

분산 객체그룹 모델의 수행감시를 위한 원격 모니터링 시스템 환경 구축

정희원 이 건 엽*, 이 원 중**, 주 수 중**

A Construction of Remote Monitoring System Environment for Tracing Execution of Distributed Object Group Model

Geon-Youb Lee*, Won-Jung Lee**, Su-Chong Joo** *Regular Members*

요 약

우리는 TINA와 같은 개방형 정보통신망 구조를 정립하기 위해서 정의된 객체그룹의 개념 및 요구사항들과 기능들을 이용하여 분산 객체그룹 모델을 구현하였다^[12,14,15]. 본 논문은 구현된 분산 객체그룹 모델 환경에서 객체그룹들의 수행과정을 검증하기 위한 방안으로, 객체그룹간 또는 객체그룹 내에서 연산객체들의 관리 및 서비스 접속에 따른 객체들의 상태변화 및 정보의 흐름을 파악할 수 있는 객체그룹 수행감시를 위한 원격 모니터링 시스템을 설계 및 구현하였다. 본 시스템은 각 객체그룹 내의 그룹 관리자 객체들로부터 모니터링 정보를 수집하는 모니터링 처리 모듈, 수집된 모니터링 정보를 저장하는 모니터링 저장 모듈, 사용자에게 모니터링 또는 수집된 정보를 표현하는 모니터링 출력 모듈로 구성된다. 모니터링 정보로는 객체그룹 내의 연산객체 또는 서브 객체그룹의 생성, 삭제, 활성화, 비활성화 등의 상태들과 이들 수행 시 객체그룹 내의 객체 구성요소들 간의 접속 과정 및 접속되는 순서와 시간별 궤도들을 GUI 환경에서 시각화한 아이콘, 텍스트 및 그래픽 형태로 보이도록 하였다. 앞으로 본 시스템은 위에서 언급한 모니터링 정보를 참조하여 분산 객체그룹 시스템에서의 성능향상의 쟁점들을 연구할 목적으로 사용될 것이다.

ABSTRACT

We have been implemented the object group model using the concepts of object group, its requirement and functions defined for specifying the open telecommunication network structure, like TINA. Our paper is to verify the executing procedures of object groups in the distributed object group model environment implemented. Hence, when objects in the same or different object groups are binding one another for service or management, for showing state changes and the information flows between/among the binding objects, we designed and implemented an remote monitoring system for tracing execution procedures of object groups. Our system consists of 3 modules, the monitoring process module for catching information collected from object group manager in each object group, the monitoring storage module for storing information caught from the monitoring processing module, and the monitoring output module which presents information caught or stored to users via GUI, as monitoring information.

This monitoring information is categorized create, delete, enable and disable states of a sequence of objects, and binding orders and binding time of these objects, showed as visual icons, texts, graphic types on GUI screen

* 군장대학 (gylee@kunjang.ac.kr)

** 원광대학교 (cosmos@gaebyok.wonkwang.ac.kr, scjoo@wonkwang.ac.k)

논문번호 : 99087-0305, 접수일자 : 1999년 3월 5일

* 본 논문은 1999년 원광대학교 교내 연구비의 지원에 의해 연구되었습니다.

of the server, our remote monitoring system.

In future, this system will be used for investigating issues of performance improvement of the distributed object group system by referring the monitoring information mentioned above.

1. 서론

컴퓨터 기술 분야와 초고속 통신망의 급속한 발전으로 분산 컴퓨팅 환경에 대한 사용자의 요구는 서비스의 질적, 양적인 측면에서 증대되고 있다. 분산 객체 시스템을 지원하는 TINA(Telecommunication-Information Networking Architecture)와 같은 개방형 정보통신망 구조 환경에서는 많은 서비스, 어플리케이션, 시스템 객체들은 전형적으로 분산된 컴퓨팅 시스템들 상에서 복잡한 인터페이스를 가진다. 이렇게 규모가 큰 개방형 통신망 구조의 시스템을 설계할 때, 객체들의 추상화 한계와 인터페이스들의 복잡성 문제가 발생한다^[1,2,3]. 이에 따라 개방형 정보통신망 구조에서는 객체들의 집합을 관리하는 구조화 메커니즘으로 객체그룹을 정의하고 있다^[4,6]. 그 동안 우리는 TINA와 같은 개방형 정보통신망 구조를 정립하기 위해서 정의된 객체그룹의 개념 및 요구사항들과 기능들을 이용하여 분산 객체 그룹 모델을 구현하였다^[12,14,15].

이를 기반으로 한 분산 환경에서 멀티미디어 서비스는 서비스 구성 객체들간의 접속과 정보흐름을 효율적으로 관리하기 위하여, 서비스 객체그룹을 모니터링 하거나 객체그룹 내의 객체간의 정보흐름을 모니터링 하여 관리와 운영에 편리성을 제공할 수 있는 모니터링 시스템이 요구된다^[5,13]. 따라서 본 논문에서는 개별 객체들의 관리 복잡성을 극복하고 개별 객체들을 기능적 또는 효율적 관리를 위해 서비스 형태들의 집합으로 객체들을 그룹화 한 객체 그룹 모델^[12,14]의 수행과정을 고찰 및 분석하기 위한 객체그룹 원격 모니터링 시스템을 설계 및 구현하였다. 구현된 객체그룹 원격 모니터링 시스템은 객체그룹간에 발생하는 연산객체들의 관리 접속에 따른 객체들의 상태변화와 객체간의 접속과정 및 접속시간을 분석하여 객체들의 수행과정을 모니터링 한다. 본 시스템은 3가지 모듈들로, 객체그룹 내의 그룹 관리자 객체로부터 모니터링 정보를 수집하는 처리 모듈, 모니터링 정보를 저장하는 저장 모듈, 사용자에게 모니터링 된 정보를 표현하는 출력 모듈로 구성된다. 모니터링 정보는 객체그룹 모델 내

의 연산객체(Computational Object) 또는 서브 객체 그룹의 생성, 삭제, 비활성화, 활성화 등의 정보와 이들 객체들이 수행 시 객체그룹 내의 구성요소(객체그룹 내 포함되는 모든 객체)들간의 접속과정 및 접속되는 순서와 시간별 궤도들이며, 이들 정보는 GUI 환경에서 시각화한 아이콘, 텍스트 및 그래픽 형태로 보이도록 하였다. 앞으로 본 시스템은 위에서 언급한 모니터링 정보를 참조하여 분산 객체 그룹 시스템에서의 성능향상의 쟁점들을 연구하는 목적으로 사용될 것이다.

본 논문의 구성은 다음과 같다. II장에서는 검증하기 위한 객체그룹의 구조 및 특성, 객체그룹을 구성하는 요소들간의 인터페이스에 대해 설명하고, III장에서는 구축한 객체그룹 원격 모니터링 시스템의 구조 및 특성, 모니터링 방안에 대하여 설명하였다. IV장에서는 모니터링을 수행한 객체의 생명주기를 스크린상의 GUI 환경으로 표현하였고, 객체들의 생명주기에 따른 객체들의 수행 시간을 확인하였으며, 객체그룹의 구성요소들간의 접속시간을 분석하였다. 그리고 V장에서는 결론을 맺는다.

II. 분산 객체그룹

분산처리 환경에서 서비스, 어플리케이션이나 시스템 객체들은 다수의 컴퓨팅 시스템들 상에 분산되어 있는 객체들의 집합으로 구성되고 있기 때문에, 개방형 정보통신망 구조에서는 개방형 정보통신망의 설계 시 복잡성과 관리의 어려움을 해결하기 위해 개별 객체들 대신에 객체들의 집합을 하나의 단위로 관리할 수 있는 객체그룹의 개념을 정립하고 있다^[4,7,9].

1. 분산 객체그룹의 구조 및 기능

객체그룹을 구성하는 요소객체들은 그림 1에 나타난 것과 같이 실질적인 서비스를 제공하는 연산객체(CO : Computational Object), 객체그룹을 관리하기 위한 그룹 관리자(GM) 객체, 객체그룹의 접근을 위한 관리 인터페이스(M-I/F) 객체와 서비스 인터페이스(S-I/F) 객체, 객체에 대한 정보 관리와 보안 규칙 관

리를 위한 객체 인스턴스 레포지토리(Object Instance Rep.) 객체와 보안 레포지토리(Security Rep.) 객체, 객체그룹 내 생성 가능한 객체들에 대한 정보를 보관하는 구현 맵(Impl. Map) 객체, 객체그룹 내 계층화된 객체그룹으로 서브 객체그룹(Sub Object Group) 객체들이 있다. 객체그룹을 구성하는 객체들의 기능 및 특성은 다음과 같다^[12,14,15].

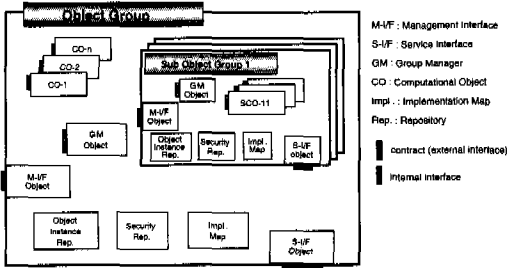


그림 1. 객체그룹 구조

그룹 관리자 객체는 객체그룹 내 연산객체들과, 서브 객체그룹, 객체 인스턴스 레포지토리 객체, 보안 레포지토리 객체, 구현 맵 객체 등을 관리한다. 관리 인터페이스 객체를 통하여 객체그룹 외부의 객체와 내부의 객체들간의 제어나 관리 오퍼레이션에 대한 중개자 역할을 하며, 객체그룹 템플릿에 규약된 연산객체만을 관리한다. 오퍼레이션은 생명주기 서비스, 정보객체 리스트 서비스, 보안 서비스, 초기화 서비스가 있다. 그룹 관리자 객체는 연산객체, 서브 객체그룹, 객체그룹의 생성을 요청하고 객체그룹 원격 모니터링 시스템에 생성에 대한 정보를 전송한다. 또한 객체그룹 내 연산객체, 서브 객체그룹, 객체그룹의 삭제 오퍼레이션을 이용하여 연산객체나 서브 객체그룹을 삭제하고 이에 대한 정보를 객체그룹 원격 모니터링 시스템에 전송한다.

연산객체는 실질적인 서비스를 제공하는 객체로 어플리케이션 요소들간의 상호작용 및 연산객체들 사이의 관리를 기술한다. 서브 객체그룹 내 연산객체나 다른 객체그룹의 연산객체간에 접속을 요구하는 경우에는 객체그룹의 관리 인터페이스를 통해 이루어진다. 서비스 요청을 받으면, 서비스를 제공하기 전에 보안 레포지토리 객체와 객체 인스턴스 레포지토리 객체를 이용하여 요청 객체의 접근 가능 여부와 자신의 상태를 검색한 후, 결과에 따라 수행한다.

관리 인터페이스 객체는 그룹 관리자 객체가 외부와 연결될 때 관리 서비스를 수행하기 위한 객체로

그룹 관리자 객체의 부하를 줄이기 위해 이를 통하여 외부와 연결된다. 객체에 대한 정보 변경과 보안 조건의 변경 등을 그룹 관리자 객체에게 요구하고, 객체에 대한 정보 탐색과 요청 객체에 대한 보안 검색을 수행한다. 그리고 그룹 관리자 객체를 생성하고 삭제하는 기능을 수행한다.

서비스 인터페이스 객체는 다른 객체그룹에서 객체그룹 내 연산객체에 대한 서비스 수행 요구 시, 해당 연산객체와의 접속 서비스를 제공하며, 접속하는 외부의 객체 속성과 객체 이름을 근거로 객체그룹 내 연산객체들과 접속한다.

객체 인스턴스 레포지토리 객체는 객체그룹 내의 객체에 대한 정보를 관리하는 객체로 그룹 관리자 객체를 통해 등록된 객체에 대한 정보를 객체 정보 객체에 두어 객체 정보를 관리한다. 객체 정보 객체는 정보 객체 이름, 연산객체의 레퍼런스를 결합할 정보와 객체의 상태를 관리한다. 객체 인스턴스 레포지토리 객체는 객체그룹 내 객체들의 정보를 생성, 삭제, 수정, 검색 등을 수행한다.

보안 레포지토리 객체는 각 객체마다 정보 객체 인스턴스와 대응하는 ACL(Access Control List) 객체를 두어 객체그룹 내의 객체들에 대한 보안 규칙을 관리하고, ACL 객체를 생성, 삭제, 갱신, 속성 획득 등의 기능을 수행한다. ACL 객체는 액세스를 객체를 생성, 삭제, 속성 획득, 속성 변경 등을 수행한다.

구현 맵 객체는 객체그룹 내 객체를 생성하기 위한 객체들과 CORBA 정의 객체간의 레퍼런스를 관리한다. 구현 맵 객체, 구현 정보 객체, 구현 정보 구조로 구성되며, 구현 맵 객체는 객체와 CORBA 정의 객체간의 매핑 정보를 가지고 있는 구현 정보 객체를 두어 관리한다. 구현 맵 객체는 지정된 구현 정보 객체의 생성, 삭제, 수정, 검색 등을 수행한다.

서브 객체그룹은 객체그룹 내의 계층화된 구조로 존재하는데 객체그룹과 동일한 구성 요소를 가지고 있으며 동일한 기능을 수행한다.

2. 객체그룹 구성 요소들간의 인터페이스

아래 그림 2는 객체그룹을 구성하고 있는 요소들간의 수행 시 인터페이스들을 나타내고 있다. 그들간의 인터페이스에 의한 수행과정은 다음과 같다.

- ① 객체그룹 외부에서 서비스 요청이 관리 인터페이스 객체에게 전달되면 관리 인터페이스 객체는 객체 인스턴스 레포지토리 객체에게 객체그룹 내 객체에 대한 정보 검색을 요청하고 그 결과를 반환 받는

다.

② 관리 인터페이스 객체는 객체그룹 내 객체들에 대한 보안 규칙을 관리하는 보안 레포지토리 객체에게 접근 가능 여부를 요청하고 그 결과를 반환 받는다.

③ 관리 인터페이스 객체는 객체 및 서브 객체그룹의 생성, 삭제, 활성화, 비활성화와 객체 정보에 대한 생성, 수정 및 보안 조건의 수정, 검색 등의 서비스를 그룹 관리자 객체에게 요청하고 수행 결과를 반환 받는다.

④ 그룹 관리자 객체는 객체에 대한 정보 변경 및 검색을 객체 인스턴스 레포지토리 객체에게 요청하고, 수행 결과를 반환 받는다.

⑤ 그룹 관리자 객체는 객체에 대한 보안 조건의 변경 및 검색, ACL 객체를 검색하도록 보안 레포지토리 객체에게 요청하고, 해당 정보 또는 결과를 반환 받는다.

⑥ 그룹 관리자 객체가 그룹 내 객체 또는 서브 객체그룹을 생성할 때 구현 맵 객체에게 객체의 매핑 정보를 요구하고, 결과를 반환 받는다.

⑦ 그룹 관리자 객체가 객체를 생성, 삭제, 활성화, 비활성화를 수행한 후, 객체 인스턴스 레포지토리 객체에게 객체 정보 변경을 요청하고, 그 결과를 반환 받는다.

⑧ 상위 객체그룹의 그룹 관리자 객체가 서브 객체그룹의 관리 인터페이스 객체에게 관리 서비스를 요청하고, 처리된 결과를 반환 받는다.

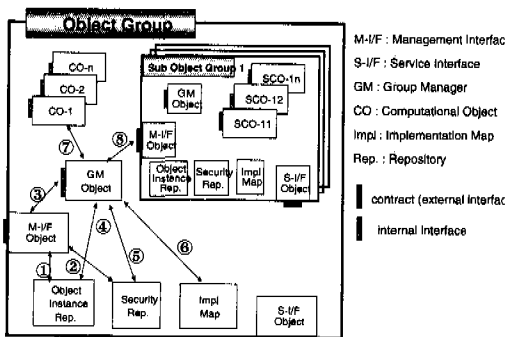


그림 2. 객체그룹 내 구성 요소들간 인터페이스 관계

III. 분산 객체그룹 지원 원격 모니터링

분산 시스템 기반의 응용서비스들은 객체들 사이에서 정보를 주고, 받으면서 기능을 수행하기 때문

에 서비스 구성 객체들간의 접속과 정보흐름을 효율적으로 관리하기 위하여, 서비스 객체그룹을 모니터링 하거나 객체그룹 내 객체들간의 정보흐름을 파악하여 관리와 운영에 편리성을 제공할 수 있는 객체그룹 원격 모니터링 시스템이 요구된다.

본 논문에서 구현한 객체그룹 원격 모니터링 시스템은 모니터링 대상 객체그룹 내 객체들의 상태변화, 객체그룹간의 접속과정, 객체들간의 접속시간을 모니터링하고 분석하여 객체그룹을 효율적으로 관리함으로써 시스템의 성능향상 방안을 강구할 수 있다. 이를 위해 객체그룹 원격 모니터링 시스템은 정보 수집 기능, 정보 출력 기능, 정보 저장 기능, 성능 평가 기능 등을 지원한다.

1. 모니터링 시스템의 구조 및 기능

객체그룹 지원 원격 모니터링 시스템은 그룹 관리자 객체로부터 전송되어진 모니터링 정보를 처리하는 모듈로 구성된 처리부와 모니터링 정보를 저장하는 모듈을 가진 저장부, 모니터링 서버 시스템 상에 모니터링 정보를 GUI 환경으로 출력해주는 모듈로 이루어진 출력부로 구성된다. 그림 3은 3개의 객체그룹들간에 수행상태를 모니터링 하는 환경을 가진 객체그룹 지원 원격 모니터링의 예를 나타낸다. 이들 객체그룹은 하나 또는 그 이상의 분산된 컴퓨팅 시스템들 상에서 수행이 가능하다.

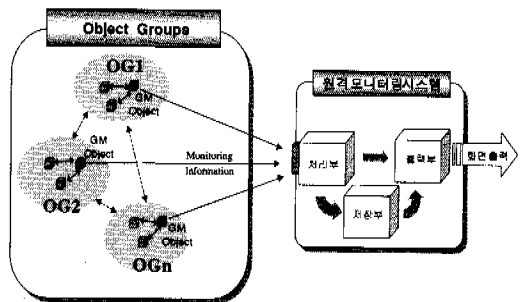


그림 3. 객체그룹 원격 모니터링 예

1) 모니터링 처리부

원격 모니터링 시스템의 모니터링 처리부에서는 그룹 관리자 객체로부터 모니터링 대상이 되는 객체그룹 내 연산객체 및 서브 객체그룹의 생성, 삭제, 활성화, 비활성화 등의 상태 변화와 이로 인하여 발생하는 구성 객체들간의 접속과정 및 접속시간에 관한 정보를 받는다. 모니터링 처리부는 그룹 관리자

객체로부터 전송된 모니터링 정보를 수집하고 처리하여 원격 모니터링 저장부 또는 원격 모니터링 출력부로 전송한다.

2) 모니터링 저장부

원격 모니터링 저장부는 객체그룹간 또는 객체그룹 내 객체들간의 접속 관계 및 정보 흐름, 각 객체그룹 내 객체들의 속성 값과 관련 정보들을 저장하는 기능을 수행하며, 객체그룹의 모니터링 신호와 정보를 객체그룹별로 저장하고 필요 시 모니터링 출력부로 제공한다.

3) 모니터링 출력부

원격 모니터링 출력부는 모니터링 처리부에서 전송된 정보나 원격 모니터링 저장부에 저장된 모니터링 정보를 GUI를 이용하여 서비스 관리자 화면에 디스플레이 해 준다. 이것을 이용하여 서비스되는 객체들의 현황을 파악할 수 있으며, 객체그룹과 객체그룹간 객체와 객체사이의 접속 상태나 정보 흐름의 오류 발생을 발견할 수 있다.

2. 모니터링 시스템의 접속 오퍼레이션

객체그룹 내에 있는 그룹 관리자 객체로부터 객체그룹 원격 모니터링 시스템으로 모니터링 정보를 전달하는 오퍼레이션들은 다음과 같다. 이때 그룹 관리자 객체는 그림 2에서 나타난 객체그룹 내의 구성요소들과의 인터페이스를 통해서 각 구성 요소의 수행 정보를 얻게된다.

1) 생성 (Create)과 삭제 (Delete)

생성과 삭제 오퍼레이션은 객체그룹, 서브객체그룹, 연산객체의 생성 및 삭제를 알리는 신호로서 모니터링 대상 객체그룹의 그룹 관리자 객체는 생성 및 삭제에 따라 create 및 delete와 이에 따른 객체그룹 내의 구성 요소들의 상태에 대한 정보를 객체그룹 원격 모니터링 시스템의 처리부에 전송한다.

2) 활성화 (Enable)와 비활성화 (Disable)

활성화와 비활성화 오퍼레이션은 연산객체, 서브객체그룹 및 객체그룹의 활성화 및 비활성화를 알리는 신호로서 모니터링 대상 객체그룹의 그룹 관리자 객체는 활성화 및 비활성화에 따라 오퍼레이션 enable 및 disable과 이에 따른 객체그룹 내 구성 요소들의 상태에 대한 정보를 객체그룹 원격 모니터링 시스템의 처리부에 전송한다.

3) 접속시간 (Binding time)

시간 오퍼레이션은 연산객체, 객체그룹, 서브 객체그룹의 생성, 삭제, 활성화, 비활성화로 인한 구성요소 객체들의 접속시간을 알리는 신호로 모니터링 대상 객체그룹의 그룹 관리자 객체는 생성, 삭제, 활성화, 비활성화 수행 시 접속시간에 대한 정보를 원격 모니터링 시스템의 처리부에 전송한다.

표 1. 객체그룹의 모니터링용 이벤트 오퍼레이션

사		건	오	퍼	레이	션
생성	객체그룹		create_group()			
	서브객체그룹		create_subgroup()			
	연산객체		create_object()			
삭제	객체그룹		delete_group()			
	서브객체그룹		delete_subgroup()			
	연산객체		delete_object()			
활성화	객체그룹		enable_group()			
	서브객체그룹		enable_subgroup()			
	연산객체		enable_object()			
비활성화	객체그룹		disable_group()			
	서브객체그룹		disable_subgroup()			
	연산객체		disable_object()			
시간	객체구성요소		capture_time()			

3. 객체그룹과 모니터링 시스템간의 수행절차

객체그룹과 모니터링 시스템간의 수행절차는 1) 동일 객체그룹 내의 연산객체의 생성 및 삭제, 2) 동일 객체그룹 내의 연산객체의 활성화 및 비활성화, 3) 서브 객체그룹 내의 연산객체의 생성 및 삭제, 그리고 4) 서로 다른 객체그룹간의 연산객체의 생성 및 삭제 등 크게 네 가지로 구분할 수 있다. 3)과 4)의 연산객체의 활성화 및 비활성화 절차는 2)와 유사하므로 생략한다.

1) 동일 객체그룹 내 연산객체의 생성 및 삭제 절차

객체그룹(OG1) 내의 연산객체(CO-1)가 연산객체(CO-2)를 생성 및 삭제과정을 보이기 위한 원격 모니터링 시스템과 객체그룹간의 절차도는 그림 4와 같다.

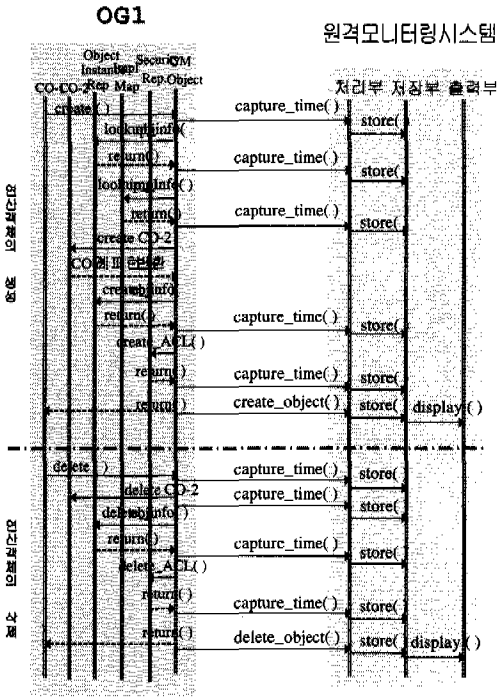


그림 4. 동일 객체그룹 내 연산 객체 생성/삭제 원격 모니터링 절차도

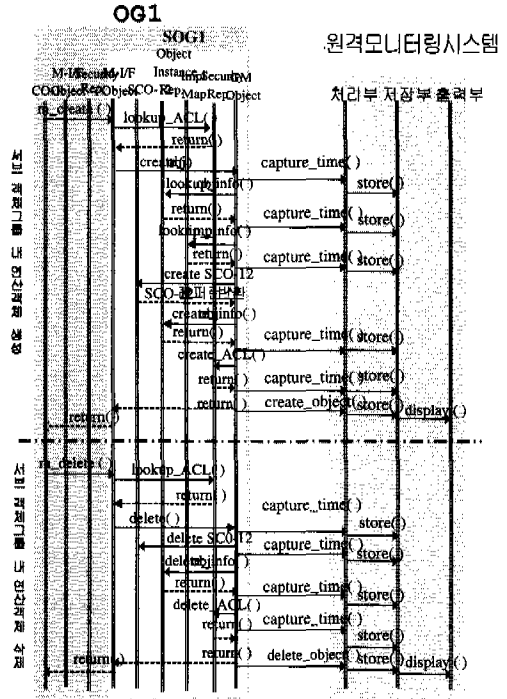


그림 6. 서로 다른 객체그룹 내 연산 객체 생성/삭제 원격 모니터링 절차도

2) 동일 객체그룹 내 연산객체의 활성화 및 비활성화 절차

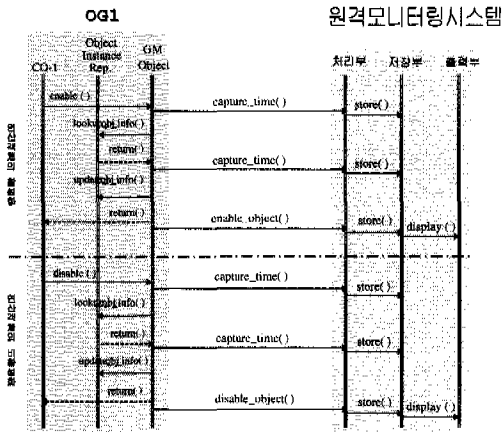


그림 5. 동일 객체그룹 내 연산 객체 활성화/비활성화 원격 모니터링 절차도

객체그룹(OG1) 내의 연산객체(CO-1)가 자기 자신을 활성화 및 비활성화 시키는 과정을 보이기 위한 원격 모니터링 시스템과 객체그룹간의 절차도는 그림 5와 같다.

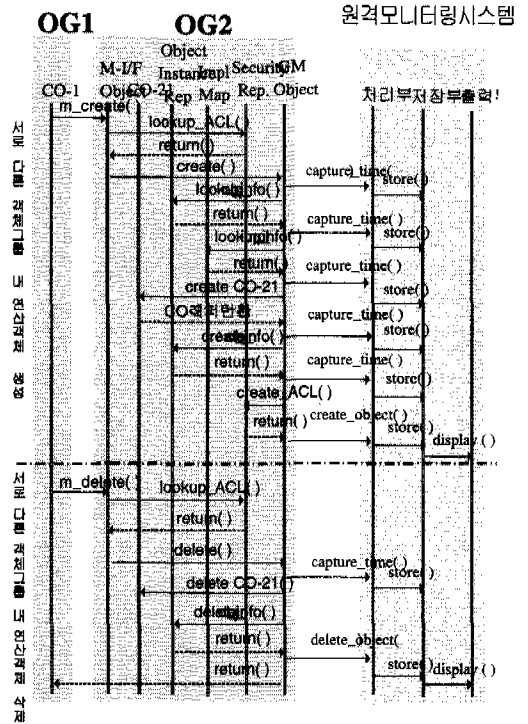


그림 7. 서로 다른 객체그룹 내 연산 객체 생성/삭제 원격 모니터링 절차도

3) 서버 객체그룹 내 연산객체의 생성 및 삭제

객체그룹(OG1) 내의 연산객체(CO-1)가 서버 객체그룹(SOG1) 안에 있는 연산객체(SCO-12)를 생성 및 삭제하는 과정을 보이기 위한 원격 모니터링 시스템과 객체그룹간의 절차도는 그림 6과 같다.

4) 서로 다른 객체그룹 내 연산객체의 생성 및 삭제

객체그룹(OG1) 내의 연산객체(CO-1)가 또 다른 객체그룹(OG2) 내의 연산객체(CO-21)를 생성 및 삭제 과정을 보이기 위한 원격 모니터링 시스템과 객체그룹들간의 절차도는 그림 7과 같다.

IV. 분산 객체그룹 지원 원격 모니터링 시스템 수행

본 장에서는 객체그룹 원격 모니터링 시스템의 구현을 위한 분산 환경을 구성하고, 객체지향 미들웨어로서 CORBA를 채택하여 분산 시스템들간의 객체 또는 객체그룹들간의 접속기능들을 제공하도록 IDL (Interface Definition Language)를 이용하여 해당 클래스의 명세를 작성하였다. 마지막으로 구현된 본 시스템을 통한 모니터링 수행의 예를 보였다.

1. 구현 환경

객체그룹 원격 모니터링 시스템의 구현 환경은 그림 8에서 나타내고 있다. 본 논문에서 구현한 원격 모니터링 시스템 환경은 객체그룹 모듈들을 가지는 하나 이상의 시스템들과 이들의 정보를 모니터링 하는 서버 시스템으로 구성하였다. 각 객체그룹을 관리 하는 그룹 관리자 객체로부터 객체그룹의 상태 변화에 대한 정보를 얻어 객체그룹 원격 모니터링 시스템의 모니터링 처리부에서 처리된 모니터링 정보들을 스크린상의 GUI 환경을 통해 보여줄 수 있도록 하였다. 객체그룹 원격 모니터링 시스템의 환경은 Windows NT 4.0, 객체그룹 모듈들을 가지는 시스템의 환경은 Windows 95/98 운영체제를 기반으로 하였다. 또한, 이들 시스템들간의 객체/객체그룹간의 접속을 위해 CORBA 사양을 따르는 Orbix 2.3을 미들웨어로 두었고, 객체그룹 원격 모니터링 시스템은 GUI 및 수행 모듈들은 Visual C++ 5.0으로 구현하였다.

2. 객체그룹 원격 모니터링 시스템을 위한 IDL 분산 객체그룹의 수행과정을 모니터링을 하기 위

한 객체그룹 원격 모니터링 시스템을 위한 IDL은 다음 그림 9와 같다.

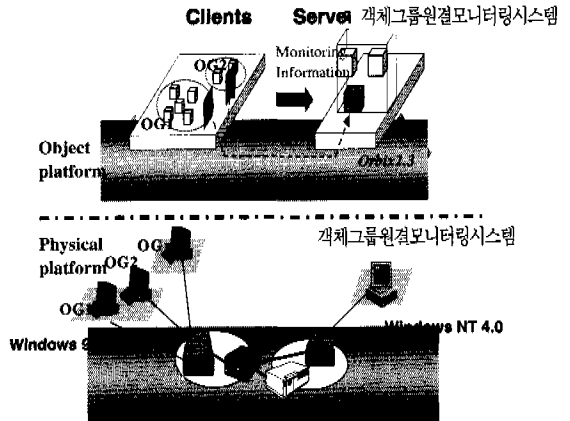


그림 8. 객체그룹 원격 모니터링 시스템 전체 구성도

```

//*****
// Name : Monitoring.IDL
// Version : Orbix 2.3C01
// Description : Object group modeling verification
//               & validation
//*****

interface monitoring {
    /** 시간 정보 검증 */
    void capture_time( in string feature );

    /** 생성에 대한 정보 */
    void create_group( in string grp_name );
    void create_subgroup( in string super_grp_name, in
        string grp_name );
    void create_object( in string super_grp_name, in
        string grp_name, in string self_name);

    /** 삭제에 대한 정보 */
    void delete_group( in string grp_name );
    void delete_subgroup( in string super_grp_name, in
        string grp_name );
    void delete_object( in string super_grp_name, in
        string grp_name, in string self_name);

    /** 활성화에 대한 정보 */
    void enable_group( in string grp_name );
    void enable_subgroup( in string super_grp_name, in
        string grp_name );
    void enable_object( in string super_grp_name, in
        string grp_name, in string self_name);

    /** 비활성화에 대한 정보 */
    void disable_group( in string grp_name );
    void disable_subgroup( in string super_grp_name, in
        string grp_name );
    void disable_object( in string super_grp_name, in
        string grp_name, in string self_name);
};
    
```

그림 9. 원격 모니터링 시스템 IDL 표현

3. 객체그룹 원격 모니터링 시스템의 수행 결과 객체그룹의 수행과정을 모니터링하기 위한 환경을

그림 10과 같이 설정하였으며, 이들 객체그룹들(OG1, OG2)은 분산된 두 시스템 상에 각각 두고, 모니터링 시스템은 별도의 서버 상에서 실행시켰다. 이 환경으로부터 객체그룹 내 객체들간의 서비스 및 관리 수행 시, 객체들의 상태변화, 객체간의 접속과정(순서) 및 접속시간을 분석하였다. 이들 수행과정으로 1) 그림 10의 ①은 동일 객체그룹 내에서 객체그룹 OG1 내의 연산객체 CO-1이 CO-2를 생성, 삭제, 활성화 및 비활성화 하는 과정이다.

그림 10의 ②는 객체그룹 OG1 내의 객체 CO-1이 서브 객체그룹 SOG1 내의 연산객체 SCO-12를 생성, 삭제하는 과정, 그리고 그림 10의 ③은 객체그룹 OG1 내의 연산객체 CO-1이 또 다른 객체그룹 OG2 내 연산객체 CO-21을 생성, 삭제하는 과정들과, 2) 그림 10의 ①에 대한 연산객체의 활성화와 비활성화 하는 과정을 각각 보인다. 이러한 과정들을 오퍼레이션들의 세부 수행절차는 3장 3절을 따른다.

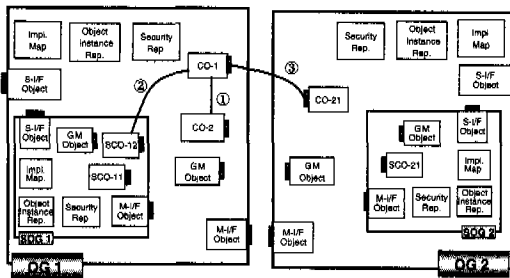


그림 10. 객체그룹간 수행과정 모니터링 환경

수행결과에 대한 모니터링 정보를 표현하기 위한 GUI 화면 구성에서 좌측부분은 수행되는 객체그룹들과 객체그룹내의 구성요소들의 실행된 상태들과, 우측 상단부분은 수행되는 객체들의 접속순서 및 접속시간을 나타내는 그래픽과 우측 하단 부분은 수행되는 객체들의 이름들이 접속되는 순서로 출력되도록 하였다.

- 1) 동일한 객체그룹 내에서 연산객체의 생성 및 삭제 (OG1 내 CO-1~OG1 내 CO-2, OG1 내 CO-1~OG1 내 SOG1의 SCO-12)

객체그룹 OG1 내에서 연산객체 CO-1이 연산객체 CO-2의 생성을 요청한 경우의 결과 화면은 그림 11과 CO-2의 삭제를 실행하는 결과 화면은 그림 12와 같다.

먼저 연산객체 CO-1이 CO-2를 생성하는 과정을 보면, OG1 내의 CO-1이 그룹 관리자 객체(OG1-GM)

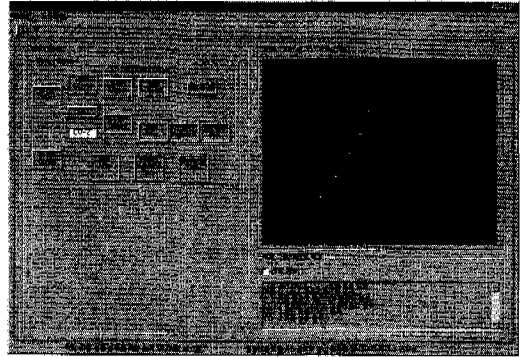


그림 11. 동일 객체그룹 내 연산 객체 생성

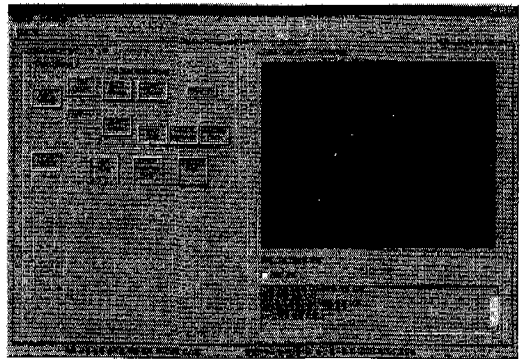


그림 12. 동일 객체그룹 내 연산 객체 삭제

에게 CO-2의 생성을 요청하면, OG1-GM은 구현 맵 객체(OG1-I-MAP : Impl. Map)에게 CO-2에 대한 정보를 얻은 후, 객체 인스턴스 레포지토리(OG1-O-I-REP : Object Instance Rep.) 상에 CO-2의 인스턴스에 대한 정보를 저장한다. 그러한 후, 보안 레포지터리(OG1-SEC-REP : Security Rep.)에 CO-2에 대한 접근 권한을 등록한 후, CO-2를 생성하게된다. 그림 11은 CO-2가 생성되었음을 보인 아이콘 화한 객체그룹의 구조와 CO-2의 생성까지의 과정을 그래픽과 텍스트로 보여주고 있다.

CO-1에 의한 CO-2의 삭제는 OG1-GM에게 CO-2의 삭제를 요구하고, OG1-O-I-REP 저장되어 있는 CO-2의 인스턴스에 대한 정보를 지운다. OG1-SEC-REP에서 CO-2에 대한 접근권한 규칙을 제거하고, CO-2를 삭제한다. 그림 12는 생성되었던 CO-2를 삭제한 객체그룹의 구조와 삭제과정을 보이고 있다.

또한, 동일 객체그룹 내 연산객체 CO-1이 서브객체그룹 SOG1 내의 연산객체 SCO-12를 생성과 삭제하는 과정은 그림 13과 그림 14에서 각각 보인다.

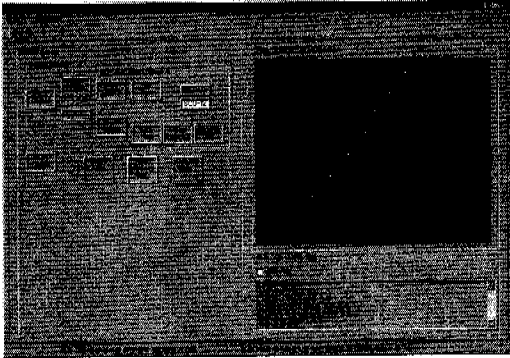


그림 13. 서버 객체그룹 내 연산 객체 생성

위에서 언급한 동일 객체그룹내의 연산객체의 생성 및 삭제과정과 동일하나, 그 차이는 연산객체 SCO-12를 생성 또는 삭제를 위해 CO-1이 SOG1-M-I/F에게 요청하여 SOG1-M-I/F로 하여금 SOG1-SEC-REP를 참조하여 SOG1 접근권한 받아 SOG1-GM에게 SCO-12의 생성 또는 삭제하는 과정이 다를 뿐이다. 이를 위해서 전제조건으로 OG1-SEC-REP로부터 SOG1의 접근권한에 대한 허가가 있어야 SOG1 내의 모든 객체를 접근할 수 있다. 서버 객체그룹은 객체그룹과 동일한 구조는 갖는 서버집합이므로, 객체그룹 내에서 연산객체의 생성 또는 삭제과정은 서버 객체그룹 내에서도 동일하게 이루어진다.

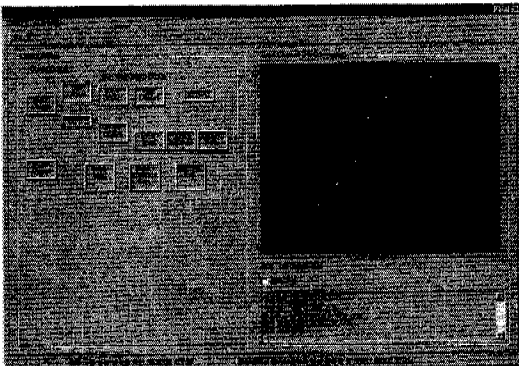


그림 14. 서버 객체그룹 내 연산 객체 삭제

2) 서로 다른 객체그룹들 내 연산객체의 생성 및 삭제(OG1 내 CO-1~ OG2 내 CO-2)
본 과정은 하나의 연산객체가 같은 객체그룹 내에 포함하는 서버 객체그룹 내에 새로운 연산객체를 생

성 또는 삭제하는 과정과 유사하다. 단지 차이는 단일 객체그룹이 아닌 서로 다른 객체그룹들간의 분산 수행을 통해서 생성과 삭제가 이루어진다는 것이다. 즉, 객체그룹들은 단일 시스템 또는 분산 환경에서 응용 서비스를 지원할 수 있어서 분산 객체그룹 모델의 기본 플랫폼이 될 수 있다.

1)의 동일 객체그룹 내에서 하나의 연산객체가 서버 객체그룹내의 연산객체를 생성과정과 비교하여보면, 연산객체 CO-21을 생성과정에서 CO-1은 SOG1-M-I/F 대신 OG2-M-I/F에게 새로운 연산객체의 생성을 요청하면, OG2-M-I/F는 OG2-SEC-REP를 참조하여 OG2의 접근허가를 받아 객체그룹 관리자 OG2-GM에게 CO-21의 생성을 요청한다. 이때, OG2-GM은 구현 맵 객체(OG2-I-MAP)로부터 CO-21에 대한 정보를 얻은 후, 객체 인스턴스 레포지토리(OG2-O-I-REP)에 CO-21의 인스턴스에 대한 정보를 저장한다.

마지막으로 보안 레포지터리(OG2-SEC-REP : Security Rep.)에 CO-21에 접근권한을 저장하고, CO-21을 생성하게된다. CO-21의 삭제는 CO-1이 OG2-M-I/F에게 CO-21의 삭제를 요구하면, OG2-M-I/F는 OG2-SEC-REP로 접근권한을 확인하여 OG2의 접근이 허용될 때, OG2-O-I-REP로부터 CO-21의 인스턴스에 대한 정보와 OG2-SEC-REP로부터 CO-21의 접근권한 규칙을 제거한다. 마지막으로 CO-21을 삭제한다. 그림 15는 OG1 내 CO-1이 OG2 내 CO-21의 생성하고 삭제를 수행하는 결과 화면을 보인다.

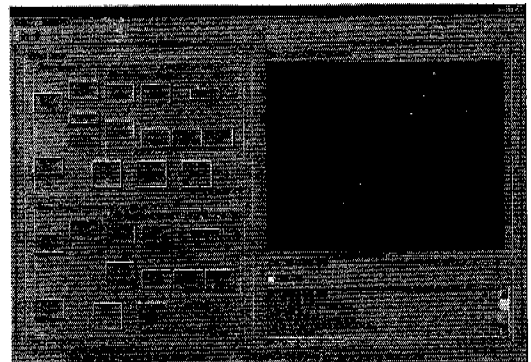


그림 15. 서로 다른 객체그룹 내 연산 객체 생성/삭제

3) 동일한 객체그룹 내 연산객체의 활성화 및 비활성화 (OG1 내 CO-1~OG1 내 CO-2)
OG1 내 CO-1이 동일 객체그룹 내 CO-2의 서비스

를 요청하기 위해서 활성화 및 비활성화 시키는 과정의 결과 화면은 그림 16과 그림 17에서 각각 보인다.

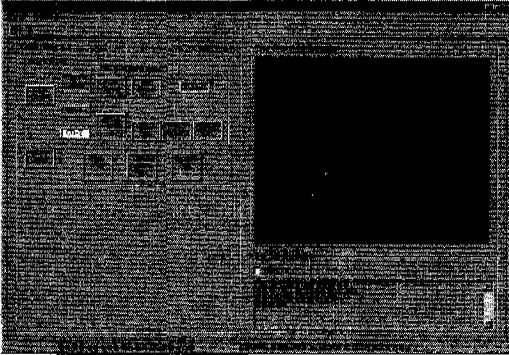


그림 16. 동일 객체그룹 내 연산 객체 활성화

이 과정은 생성된 CO-2의 인스턴스는 객체 인스턴스 레포지토리(OG1-O-I-REP) 상에 저장되어 있어야만 활성화 및 비활성화가 가능하며, 생성과정을 거친 연산객체는 디폴트로 활성화 상태이다. CO-1이 OG1-GM에게 CO-2의 활성화 또는 비활성화를 요청한다. OG1-GM은 OG1-O-I-REP에게 CO-2의 인스턴스에 대한 정보의 존재 여부를 확인하게 된다. 만약 여기에 등록되어 있다면(CO-2 연산객체가 생성되어 있음), OG1-O-I-REP에 있는 CO-2의 상태필드를 활성화 또는 비활성화로 설정하므로써 CO-1은 CO-2의 서비스를 지원 받을 수 있게된다. 서로 다른 객체그룹 내에서 연산객체들의 활성화 또는 비활성화 과정도 OG2-M-I/F로 하여금 OG2-SEC-REP를 참조하여 OG2 접근권한을 받아 OG2-GM까지의 접속순서를 제외하면, 위 과정과 유사하므로 반복된 설명을 피한다.

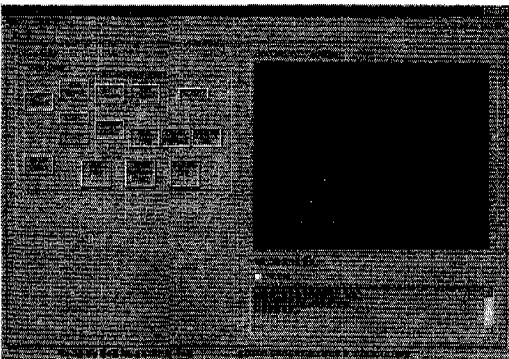


그림 17. 동일 객체그룹 내 연산 객체 비활성화

4) 모니터링 정보의 수행결과 검토

본 객체그룹을 지원하는 원격 모니터링 시스템은 객체그룹내의 구성요소들인 객체들의 수행중의 상태, 객체들간의 정보전달 순서 및 접속시간들을 GUI 환경에서 시각화한 아이콘, 텍스트 및 그래픽 형태로 보여주고 있다. 이들 모니터링 된 정보를 이용하여 분산 객체그룹 시스템의 수행 시, 수행오류를 발견하고 시스템 성능에 영향을 주는 요인들(접속순서, 접속시간)을 분석하여 성능 향상 방안을 찾고자 하는데 그 의의를 둔다. 이를 위해 본 모니터링 시스템의 실행과정을 3가지 형태로 분류해서 보였다. 이들 각각에 대한 결과를 검토한다.

1) 동일한 객체그룹 내에서 연산객체의 생성 및 삭제과정으로 첫째, OG1의 CO-1과 OG1의 CO-2간의 관계를 보면(그림 11, 12), 연산객체의 생성과 삭제에 대한 각각의 총 시간은 20ms와 27ms를 나타내고 있다. 삭제시간이 생성시간보다 오래 걸리는 이유는 OG1-O-I-REP 저장되어 있는 CO-2의 인스턴스에 대한 정보와 OG1-SEC-REP 내의 CO-2에 대한 접근권한 규칙을 제거하는 시간이 생성 시 이들 REP들을 참조하는 시간보다 더 걸리기 때문임을 알 수 있다. 둘째로, OG1의 CO-1과 OG1 내 SOG1의 SCO-12간의 관계를 보면(그림 13, 14), 이때 연산객체 SCO-12의 생성에 대한 총 소요시간은 31ms이고, 삭제 시 총 소요시간은 35ms가 걸림을 알 수 있다. 이 시간은 전자의 경우보다 시간이 소요시간이 긴 이유는 서브 객체그룹내의 연산객체를 접속하기 위해 접근권한의 받기 과정으로 SOG1-M-I/F, SOG1-MAP, SOG1-SEC-REP(생성 시: 참고 후, 등록, 삭제 시 : 참고 후, 삭제) 객체들을 추가적으로 접속해야 하기 때문이다. 전자의 경우처럼 서브 객체그룹이 없는 객체그룹 내의 연산객체들의 생성 및 삭제시간보다 시간이 더 걸렸음을 알 수 있다. 그러나 본 객체그룹 모델의 개발 목적에 따라 많은 객체들의 수와 그들간에 복잡한 인터페이스들을 객체들간의 관리 및 서비스 관점에서 이는 사소한 문제이다.

2) 서로 다른 객체그룹들간에 연산객체의 생성 및 삭제 과정으로서 OG1 내 CO-1과 OG2 내 CO-21간의 관계를 보면(그림 15), OG1 내 CO-1과 OG2 내 CO-21을 생성시키는 데 걸리는 총 소요 시간은 35ms이고, 삭제과정은 39ms가 걸린다. 이 과정은 1)의 두 번째 방법과 비교하여 4ms의 시간 지연이 발생됨을 알 수 있다. 이는 동일 객체그룹 내에서의 연산객체의 생성이나 삭제 과정과 달리 다른 객체그룹

을 찾는 시간비용이 요구되기 때문이다. 측정된 시간은 이들 객체그룹들은 단일 시스템 상에서 수행시킨 결과이다. 만약, 이들 객체그룹들이 각각 서로 다른 시스템 내에서 실행되는 경우를 고려한다면 시스템 간의 데이터 전송비용이 추가되어야 할 것이다.

3) 마지막으로 동일 객체그룹내의 연산객체의 활성화 및 비활성화 과정을 보면(그림 16, 17), 단순히 OGI-GM을 통해 OGI-O-I-REP에게 CO-2의 인스턴스에 대한 정보의 존재 여부를 확인한 후, CO-2의 상태필드에 활성화(서비스 수행 가능한 대기상태) 또는 비활성화로 설정하기만 하면 된다. 이때 연산객체의 활성화와 비활성화 과정에 대한 총 소요시간은 각각 13ms와 10ms가 걸린다. 활성화 과정의 시간비용이 많은 이유는 CO-2 연산객체의 레퍼런스를 찾는 비용이 추가되기 때문이다. 비활성화 과정은 활성화 시 얻은 CO-2의 레퍼런스를 그대로 사용한다. 동일 객체그룹 내의 서브 객체그룹 안에 있는 연산객체나 서로 다른 객체그룹 내 연산객체의 활성화와 비활성화 과정은 위 수행형태와 유사하므로 생략한다.

V. 결론

분산 컴퓨팅 환경은 객체지향 기반의 초고속 통신망을 통한 개방형 정보통신망 구조로 변화되고 있다. 개방형 정보통신망 구조 기반의 분산처리 환경에서 제공되는 분산 소프트웨어는 객체단위와 객체그룹 단위로 기능이 수행되도록 요구됨으로써 다양한 멀티미디어 서비스를 지원할 수 있다.

객체그룹을 효율적으로 관리하기 위해서는 객체그룹 내의 구성요소들간의 관계를 통한 분산 객체그룹 간에 서비스 수행에 따른 객체그룹 구성 객체들 사이의 정보 흐름을 파악할 수 있는 모니터링 시스템이 요구된다.

이러한 요구를 만족시키기 위해서 본 논문에서는 동일 객체그룹 및 분산 객체그룹간에서 관리 접속에 따라 발생하는 객체들의 상태 변화, 접속 관계, 접속 시간의 정보를 파악할 수 있는 객체그룹 원격 모니터링 시스템을 설계 및 구현함으로써 객체그룹을 효율적으로 관리하고 시스템의 성능 향상 방안을 강구할 수 있다.

객체그룹 원격 모니터링 시스템은 그룹 관리자 객체로부터 생성, 삭제, 활성화, 비활성화에 의한 구성 객체들간의 접속에 의한 상태 변화 및 접속시간 등의 정보를 전송 받아 처리부를 거쳐 모니터링 정보

를 분석 및 처리한 후 저장부에 저장하고 출력부를 통하여 GUI 환경으로 화면상에 시각적인 아이콘, 텍스트 및 그래픽 형태로 실시간으로 출력하도록 하였다. 이렇게 함으로서 본 시스템은 객체그룹의 관리 접속에 의한 객체들의 상태 변화와 접속과정 및 접속시간들의 모니터링 정보를 참조하여 분산 객체그룹 시스템을 위한 성능향상의 쟁점들을 강구하는 목적으로 사용될 것이다.

현재 본 시스템은 시스템 레벨이 아닌 응용 레벨인 객체 플랫폼 상에서 수행되는 과정을 모니터링하고 있다. 추후 현 객체그룹 모델과 연계하여 CORBA와 운영체제간의 인터페이스 및 성능의 향상 요인 분석을 위한 시스템 모니터링 연구 및 멀티미디어 스트림 서비스를 위한 세션 관리 및 QoS 모니터링 연구를 진행할 것이다.

참고 문헌

- [1] Christensen, E.Colban, "Information Modelling Concepts", DRAFT, TB-EAC. 001-1. 1-93, TINA-C, November 1994.
- [2] N.Natarajan, F.Dupuy, N.Singer, H.Christensen, "Computatioanl Modelling Concepts", TB-A2. HC. 012-1.2-94, TINA-C, February 1995.
- [3] M.Kudela, K.MacKinnon, "Engineering Modelling Concepts(DPE Kernel Specificatons)", TR-KMK.001-1.1-94, TINA-C, November, 1994
- [4] Silvano Maffeis, "The Object Group Design Pattern", Dept. of Computer Science, Cornell University, 1996. 2.
- [5] Yigal Hoffner, "Monitoring in Distributed System", ANSA Document 3, Specification, 1994. 10.25.
- [6] James Runbaugh et al. "Object-Oriented Modeling & Design", 1991.
- [7] Bersia, Bosco, Monione, moiso, and Spinolo, "A case environment for TINA-oriented application", CSELT, 1994.
- [8] IONA Technologies Ltd, "The Orbix Architecture," 1995.

[9] Tom Handegard, "Object Grouping and Configuration Management(Ver 1.1.)", TINA-Document, 1996. 12.

[10] Jon Siegel, "CORBA Fundamentals and Programming" Jone WILEY & Sons, Inc, 1995

[11] OMG, "CORBA Services : Common Object Services Specification", 1997

[12] 주수중, 신영석, 이승용, "분산처리환경에서 객체그룹 모델링 및 성능분석에 관한 연구", 최종 연구보고서, 한국전자통신연구원, 1997. 11.

[13] 김진영, 이진엽, 김명희, 주수중, "분산 객체그룹 모니터링 시스템 설계", 한국정보처리학회, 추계 학술대회 논문집, 제4권 제2호, pp 1086-1090, 1997

[14] 이승용, 정창원, 신영석, 주수중, "개방형 분산 환경에서 객체그룹 모델의 설계", 한국통신학회 논문지, 제 23권 9호, 1998. 9. pp 2258-2270.

[15] 김명희, 신영석, 정창원, 신경민, 주수중, "분산 처리환경에서 멀티미디어 서비스를 위한 객체그룹 모델의 설계 및 구현", 한국정보처리학회 학술지, 1998. 4. CD-ROM file://FI/Proceeding/non13/non12 .html.

이 원 중(Won-Jung Lee)

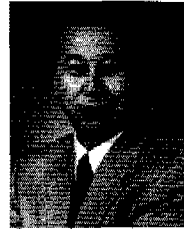


1991년 2월 : 전북대학교 전산
· 통계학과 졸업
1998년 2월 : 원광대학교 교육대
학원 컴퓨터공학과 석사
1998년 3월~현재 : 원광대학교
컴퓨터공학과 박사과정

<주관심 분야> 분산객체모델, 멀티미디어 데이터 베이스

주 수 중(Su-Chong Joo)

정회원



1986년 2월 : 원광대학교
전자계산공학과 졸업
1988년 2월 : 중앙대학교
컴퓨터공학 석사
1992년 3월 : 중앙대학교
컴퓨터공학 박사

1993년 : 미국 Univ of Massachusetts at Amherst
전기 및 컴퓨터공학과 Post-Doc.

1990년~현재 : 원광대학교 컴퓨터공학과 교수

<주관심 분야> 분산 실시간 컴퓨팅, 분산객체모델, 시스템 최적화, 멀티미디어 데이터 베이스, 지리정보 시스템(GIS)

이 건 엽(Geon-youb Lee)

정회원



1995년 6월 : 군산대학교 정보통신
공학과 졸업
1997년 2월 : 군산대학교 정보통신
공학과 석사
1997년 3월~현재 : 원광대학교
컴퓨터공학과 박사과정

1997년 9월~현재 : 군장대학 전임강사

<주관심 분야> 분산시스템, 분산객체모델, 멀티
미디어 통신