

분산 객체 컴퓨팅 환경에서 실시간 객체그룹을 이용한 객체 관리 및 서비스 연구

김명희^U 신창선 주수중
원광대학교 컴퓨터공학과
{hee, scjoo}@wonkwang.ac.kr
csshin@gaebiyok.wonkwang.ac.kr

A Study of Object's Managements and Services Based on Real-Time Object-Group in Distributed Object Computing Environments

Myung-Hee Kim^U Chang-Sun Shin Su-Chong Joo
Dept. of Computer Engineering, Wonkwang University

요 약

본 논문은 분산환경에서 객체들의 관리를 용이하게 하고, 실시간 서비스를 지원할 수 있는 실시간 객체그룹의 플랫폼을 제시한다. 기존에 이루어지고 있는 실시간 분산 환경은 표준 CORBA를 사용하지 않고 ORB를 수정하거나 확장하는 방법으로 새로운 실시간 CORBA 환경을 만들고 있다. 따라서, 본 논문은 ORB를 수정하지 않고 표준 CORBA라는 환경아래에서 실시간을 지원할 수 있는 방안을 제시한다.

실시간 객체그룹은 객체의 관리에 대한 그룹관리자와 실시간 처리 즉, 서비스에 대한 실시간관리자를 각각 두어 분산 객체들의 효율적인 관리와 실시간 특성에 대한 클라이언트와 서버의 투명성을 제공한다. 이를 위해, 본 연구에서는 실시간정보라는 구조체를 정의하여 실시간 파라미터등을 처리하였고, 타이머객체를 두어 클라이언트와 서버 각각의 마감시간 확인을 용이토록 했다. 또한 스케줄러객체를 따로 두어, 본 연구의 플랫폼을 사용하는 시스템의 상황에 맞는 스케줄링 알고리즘을 선택하여 유동적으로 사용할 수 있도록 하였다.

1. 서론

최근에, 분산 컴퓨팅은 OSF의 DCE나 OMG의 CORBA와 같은 분산 객체 컴퓨팅 환경에 관심을 갖는 추세이다. 분산 객체 컴퓨팅은 이질적인 클라이언트와 서버들간의 상호작용성을 요구하는 어플리케이션을 위한 프로그래밍 패러다임이 되고 있다.

OMG는 CORBA를 그러한 분산 환경에 대한 표준 소프트웨어 규정(specification)으로 개발했다. 이 표준은 분산 컴포넌트(component)들의 기능적 행동(functional behavior)에 대한 인터페이스의 기술을 위해 IDL을 나타낸다. 표준은 또한 객체 서비스(즉, Naming, Event 등)와 분산된 클라이언트 객체와 서버 객체간의 상호작용을 고려한 미들웨어인 Object Request Broker(ORB)를 기술한다.

공장 자동화 제어, 항공분야와 같은 많은 분산 실시간 어플리케이션은 CORBA와 같은 분산 아키텍처로부터 이득을 얻을 수 있다. 많은 이들 어플리케이션의 설계자들은 그들의 아키텍처를 위해 CORBA를 고려했다. 그러나 현재 실시간 요구사항을 지원하기가 부적절하다는 것을 발견했다. 예를들면, IDL은 분산 컴포넌트들의 기능적 행동에 대한 인터페이스를 기술한다. 그러나 그들의 행동에 대한 시간 제약조건을 명확하게 기술할 수가 없다. 더 나아가, 분산 환경에 의해 제공되는 시스템 서비스들은 환경 전반에 걸친 중단간 실시간 스케줄링을 위한

지원을 거의 제공하고 있지 않다. 실제로 일부 환경들은 클럭 동기화와 같은 기본적인 서비스들도 제공하지 않고 있다.

최근에, SIG(Special Interest group)가 현재 CORBA 표준을 검사하고 실시간 어플리케이션을 지원하기 위한 요구사항들을 정의할 목적으로 OMG내에 설립되어졌다. 특히, 실시간 SIG는 현재 CORBA 표준을 확장(CORBA/RT)함으로써 시간 제약조건을 표현하고 실행할수 있는 능력을 지원하는데 초점을 맞추고 있다. SIG는 실시간을 지원하기 위한 분산 객체 컴퓨팅 환경을 위해 요구되었던 사항들을 자세히 다루는 백서(white paper)[1]를 내놓았다. 이에 따라 세계적으로 많은 연구들이 이루어지고 있으나, 여기에는 CORBA의 핵심이 되는 ORB를 수정하거나 새롭게 확장하여 실시간을 지원하도록 하는 실시간 CORBA[2]를 제시하고 있다. 따라서, 본 연구는 기존의 표준 CORBA[3]상에서 ORB를 수정하지 않고 실시간을 지원할 수 있는 구조에 초점을 맞춘다. 또한, 분산된 객체들의 효율적이고 체계적인 관리와 실시간 지원을 위해 개별 서비스 객체들에게 과중되는 관리 절차를 간편하게 하기 위해 객체그룹[6,7]의 개념을 도입한다. 이렇게 함으로써, 본 연구는 기존의 표준 CORBA에서 분산 객체들의 관리와 분산 실시간 서비스[4]를 지원할 수 있는 실시간 지원 객체그룹 모델을 제시하며, 이에 따른 요구사항과 기능들을 정립하며, 관리와 서비스측면에서 객체들간의 상호작용의 절차를 기술한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에서 고려해야 할 사항에 초점을 맞추어 실시간 CORBA에 대한 기술을 하고, 3장에서는 본 연구에서 제시하는 객체그룹의 구조와 기능들을 정립하여 그들간의 관리와 서비스에 따른 접속 과정을 보여주며, 마지막으로 결론과 향후 연구과제를 기술한다.

2. 실시간 CORBA

실시간 CORBA 시스템이란 CORBA 시스템에서 종단간 서비스를 실행하는데 실시간 조건을 만족하는 시스템 즉, 표준 CORBA에 실시간 개념을 포함한 것을 의미한다. 여기에서 고려되어야 할 실시간 개념은 시간관리, QoS, 스케줄링, 동시성, 고장허용, 성능등이며, 실시간 기술로는 CPU스케줄링과 디스패칭 기술, 실시간 통신 프로토콜, 실시간 시스템에서 결합허용 소프트웨어의 구현, QoS 관리, 클럭 동기화등이다.

본 연구에서는 RT/SIG에서 고려한 실시간 요구 사항들의 세 가지 영역 즉, 오퍼레이팅 환경을 위한 요구사항, ORB 아키텍처를 위한 요구사항, 객체 서비스[5]와 기능을 위한 요구사항 중에서 시간 제약조건을 표현하고 지원하는 객체 서비스와 기능을 위한 요구사항과 ORB 아키텍처에 중점을 두고 연구한다. 이들 요구사항은 분산 컴퓨팅 환경에 포함되어져야 하는 CORBA의 메소드 호출, 전역 시간 서비스와 실시간 스케줄링과 우선순위 결정 방법등의 분석이 포함된다. 또한, 기존의 ORB를 수정하지 않고 실시간 메소드 호출이 가능하도록 시간 제약조건의 표현 형식에 대한 연구도 포함한다.

DCE와 CORBA와 같은 대부분의 분산 컴퓨팅 환경은 객체간의 연결을 위해 RPC(Remote Procedure Call) 파라다임을 가진 클라이언트/서버 모델을 사용하기 때문에, 이러한 클라이언트/서버 모델에 시간성을 제공할 수 있도록 한다. RPC상에 시간성을 제공하여 실시간을 지원할 수 있도록 한다는 것을 CORBA를 기반으로 하여 바꾸어보면, CORBA는 객체 모델을 기본으로 하므로 단순한 프로시저 호출이 아닌 시간성을 지원하는 메소드 호출(method invocation)[4]로 정의한다.

3. 객체그룹

이 장에서는, 분산객체들의 실시간 서비스 수행이 가능하며, 효율적인 관리를 지원하는 객체그룹 모델의 구조를 제시하고, 구성요소들에 대한 기능을 정립한 후, 관리 및 서비스 수행에 따른 구성요소들간의 접속 과정을 제시한다.

3.1 객체그룹의 구조

본 연구의 객체들은 실시간 서비스를 수행하기 위해, 실시간 특성을 가지고 있다. 다시 말해서, 객체에 대한 서비스 요청은 단순한 메소드 요청이 아닌, 시간 제약조건(예를들면, 마감시간)등을 포함한 메소드 요청으로 이루어진다. 앞서 말한바와 같이 이러한 시간 제약조건을 표준 CORBA상에 ORB의 수정없이 분산된 서비스 객체에게 전달하는 것이 불가능하므로, 본 연구에서는 이를 가능토록하는 객체그룹 모델을 제시하는데 초점을 맞춘다. (그림 1)은 본 연구에서 제시한 객체그룹의 구조이다.

본 객체그룹은 그룹관리자(GM), 실시간관리자(RM), 팩토리 객체(Factory), 스케줄러객체(Scheduler), 타이머객체(Timer), 객체정보레포지토리(Object Info.), 보안객체(Security),

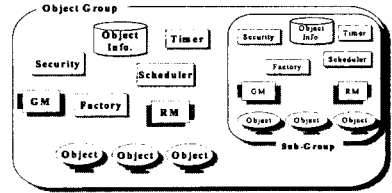


그림 1. 객체그룹의 구조

서비스객체(Object)들과 서브객체그룹(Sub-Group)으로 구성된다. 먼저 그룹관리자는 객체그룹 관리의 책임을 가지므로, 모든 관리 요청과 전달이 그룹관리자를 통해서 이루어지며 그에 따라 보안객체, 팩토리객체등과 상호 작용한다. 보안객체는 객체그룹 외부에서 객체그룹내의 객체나 서브객체그룹의 객체에 대한 접근권한여부를 검사한다. 팩토리객체는 그룹관리자의 요청에 따라 객체정보레포지토리에 있는 소속객체들의 정보를 이용하여 객체나 서브객체그룹을 생성하고, 그들의 레퍼런스를 반환한다. 실시간관리자는 객체에게 실시간 특성을 부가하여, 이에 따른 처리를 위해 스케줄러객체, 타이머객체와 상호 작용한다. 스케줄러객체는 실시간관리자의 요청에 따라 마감시간을 이용하여 서비스요청객체에 대한 우선순위를 스케줄링한다. 타이머객체는 현재 수행되는 서비스의 마감시간을 설정하거나 해제한다. 이를 이용하여 마감시간 위배사실을 알 수 있다. 또한, 서브객체그룹은 객체관리에 대한 편의성을 더욱 증가시키며, 본 연구는 1단계의 서브객체그룹을 전제로 한다.

3.2 기능정립

본 절에서는 객체그룹 구성요소들 중 실시간 처리의 핵심이 되는 실시간관리자의 기능을 정립한다.

먼저 실시간관리자는 서비스를 요청하는 객체가 소속되어있는 그룹내에 있는 실시간관리자와 서비스를 수행하는 객체가 소속되어있는 그룹내에 있는 실시간관리자의 기능이 구별된다. 서비스 요청그룹내의 실시간관리자는 서비스 요청객체의 마감시간에 따라 타이머를 설정하여, 요청이 마감시간을 위배하는지 확인하며, 현재 전역시간과 마감시간 정보를 실시간정보(RI)라는 구조체(Structure type)에 저장하여 전달 파라미터로써 사용될 수 있도록 한다. 서비스 수행그룹내의 실시간관리자는 전달된 파라미터의 실시간정보에서 마감시간 정보를 추출하여 새로운 마감시간을 계산하며, 스케줄러 객체에게 서비스 요청객체의 우선순위를 스케줄링하도록 요청하고, 스케줄러 객체의 반환값을 토대로 해당 서비스 요청객체의 마감시간을 타이머에 설정하고, 서비스 수행객체에게 클라이언트의 정보를 넘겨준다.

```
interface RM
Start_Client_RM( ); //타이머 설정, 마감시간과 현재전역시간을 실시간정보(RI)에 저장//
End_Client_RM( ); //타이머 해제//
Start_Server_RM( ); //마감시간 계산, 스케줄링 요구, 타이머 설정//
End_Server_RM( ); //타이머 해제//
}
```

3.3 관리 및 서비스 접속 절차

본 절에서는, 실시간 객체그룹 플랫폼상에 클라이언트가 실시간 서비스를 수행하기 위해 필요한 관리적인 절차과정과 실

질적인 서비스를 요청하는 서비스 절차과정을 나타낸다. 객체 그룹의 객체 관리에 필요한 기능들은 객체 그룹내의 객체들과 서브객체들의 생성, 삭제, 활성화, 비활성화등이 있다. 이때 이러한 기능들은 그룹관리자의 책임하에 이루어진다. 반면에, 객체에 대한 실시간 서비스 수행요청에 관련된 모든 책임은 실시간 관리자가 맡게 된다. (그림 2)는 클라이언트 객체와의 관리적 접근절차를 나타낸다.

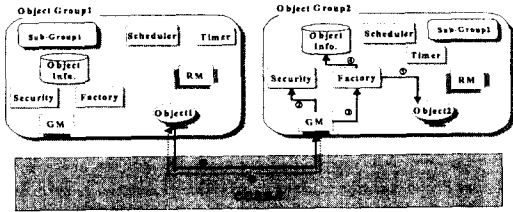


그림 2. 관리적 접근절차

(그림 2)에서 객체그룹1의 객체1(클라이언트라 부름)이 서비스 객체와의 연결을 설정하기 전에 내임 서비스를 통해 서비스를 수행하는 그룹(객체그룹2)의 그룹관리자와의 연결을 설정한다. 그룹관리자는 클라이언트에 대한 보안 검사를 보안 객체에 의뢰하고, 접근이 가능한 객체라면 그룹관리자는 팩토리 객체에게 서비스 객체(객체2)의 생성을 요구한다. 팩토리객체는 그룹에 소속되어 있는 객체의 정보를 이용하여, 서비스 객체를 생성한 후, 그룹관리자에게 서비스 객체의 레퍼런스를 반환한다. 그룹관리자는 반환 받은 레퍼런스를 클라이언트에게 다시 반환해준다. 지금까지의 일련의 과정이 객체1(클라이언트)이 객체2(서비스 객체)에게 서비스를 요청하기 위해 바인딩에 필요한 객체의 레퍼런스를 얻어오는 절차이다.

(그림 3)은 앞선 레퍼런스 획득의 과정을 거친 후, 객체 1이 객체 2에게 시간 제약조건을 표현하며 서비스 요청을 하는 접속절차를 나타낸다.

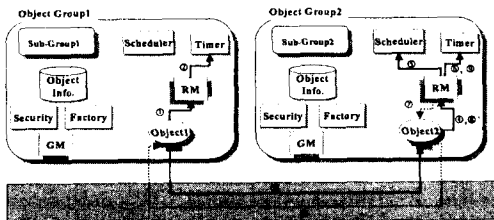


그림 3. 서비스 접속절차

(그림 3)에서 객체1(클라이언트라 부름)은 자신의 객체그룹(객체그룹1)내에 있는 실시간관리자(RM)에게 마감시간정보를 준다. 실시간관리자는 전역시간 서비스를 이용하여 계속 전역 시간정보를 유지하고 있으면서, 현재 전역 시간과 마감시간에 대한 상대값을 실시간정보(RI)에 저장한다. 실시간 관리자는 클라이언트가 준 마감시간에 따라 자신의 그룹에 있는 타이머를 설정한다. 클라이언트는 객체 2에게 서비스를 요청한다. 객체2는 클라이언트에게서 받는 실시간정보를 자신의 그룹(객체그룹2)에 있는 실시간관리자에게 전달한다. 실시간관리자는 실시간정보를 이용하여, 전송시간을 고려한 서비스수행 마감시간을 계산한다. 실시간관리자는 계산된 마감시간을 스케줄러 객체에게 넘겨주어, 클라이언트의 우

선순위를 스케줄링한다. 스케줄링의 우선순위에 따라 객체그룹 2의 타이머를 계산된 마감시간으로 설정한다. 실시간관리자는 객체2에게 우선순위에 따른 클라이언트의 정보를 반환하고, 객체2는 서비스 수행후, 클라이언트에게 결과값을 반환하고 실시간 관리자에게 서비스 수행이 끝남을 알린다. 실시간 관리자는 타이머 설정을 해제한다. (그림 4)는 (그림3)의 접근절차를 객체의 메소드 호출을 중심으로 예로써 나타낸다.

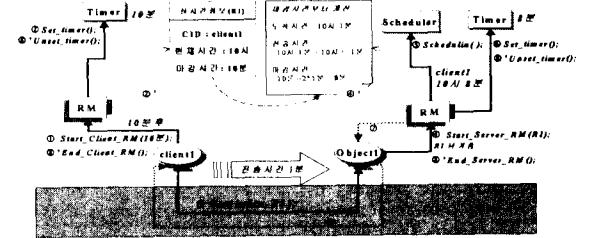


그림 4. 예로 통한 서비스 접속절차

4. 결론

본 연구는 표준 CORBA에 실시간 특성인 시간 제약조건등을 처리할 수 있는 객체그룹 플랫폼 구조를 제시한다. 이에 따라, 표준 CORBA환경에서 ORB의 수정 없이 시간 제약조건 의 표현과 처리가 간단해지며, 클라이언트나 서비스 객체에게 실시간 처리에 대한 투명성이 제공되며, 실시간 특성에 대한 정보 과라미터들을 확장 가능하여 유연성을 제공하고, 이러한 확장성이 클라이언트나 서비스 객체와 전혀 무관하게 이루어질 수 있다. 또한, 스케줄러 객체에 실시간 응용서비스의 특성에 맞는 스케줄링 알고리즘을 적용하여 구현 가능하게 한다.

추후 연구로는 설계된 객체그룹의 구현과 프로토타입에 대한 실시간 응용 서비스의 적용등이 필요하다.

5. 참고문헌

- [1] OMG Realtime Platform SIG, "Realtime CORBA A White Paper", http://www.omg.org/realtime/real-time_whitepaper.html
- [2] Victor Fay Wolfe, et al., "Real-Time CORBA", In proceedings of the third IEEE Real-time Technology and Applications Symposium, 1997.
- [3] OMG, "The Common Object Request Broker: Architecture and Specification 2.2", <http://www.omg.org/corba/corbaCB.htm>, 1998.
- [4] John. K. Black, et al. "Real-time Method Invocations in Distributed Environments", University of Rhode Island Department of Computer Science and Statics, Technical Reports TR95-244, 1995.
- [5] OMG, "CORBA Services: Common Object Services Specification", <http://www.omg.org/corba/sectran1.htm>, 1997.
- [6] 신경민, 김명희, 주수종, "CORBA환경에서 실시간 서비스 지원을 위한 분산 객체의 그룹화 및 관리", 한국정보처리학회 논문지, 6권 5호, 1999. 5
- [7] 김명희, 주수종, "CORBA환경에서 실시간 응용 지원을 위한 분산 객체그룹 플랫폼의 설계 및 구현", 한국정보처리학회 논문지, 7권 4호, 2000. 4