림 3의 프로세스 어셈블리 라인 다이어그램은 OMS의 핵심이 되는 DM, DRM, DAM을 세가지 프로세스로 구분하고, 각 프로세스가 필요로 하는 요소를 관리 객체들이 저장되어 있는 Managed Data Object, RDB의 정보가 있는 RDB, DAM의 질의 형성 시 관련 정보를 저장하는 OMS-MDR, Interface의 정보를 갖는 OMS viewer Interface, User Interface의 패키지로 정의하였다. 각각의 프로세서에서 입,출력되는 흐름을 패키지 형태로 표현하여 어셈블리라인의 입,출력 데이터의 흐름과 프로세스의 흐름을 설계한다.

3.3.1 get() 오퍼레이션

어플리케이션으로부터 get() 요청을 받은 DM은 OMS-MDR에 저장되어 있는 MO 정보를 확인하여 합당한 MO가 존재하면 DAM으로 정보를 전달한다. 본 연구에서의 질의는 OMS-MDR에 저장된 데이터 요소를 인자 값으로 하는 질의 형태이다. 그러므로 데이터 저장소에 의존하는 형태의 질의를 하지않고 다양한 형태의 어플리케이션의 요구를 충족시킬 수 있는 장점이 있다. DAM에서 데이터 저장소를 검색하여 get() 요청에 합당한 튜플을 검색한 후 질의 결과를 DM에게 넘겨준다. 또한 DAM에서 구성된 MO에 관한 데이터 요소는 DRM에 의해 OMS-MDR에 저장된다. 요청을 위한 질의는 Query형의 객체로 만들어져 넘겨지게 된다.

3.3.2 set() 오퍼레이션

DM은 어플리케이션으로부터 데이터 객체의 입력, 갱신, 삭제 요구를 받는다. set() 요청을 받은 DM은 질의를 DRM에게 전달하고 DRM은 질의에 합당한 데이터 요소 정보를 OMS-MDR로부터 추출하여, 데이터 요소 정보를 DAM에게 전달한다. DAM은 데이터 요소 정보로부터 이에 합당한 데이터 저장소의 튜플 정보를 갱신 또는 삭제한다. 또한 그 정보를 OMS-DRM에게 전송하여 OMS-MDR의 정보를 최적 상태로 유지한다.

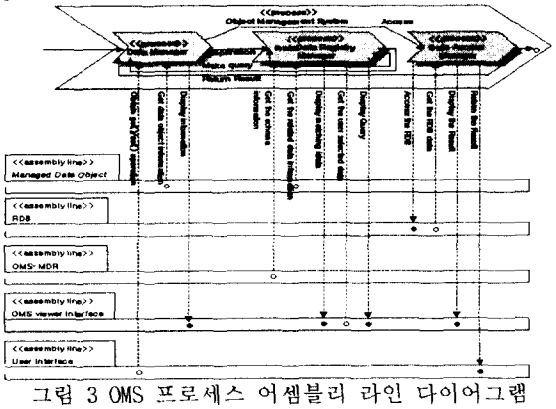

그림 3 OMS 프로세스 어셈블리 라인 다이어그램

3.5 OMS 구현

DM과 DRM, DAM의 구현은 JAVA를 사용하였고, CORBA를 기반으로 하여 DM과 MO의 통신을 위하여 Visibroker 4.0을 사용하고, OMS-MDR의 구현을 위한 DB는 Oracle 8i를 사용하였다. 그림 4는 OMS Viewer의 예이다. OMS와 접속한 후에 변형되거나 참조된 값을 보여주고 있다. 이 참조값으로 OMS-MDR내의 데이터 값의 참조를 나타낸다.

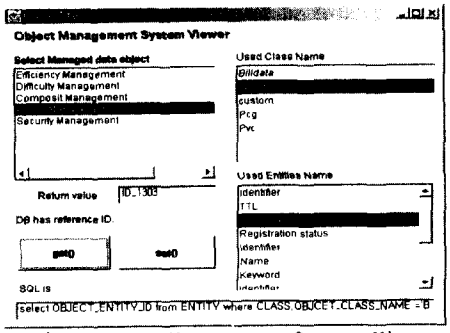

그림 4 Object Management System Viewer

4. 결론 및 향후 연구

본 연구의 목적은 분산환경하의 관리객체(Managed Object: MO)를 위한 객체지향 응용 프로그램과 데이터 저장소 사이의 투명한 인터페이스를 제공하는 OMS를 프로세스 관점에서의 설계와 구현이다. 또한 어플리케이션 계층과 실제 데이터 접속 기능 미들웨어를 분리시키고 어플리케이션 계층과 OMS을 CORBA기반의 분산 객체 미들웨어로 연결할 수 있도록 한다. 그리하여 분산 환경하의 MO의 관리를 효율적으로 할 수 있다. 그리고 서로 다른 저장소에 저장되어 있는 객체들을 효과적으로 관리하기 위하여, 메타데이터를 이용한 OMS-MDR을 제안하고 구현하였다. 이 OMS-MDR은 ISO/IEC 11179와 ANSI X.3.285에서 제안한 개념적 모델을 OMS에 맞도록 변형하여 상호 운용성, 재사용성 등의 핵심 기능을 제공하도록 하였다.

본 논문의 향후 연구로서 제안한 구조와 구현한 시스템에 대한 검증을 위해 OMS 프로세스에 대한 명세에 대한 SPIN을 통한 검증이 필요하다.

5. 참고문헌

[1] W. J. Barr, T. Boyd, and Y. Inoue., "TINA Initiative", IEEE Communications Magazine, Mar. 1993.

[2] TINA-C, http://www.tinac.com

[3] M. Keller, Richard Jensen, Shailesh Agarwal, "Persist Software: Bridging Object-Oriented Programming and Relational Database," Proceeding of ACM SIGMOD International Conference on Management of Data, Washington DC, pp.523-528, 1993

[4] Luis Gravano, Hector Garcia-Molina, and Anthony Tomasic, "The Effective of GLOSS for The Text Database Discovery Problem," Proc. of The ACM SIGMOD Conference, ACM Sigmod Record, Vol.23 No.2, pp. 126-137, May 1994

[5] J.C. Fanchitti, R.King, and O. Boucelma, "A toolkit to Support Scalable Persistent Base Infrastructure," Proc. of the Sixth International Workshop on Persistent Object System, Tarascon, France, Springer-Verlag LNCS, 1994

[6] R. Neches, R.Fikes, T.Finin, T.R. Gruber, "Enabling Technology for Knowledge Sharing," AI Magazine, Vol.12 No.3, pp.37-56, 1993

[7] OMG, Object Management Architecture Guide, Third Edition, Wiley & Sons, 1995

[8] ITU-T Recommendation M.3100 : Generic Network Information Model, July, 1995.].

[9] Roberto Minetti , Flavio Piolini ," TINA-Based Network Resource Information Model for Next-Generation Mobile Systems", The proceeding of the TINA'97-Global Convergence of Telecommunications and Distributed Object Computing, 1997, IEEE

[10] 박혜숙, 나홍석, 남궁영환, 최오훈, 백두권 " 데이터 레지스트리에 기반한 전자상거래 메타데이터 공유 환경" , 한국정보과학회 춘계 학술발표논문집(B), 27권 1호,pp. 60-62, 2000