

□ 특집 □

분산처리 환경에서 클라이언트/서버 기반 객체 관리 시스템 설계

신 영 석* 이 상 백** 정 태 의*** 박 동 선****

◆ 목 차 ◆

- | | |
|-----------------|-----------------|
| 1. 서 론 | 4. 객체 관리 시스템 설계 |
| 2. 객체 지향 분산 서비스 | 5. 결 론 |
| 3. 분산 객체 관리 시스템 | |

1. 서 론

최근의 폭발적인 인터넷(Internet) 이용 증가 경향은 네트워크와 분산 시스템을 기반으로 하는 서비스들의 등장을 가속화하고 있다. 이에 따라 기존의 단순한 클라이언트와 서버간의 통신을 기반으로 하는 서비스 구조로는 다양한 멀티미디어 서비스를 고속으로 제공하고자 하는 이용자의 요구를 만족시킬 수 없게 되었다. 이러한 기존 통신 서비스 구조의 한계는 서비스 제공 시스템의 개념을 단일 시스템에서 네트워크를 매개로 연결된 분산 시스템으로 확장(up-sizing)하면서 극복할 수 있게 되었으며, 분산처리 환경은 이러한 시스템적

성능 확장을 가능하게 하는 핵심적인 기능을 제공하였다. 특히 객체 지향 분산처리 환경은 이를 기반으로 개발되는 서비스 객체의 재사용성 및 이식성 등의 장점이 인정되어 다양한 멀티미디어 서비스 개발의 기반 환경으로 이용되고 있다[1,8]. 이러한 객체 지향 분산 환경을 기반으로 하는 분산 멀티미디어 서비스는 객체의 재사용을 통한 신속한 서비스 개발이 용이할 뿐만 아니라, 서비스 객체의 개발이 개발 환경이나 하부 시스템에 무관하게 수행될 수 있으며, 서비스 객체를 여러 시스템에 분산 배치함으로써 서비스 제공 시스템의 부하를 줄여 서비스의 고속화가 가능한 장점을 지닌다[1]. 그러나 분산 서비스는 객체의 독립적인 개발 및 디버깅 특성과 서비스 객체의 분산 배치 특성으로 인하여 서비스의 통합 및 제공 단계에서 발생하는 객체간 접속 및 정보 전달 오류와 성능에 따른 여러 문제점을 발견하기 어려울 뿐만 아니라, 서비스 객체의 초기화를 비롯한 서비스 객체들의 관리 절차가 복잡한

* 정 회 원 : 한국전자통신연구원 선임연구원

** 정 회 원 : 전북대학교 정보통신공학과 석사과정

*** 중신회원 : 서경대학교 컴퓨터과학과 조교수

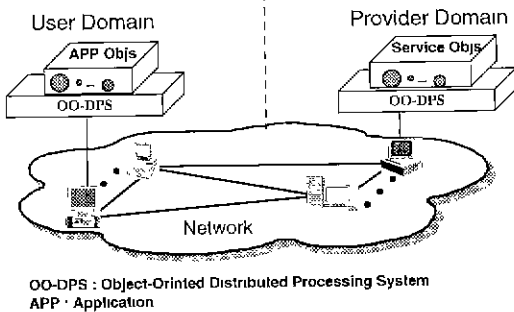
**** 정 회 원 : 전북대학교 정보통신공학과 부교수

단점을 지나고 있다[5,6].

이에 본 논문에서는 객체 지향 분산처리 환경을 기반으로 하는 객체 지향 분산 서비스의 개발과 서비스 제공 단계에서 분산 서비스 객체들의 초기화와 오류 감시 및 복구 등의 객체 관리 기능을 제공함으로써 신속한 서비스 개발 및 안정적인 서비스 제공을 가능하도록 하는 개방형 객체 관리 시스템을 설계하였다.

2. 객체 지향 분산 서비스

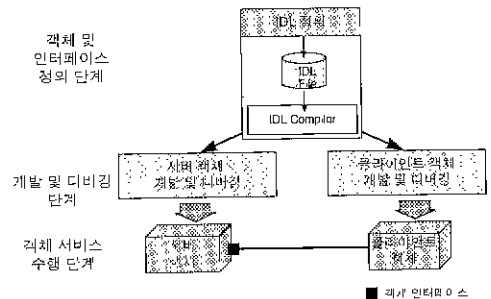
객체 지향 분산 서비스는 네트워크를 통하여 서비스 객체 및 사용자 객체들 사이의 연결 기능과 투명한 정보 전달 기능을 수행하는 객체 지향 분산 처리 환경을 기반으로 하여 제공되는 서비스이다. 일반적으로 분산 시스템 환경에서 사용자 응용 프로그램 구성하는 객체들은 (그림 1)과 같이 클라이언트/서버간의 상호 인터페이스를 통하여 이용자가 원하는 서비스 요청 내용을 서비스 제공자 측에 전달하고, 서비스 객체들 또한 객체들 사이의 인터페이스를 통하여 이용자가 요청한 서비스를 제공한다.



(그림 1) 객체 지향 분산 서비스 구조

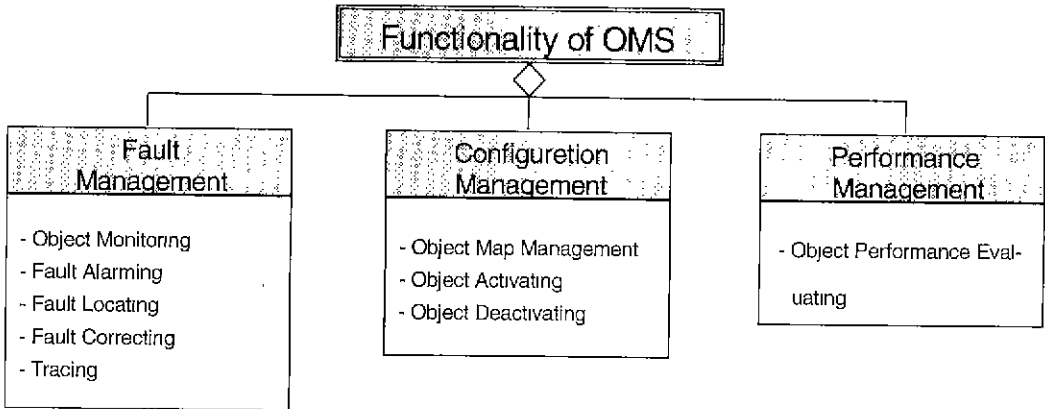
객체 지향 분산 서비스는 객체를 기본 단위로 하여 개발 및 관리된다. 즉, 서비스 개발자는 객체의 개발 과정에서 해당 객체가 제공하는 기능들을 IDL(Interface Definition Language)로 정의한

후, 정의된 IDL에 따라 각 기능의 세부적인 구현을 수행하며, 클라이언트 객체에는 정의된 IDL을 참조하여 서비스의 요청에 필요한 코드를 추가한다[7,8]. (그림 2)는 객체 지향 분산 서비스 객체의 개발 과정을 보이고 있다.



(그림 2) 서비스 객체 개발 과정

이와 같이 객체 지향 개념에 의해 설계되는 분산 서비스는 소프트웨어의 재사용과 유지 보수, 신속한 신규 서비스 제공이 용이한 장점을 가지고 있다. 그러나 객체 지향 분산 서비스는 각각의 객체가 분산 처리 환경에서 상호 독립적으로 개발되고 여러 시스템에 분산 배치되기 때문에 서비스의 통합 시험을 수행하는 디버깅 단계 및 서비스 제공 단계에서 객체간 접속 및 정보 전달시 발생하는 오류를 발견하기 어렵고, 서비스 초기화를 비롯한 객체 관리 절차가 복잡하다. 따라서 서비스 개발자는 이러한 객체간 오류의 디버깅에 상당한 시간을 소모하게 되며, 서비스 운전자 또한 신속한 오류 복구의 어려움과 관리 절차의 복잡성으로 인하여 효율적인 서비스 관리를 수행할 수 없게 된다. 이와 같은 객체 지향 분산 서비스의 개발 및 서비스 제공 단계에서의 문제점은 객체간 접속 및 정보 흐름을 감시하고 신속한 오류 복구 능력 및 초기화 능력 등을 서비스 운용자에게 제공하는 분산 객체 관리 시스템 (Distributed Object Management System: DOMS)을 설치함으로써 용이하게 해결될 수 있다.



(그림 3) 객체 관리 시스템 기능

3. 분산 객체 관리 시스템

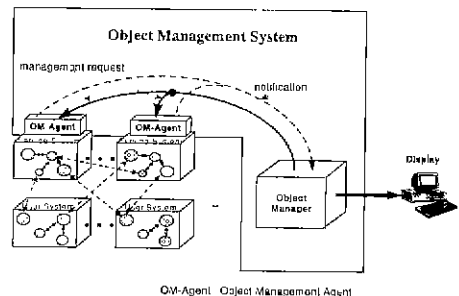
3.1 시스템 기능

분산 객체 관리 시스템은 객체 지향 분산 서비스를 구성하는 객체들을 초기화하고, 객체들의 상태변화를 감시하여 서비스 운영자가 서비스의 진행 상황을 쉽게 파악할 수 있도록 한다. 또한 서비스 객체의 오류 발생시 이를 운영자에게 알리고 신속한 자체 복구 또는 운영자에 의한 복구를 수행토록 하여 안정적인 서비스의 제공을 가능하도록 한다.

(그림 3)과 같이 객체 관리 시스템의 기능은 OSI의 관리 기능에 의한 영역 구분에 따라 오류 관리와 객체 설정 관리, 객체 성능 관리 기능으로 구분된다. 객체 관리 시스템은 객체들 간 접속과 수행에 따른 오류의 관리를 위한 세부 기능으로서 상태감시 기능과 오류 통보, 오류 위치 검출, 오류 복구, 상태 추적 기능을 제공하며, 객체의 설정을 위하여 객체 초기화 기능과 해제 기능 및 객체 배치 정보 관리 기능을 제공한다. 또한 서비스 객체의 성능 측정을 통하여 서비스 운영자가 객체의 효율성을 판단할 수 있도록 한다.

3.2 시스템 구조

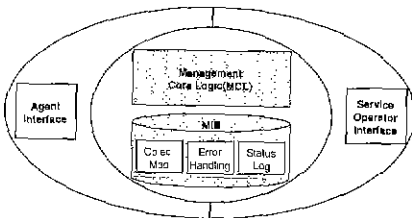
객체 관리 시스템은 여러 시스템에 분산되어 있는 서비스 객체들의 효율적 관리를 위한 분산 소프트웨어로서, 전체적인 서비스 관리 및 운영자 인터페이스를 수행하는 객체 관리자(Object Manager: OM)와 각각의 서비스 시스템에 설치되는 객체 관리 에이전트(Object Management Agent: OMA)로 구성된다. 객체 관리자와 에이전트의 인터페이스는 객체 관리 수행을 위한 인터페이스와 상태 감시를 위한 인터페이스로 구분된다. 객체 관리자는 운영자의 요구나 서비스 진행 상황에 따라 에이전트에게 객체의 관리를 위한 요구 메시지를 전달하여, 객체 관리 에이전트는 객체 관리자의 요구에 따라 객체를 관리하고 자신의 관리하에 있는 객체들의 상태 정보를 객체 관리자에게 전달한다.



(그림 4) 객체 관리 시스템 구조

3.2.1 객체 관리자

객체 관리자는 서비스 운용자의 요구에 따라 서비스 시스템에 분산되어 있는 객체 관리 에이전트에게 객체 관리 요청 메시지를 전달하고, 에이전트로부터 전달받은 관리 대상 객체들의 상태 정보를 서비스 운용자에게 표현하는 기능을 제공하는 객체 관리 시스템의 중심부로서 (그림 5)는 객체 관리자의 구조를 보이고 있다.



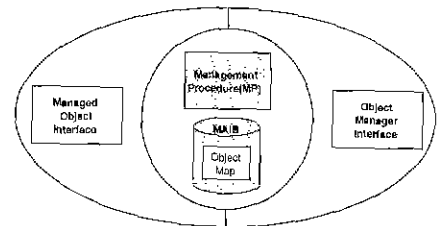
(그림 5) 객체 관리자

객체 관리자는 객체 관리 핵심부(Management Core Logic: MCL)와 객체 관리 정보 블록(Management Information Block: MIB), 에이전트 인터페이스, 서비스 운용자 인터페이스로 구성된다. 객체 관리 핵심부는 서비스 운용자의 관리 요구 내용을 파악하고, 이에 따라 에이전트에게 실질적인 관리 작업을 수행하도록 지시하는 부분이다. 객체 관리 정보 블록은 객체 배치 정보와 오류 발생시 자체 복구를 위한 절차를 저장하고 있는 오류 처리 정보, 객체의 상태 변화를 저장하고 있는 객체 상태 로그로 구성되며, 이들 정보는 객체 관리 핵심부에 의하여 이용된다. 객체 관리 핵심부와 객체 관리 정보 블록을 둘러싸고 있는 에이전트 인터페이스와 서비스 운용자 인터페이스는 각각 객체 관리 에이전트와 서비스 운용자와의 통신 또는 명령 전달 기능을 제공한다.

3.2.2 객체 관리 에이전트

객체 관리 에이전트는 객체 관리자의 요구에

따라 자신의 영역(domain) 내에 있는 객체들을 관리하고, 객체들의 상태 변화를 객체 관리자에게 전달하는 기능을 제공하는 부분으로서, (그림 6)은 객체 관리 에이전트의 구조를 보이고 있다.



(그림 6) 객체 관리 에이전트

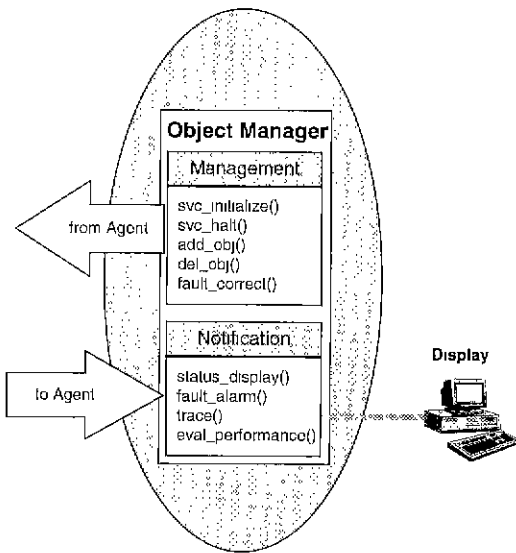
객체 관리 에이전트는 객체 관리 수행부(Management Procedure: MP)와 객체 관리 에이전트 정보 블록(Management Agent Information Block: MAIB), 관리 객체 인터페이스, 객체 관리자 인터페이스로 구성된다. 객체 관리 수행부는 객체 관리자의 요구에 따라 실질적인 객체 관리를 수행하는 부분으로서, 에이전트 정보 블록의 객체 배치 정보에 저장되어 있는 객체 목록을 참조하여 자신의 관리 영역 내부의 객체를 관리한다. 객체 관리 수행부는 관리 대상 객체나 객체 관리자와의 통신을 위하여 관리 객체 인터페이스와 객체 관리자 인터페이스를 이용한다.

4. 객체 관리 시스템 설계

본 연구에서 설계한 객체 관리 시스템은 OMG CORBA를 기반으로 하는 분산처리 환경에서 한 개의 객체 관리자와 한 개 이상의 객체 관리 에이전트로 구성되며, 객체 관리자와 객체 관리 에이전트 간의 메시지 전달은 클라이언트-서버 간의 통신 방식을 근간으로 하여 수행토록 하였다. 본 장에서는 객체 관리 시스템의 각 부분별 세부 설계 내용을 기술한다.

4.1 객체 관리자

객체 관리자의 설계는 (그림 7)과 같이 사용자의 요구 또는 서비스 진행 상황에 따라 객체의 관리를 수행하는 객체 관리 인터페이스와 객체 관리 에이전트로부터 객체의 상태 정보를 수신하고 이를 서비스 운용자에게 표현하는 상태 감시 인터페이스의 정의를 통하여 수행하였다.



(그림 7) 객체 관리자 설계

객체 관리 인터페이스에서 제공되는 기능은 서비스 초기화와 종료, 서비스 객체의 추가 및 삭제, 오류 복구의 다섯 가지로, 객체 관리자는 서비스 운용자로부터 각 기능을 요청 받게 되며, 요청된 명령은 실질적인 관리를 위하여 에이전트에게 전달된다. 상태 감시 인터페이스에서 제공되는 기능은 객체 상태의 출력과 오류 통보, 상태 추적, 성능 평가의 네 가지이며, 객체 관리자는 객체 관리 에이전트로부터 객체의 상태 정보를 수신하여 서비스 운용자의 화면에 이들 정보를 출력한다. <표 1>은 객체 관리자가 수신하는 상태 정보의 내용을 보이고 있다.

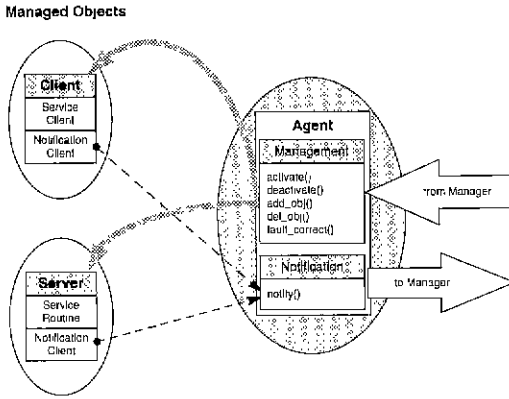
<표 1> 객체 상태 정보

정보이름	저장 값	기 능
Source	객체의 이름	상태 정보를 송출한 객체를 표시함
Destination	객체의 이름	Source 객체의 인터페이스를 수행하는 객체를 표시함
Event	객체 연결 상태 변화, 객체간 정보 전달, 오류 발생 등	Source 객체와 Destination 객체 사이의 인터페이스 내용을 표시함
Operation	서비스 내용	서버가 클라이언트에게 제공하는 실질적인 서비스 내용(또는 요청 내용)
Cur_time	현재 시간	Event가 발생한 시간으로서 객체의 성능 평가에 이용됨

4.2 객체 관리 에이전트

객체 관리 에이전트의 설계는 객체 관리자로부터 수신한 요구 내용에 따라 객체 관리를 수행하는 객체 관리 인터페이스와 객체들로부터 각 객체의 상태 정보를 수신하여 객체 관리자에게 전달하는 상태 감시 인터페이스를 정의하여 수행하였다.

객체 관리 인터페이스에서 제공되는 기능은 객체의 실행과 해제, 추가 및 삭제, 오류 복구의 다섯 가지로 구성되며, 객체 관리 에이전트는 객체 관리자로부터 이들 기능을 요청받아 실질적인 관리를 수행한다. 상태 감시 인터페이스에서는 객체 관리자와는 달리 상태 통보의 한가지 기능만을 제공하는데, 이는 객체 관리 에이전트에서는 객체의 상태를 파악하고 분석할 필요 없이 단순히 객체 관리자에게 전달하는 기능만이 요구되기 때문이다. (그림 8)은 객체 관리 에이전트와 관리대상 객체간의 관계를 보이고 있다.



(그림 8) 객체 관리 에이전트와 관리대상 객체간 관계

5. 결 론

객체 지향 분산처리 환경을 기반으로 하는 분산 서비스는 서비스 객체의 재사용성 및 이식성 등의 장점이 인식되어, 분산처리를 바탕으로 하는 다양한 멀티미디어 서비스 개발을 위한 플랫폼으로 이용되고 있다. 이러한 객체 지향 분산 서비스는 객체의 재사용을 통한 신속한 서비스 개발이 용이할 뿐만 아니라, 서비스 객체의 개발과 운용이 개발 환경이나 운용 시스템에 무관하게 수행될 수 있으며, 서비스 객체를 여러 시스템에 분산 배치함으로써 서비스 제공 시스템의 부하를 줄여 서비스의 고속화가 가능한 장점을 지닌다. 그러나 객체 지향 분산 서비스는 객체 단위의 독립적인 개발 및 디버깅 특성과 서비스 객체의 분산 배치 특성으로 인하여, 서비스의 통합 및 제공 단계에서 발생하는 객체간 접속 및 정보 전달 오류를 발견하기 어렵을 뿐만 아니라, 서비스의 초기화 등의 관리 절차가 복잡한 단점을 지니고 있다. 따라서 서비스 개발자 및 관리자는 서비스의 디버깅 및 오류 복구, 초기화 등의 작업에 상당한 시간을 소모하게 된다.

이에 본 논문에서는 객체 지향 분산처리 환경을 기반으로 하는 서비스의 개발 및 제공 단계에서 서비스를 초기화하고, 객체들 사이의 접속 및 정보 흐름 감시를 통하여 서비스 개발자 및 관리자의 신속한 서비스 오류 감지 및 복구를 용이하게 하는 객체 관리 시스템을 설계하였다. 설계한 객체 관리 시스템은 객체 관리자와 객체 관리에 이진트로 구성되어 있으며, 이들 각 부분은 클라이언트-서버 간의 통신 방식을 이용하여 메시지를 전달한다. 설계된 객체 관리 시스템은 분산 객체 초기화 및 상태 감시 기능 외에 상태 추적 기능과 객체 성능 평가 기능, 객체 배치 기능 등을 제공하고 있어, 다양한 객체 지향 분산 서비스의 효율적 개발 및 관리에 이용될 수 있을 것이다.

참고문헌

- [1] Randy Otte, Paul Patrick, Mark Roy, "Understanding CORBA", 1996.
- [2] IONA Technologies, "Orbix 2.0 Programming Guide & Reference Guide", 1995.
- [3] MetaCard Corporation, "MetaCard 2.0 Users Guide", Jan., 1996.
- [4] Y. S. Shin, H. J. Oh and B. D. Ko, "The Desktop Video Conference System Based on Open Networking Architecture", The Proceeding of ITC-CSCC'97, pp. 583-586, Nov.
- [5] 신영석, 이상백, 박동선, 고병도, "개방형 서비스 객체 모니터링 시스템 설계," HCI '97 논문집, pp. 356-360, 1997.2.
- [6] Young Seok Shin, Sang Beak Lee and Dong Sun Park, "An Object Management System for Open Distributed Services", The Proceeding of APNOMS'97(to appear), Nov., 1997.
- [7] Douglas C. Schmidt, Aniruddha S. Gokhale,

Timothy H. Harrison and Guru Parulkar, "A High-Performance End System Architecture for Real-Time CORBA", IEEE Communication Magazine, pp. 72-77, Feb., 1997.

- [8] Douglas C. Schmidt, Tim Harrison, Ehab Al-Shaer, "Object-Oriented Components for High-Speed Network Programming", The Proceeding of USENIX'95, June, 1995.
- [9] IONA Technologies, "OrbixOTM brings new functionality to CORBA", Orbix Journal, pp. 16-17, Q2, 1997.
- [10] RACE ACTS, RACE ACTS home page, <http://www.infowin.org/ACTS/ANALYSYS/PROJECTS/ACT003.html>
- [11] Tuncay Saydam, Xavier and Simon Znaty, "A Service Management Architecture", The Proceeding of ICT 97, pp. 234-237, April, 1997.



신 영 석

- 1982년 전북대학교 전자공학과 (학사)
- 1984년 전북대학교 전자공학과 (석사)
- 1993년 전북대학교 전자공학과 (박사)

1993년-1994년 일본 NTT 통신망종합연구소 객원연구원
 1984년-현재 한국전자통신연구원 선임연구원
 관심분야 : 개방형 통신망 구조(TINA), 분산처리 시스템, 객체 모델링, ATM 프로토콜



이 상 백

- 1996년 전북대학교 정보통신공학과 (학사)
- 1996년-현재 전북대학교 정보통신공학과 석사과정
- 관심분야 : 분산처리 시스템, 멀티미디어, 개방형 통신망 구조(TINA)



정 태 의

- 1979년 : 고려대학교 전자공학과 (학사)
- 1982년 · 미국 오하이오주립대 전기공학과(석사)
- 1989년 : 미국 오클라호마대학 전산학과 (석사)

1994년 · 미국 오클라호마대학 전산학과 (박사)
 1983년-1986년 금성반도체연구소 컴퓨터부문 선임연구원
 1986년-1987년 United Microtek, Inc. (San Jose, California) Engineering Manager
 1995년-현재 서경대학교 컴퓨터학과 조교수
 관심분야 : 형식언어, 컴파일러, 알고리즘, 객체지향 시스템



박 동 선

- 1979년 고려대학교 전자공학과 (학사)
- 1984년 미국 미주리 주립대 전기 및 컴퓨터공학과(석사)
- 1990년 미국 미주리 주립대 전기 및 컴퓨터공학과 (박사)

1991년-현재 전북대학교 정보통신공학과 부교수
 관심분야 : 멀티미디어 서비스 컴퓨터 시각, 패턴 인식