

---

# 다중노드 시스템에서 TMO를 이용한 실시간 서비스 메시지 보장

김광준\* · 서종주\* · 강기웅\* · 윤찬호\*\*

The Guarantee of Real Time Service Message with TMO in Multi-nodes Systems.

Gwang-Jum Kim\* · Jong-Joo Seo\* · Ki-Ung Kang\* · Chan-Ho Yoon\*\*

<요 약>

최근에 급성장하고 있는 실시간 통신 분산 컴퓨팅은 최근에 컴퓨터 응용분야의 하나로서 컴퓨터 과학과 공학 분야에서 급성장하고 있는 한 분야이다. 실시간 객체 지향 분산 컴퓨팅은 분산된 컴퓨터 시스템에서 객체 네트워크의 형태로 구성된 실시간 분산 컴퓨팅이다.

본 논문에서는 TMO 구조를 이용하여 실시간 통신 시뮬레이션 프로그래밍을 하기 위해 DHS(Distributed High-Precision Simulation)라는 응용 환경에 적용하였다. TMO 구조에서 사용되는 시간 구동 및 메시지 구동 구조는 충분한 서비스를 제공하기 위해 데드라인 시간을 보장함으로써 실시간 시뮬레이션 프로그래머의 노력을 줄일 수 있는 이점을 가지고 있다. 데드라인 시간 보장은 프로그래머의 설계 시간을 처음 객체부터 적용하여 보장함으로써 형성될 수 있다. 분산된 객체지향 실시간 통신 시스템에서 TMO 구조를 이용한 다단계 프로그래밍 설계를 함으로써 실시간 통신 프로그래밍을 유연하게 작성할 수 있다.

ABSTRACT

One of the computer application fields which started showing noticeable new growth trends in recent years is the real time communication distributed computing application field. Object-oriented(OO) real time(RT) distributed computing is a form of real-time distributed computing realized with a distributed computer system structured in the form of an object network.

In this paper, we describes the application environment as the DHS (distributed high-precision simulation) with TMO structure. The TMO scheme is aimed for enabling a great reduction of the designer's effort in guaranteeing timely service capabilities of distributed computing application systems. It has been formulated from the beginning with the objective of enabling design-time guaranteeing of timely action. In the real time simulation techniques based on TMO object modeling, we have observed several advantages to the TMO structuring scheme. TMO object modeling has a strong traceability between requirement specification and design, cost-effective high-coverage validation, autonomous subsystems, easy maintenance and flexible framework for requirement specification.

키워드

Real Time Communication, Real Time Object-oriented, TMO(Time-triggered and Message-triggered Object)

---

\* 전남대학교 컴퓨터공학과  
접수일자 : 2006. 00. 00

\*\* 조선대학교 컴퓨터공학과

## I. 서론

1990년대 초반부터 시작된 기존의 실시간 통신 컴퓨팅 시스템과 관련된 연구는 대개 태스크, 스케줄링이나 시간 모델에 관련된 것으로서, 시간성 검증에 관한 이론적인 연구에 치우쳐 실제 시스템 개발에는 큰 도움을 주지 못했다[1]. 또한 많은 실시간 통신 컴퓨팅 시스템들이 분산 네트워크 환경에서 태스크들 사이에서만 상호작용이 이루어지도록 설계되어 실시간 통신시스템의 다양한 요구사항을 충분히 반영시키지 못하고 있다[2][3].

실시간 통신 객체 모델에서는 기본 병행성 제약 사항을 두어 서로 공유하고 있는 객체에 대해 메시지 구동 메소드 실행 시간과 시간 구동 메소드의 실행 시간이 충돌하는 경우에 시간 구동 메소드에 우선권을 준다. 이는 시간 구동 메소드의 실행 시간을 보장함으로써 실시간 통신시스템 설계시 신뢰성을 보장할 수 있는 장점은 가지고 있으나 시간 구동 메소드보다 긴급한 메시지 구동 메소드의 모델링이 어려워지는 단점을 가지고 있다[4][5]. 그러므로 이러한 단점을 해결하기 위해 실시간 통신 특성에 따른 시간 구동 메소드와 서비스 메시지 구동 메소드가 기본 병행성 제약 사항을 따르지 않고 동시성을 만족하면서 객체에 유연하게 적용될 수 있는 새로운 실시간 통신 객체 모델이 필요하다.

이러한 문제점을 해결하기 위해 실시간 통신 시뮬레이터 클럭을 응용한 TMO 구조를 이용하여 제안된 통합된 모델에 시간 구동 메소드와 메시지 구동 메소드를 적용함으로써 객체지향 개념과 실시간 통신 특성들을 모델링하고 기존의 실시간 통신 객체 시스템이 가지고 있는 마감종료 시간 이내에 서비스 메시지를 제공함으로써 실시간성의 특성을 반영하고자 한다.

## II. 실시간 통신 객체 모델

### 2.1 분산된 객체 노드사이의 서비스 보장

분산된 객체 노드의 시뮬레이터 클럭에 시간 구동 개념을 적용한 접근방식은 다른 분산 시뮬레이션 접근 방식보다는 중요한 장점을 가지고 있다. 이는 호스트 노드들 사이의 메시지 교환을 필요로 하지 않

는다는 가정하에서 분산된 시간 구동 시뮬레이션 객체가 각각의 분산된 시뮬레이터 클럭 객체에 의해 실행된 시뮬레이션 과정들이 모두 동기화 되어 있기 때문이다. 이러한 장점은 분산된 실시간 통신 시뮬레이션 환경에서 각각의 실시간 통신 객체 노드에 부여된 과중한 부하를 줄일 수 있다. 그러나 분산된 객체 노드에 시간 구동을 적용한 시뮬레이션 접근 방식은 짧은 시간 간격 동안에 시간 구동과 메시지 구동에 의해 생성되는 시뮬레이션 결과인 메시지 교환에 대한 보장이 없다는 것이다. 그러므로 짧은 시간 동안에 메시지의 교환을 보장하기 위해서 시뮬레이션 시간 간격을 충분히 크게 함으로서 이러한 문제점을 해결할 수 있다.

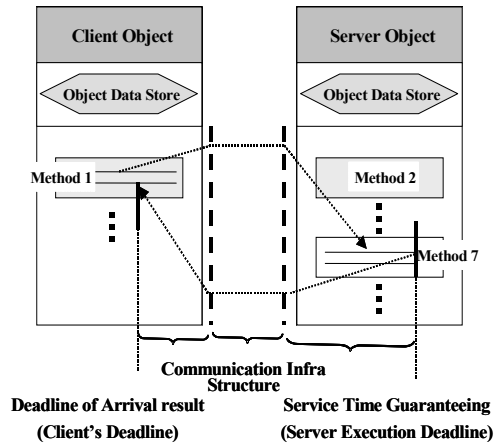


그림 2-1 클라이언트 객체의 데드라인에 대한 서버객체의 서비스 시간 보장  
 Fig. 2-1 Service time guarantee of Client object deadline vs server object

그림 2-1은 분산된 컴퓨팅 객체로 구성된 실시간 통신 요소를 나타내는 것으로서 시스템을 구성하고 있는 클라이언트와 서버(Server) 객체 사이의 관계를 나타내고 있다. 그림에서 나타낸바와 같이 클라이언트 객체 노드의 중간에서 실행되고 있는 Method 1은 서버객체의 Method 7을 통해 서비스가 호출되며, 호출된 서버객체의 서비스 실행은 임의의 짧은 시간 간격 동안에 이루어지고, 클라이언트 객체는 데드라인 시간 이내에 서버 객체로부터 실행된 서비스 결과를 획득한다. 그러므로 실시간 통신 시스템 설계자는 서버 객체로부터 실행된 결과를 획득할 수 있는 클라이언트 객체의 데드라인을 설정할 때 서버 객체 시스템의 속

도와는 무관하게 이루어질 수 있도록 서비스 시간을 보장하여야 한다.

실제적으로 실시간 통신시스템은 서버 객체 노드의 서비스를 보장하기 위해서 필요한 시간과 다른 실시간 통신 객체 노드들 간의 메소드 통신에 소요되는 시간의 합은 서버 객체 노드의 서비스 보장 시간에 실질적으로 분산된 실시간 통신 객체 노드들 간의 통신에 부여되는 최대 전송 시간의 합보다는 적어야 한다. 이는 또다시 클라이언트 객체가 서버 객체에 대한 요구된 서비스 결과를 되돌려 받을 수 있도록 소요되는 데드라인 시간보다도 적어야 한다.

## 2.2 실시간 통신 TMO 구조

위에서 언급된 시간 구동 개념을 토대로 객체 지향 실시간 통신 분산 프로그래밍 구조를 구체적으로 나타내기 위해 시간 구동과 메시지 구동에 대한 구조를 일반적인 형태의 구조로 모델링 할 수 있어야 하며, 이러한 모델링은 분산된 실시간 통신 객체와 비실시간 객체를 포함하여야 하고, 메시지 구동과 시간구동을 모든 객체 구조 형태에 적용함으로써 실시간 통신 프로그래밍 설계가 이루어질 수 있다.

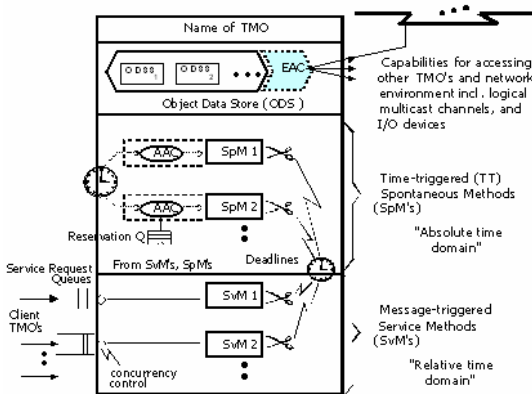


그림 2-2 실시간 통신 TMO 구조  
Fig. 2-2 Structure of TMO of Real time Communication

그림 2-2는 실시간 통신 객체 지향 방식을 이용하는 시간 구동과 메시지 구동 방식의 TMO 구조를 나타낸 것으로서 다음과 같이 4개의 영역으로 구성되어

있다[3][6].

ODSS(Object-Data-Store section)은 객체 데이터의 저장 영역을 의미하는 것으로서 실시간 통신 시물레이션을 수행하기 위한 객체들의 데이터 멤버이다. 데이터 멤버는 외부 클라이언트로부터 송수신된 메시지의 결과를 통해 갱신된다. 외부 클라이언트로부터 메시지 송신은 다른 객체 노드의 SpM 메소드 또는 SvM 메소드 영역으로 통신 가능하며, 이는 서비스 요구를 호출한 후 돌아온 결과 메시지를 계산함으로써 이루어진다.

SpM(Service Spontaneous Method Section)은 자동적인 시간 측정을 위한 메소드 영역으로서 외부 클라이언트와 서버 객체 노드의 통신에 의해 자동적인 시간 측정을 위한 영역이다.

SvM(Service Message Method section)은 TMO 영역에 속해있는 객체 멤버들의 갱신 상태를 다른 객체 노드 SpM 메소드 영역이나 SvM 메소드 영역에 정보를 주고받음으로써 분산된 객체 노드간의 동시성을 만족함과 동시에 설정된 데드라인 시간내에 서비스 메시지의 송수신이 완료된다.

EAC(Environment Access-Capability section)은 다른 객체와의 효율적인 호출경로를 설정해주거나 또는 입출력 장치의 인터페이스 설정 영역이다.

실시간 통신 스케줄링 대상은 TMO 구조의 실시간 통신 특성에 따라 스케줄링 클래스로 나누어지고, 실시간 통신 시간 구동 메소드와 서비스 메시지 구동 메소드를 제공함으로써 실시간 통신 특성을 부여할 수 있다. 또한 TMO 구조는 실시간 통신 객체 모델에서 이용된 기본 병행성 제약을 따르지 않고 서비스 메시지 메소드 구동에 따라 실시간 통신 객체들을 유연하게 적용할 수 있다.

TMO 구조에서 나타난 실시간 통신 메소드는 시간 구동 메소드와 서비스 메시지 구동 메소드로 나뉜다. 시간 구동 메소드는 매 주기마다 능동적으로 동작하며, 다른 분산 노드 TMO의 실시간 통신 메소드에 의해 호출되지는 않는다. 따라서 시간 구동 메소드에 인자를 전달하기 위해서는 객체 내의 자료를 통해서 간접적으로 전달하거나 시간 구동 메소드가 능동적으로 서비스 메시지 메소드를 호출하여 인자를 얻는다. 서비스 메시지 메소드는 실시간 통신 메소드의 호출에 의해서 수동적으로 동작되며 실시간 통신 객체들

간의 메시지 전달은 일반 객체 모델과 마찬가지로 서비스 메소드 호출을 통해서 이루어진다.

### III. 시간 구동 메소드의 시간동작

시간 구동 메소드에 의해 구동된 시간 제약은 설계 시간 동안에 구체적으로 설정하기 위해서 시간 구동 메소드 영역의 첫 번째 실행 구문에 표현되도록 한다. 그림 3-1은 시간 구동 메소드에서 수행되는 시간 제약 사항을 자동 동작 조건식으로 표현한 것이다.

```

"for t = from 10am to 10:50am every 30min
start-during (t, t+5min) finish-by t+10min"
which has the same effect as
"start-during (10am, 10:05am)
finish-by 10:10am",
"start-during (10:30am, 10:35am)
finish-by 10:40am"
    
```

그림 3-1 시간 구동 메소드의 자동 동작 조건 표현식  
Fig. 3-1 Autonomous activation condition Expression of time triggered method

그림 3-1에서 나타난 표현식은 실시간 시뮬레이션 동작 시간이 오전 10시부터 시작되어 매 30분을 주기로 하여 오전 10시 50분까지 실행이 완료된다는 의미를 가지고 있다. 즉 첫 번째 시뮬레이션 동작 시작 시간이 오전 10시에서 10시 5분 이내의 어느 시각에서나 가능하고 반드시 오전 10시 10분에 끝나야 함을 의미한다. 또한 다음 두 번째 시뮬레이션 실행 시간은 주기를 30분으로 설정하였기 때문에 오전 10시 30분에서 35분 이내의 어느 시각에서나 시작은 가능하고 반드시 10시 40분에 실행을 완료 하여야 한다.

시간 구동 메소드에 대한 자동 동작 조건 영역은 실제로 구동된 시간을 정하지 않고 단지 구동된 시간을 후보로 포함시킨다. 이는 자동 동작 영역에서 나타나는 구동된 시간 후보군 중에 동적으로 선정되기 때문이다. 객체내의 서비스 메소드가 실행되는 시간 구동을 동적으로 설정하는 것은 시간 구동 메소드의 특

성 실행 시간 내에 서비스를 요구할 때만 발생한다. 실제로 구동된 자동 동작 조건 후보군은 각각의 시간 구동 메소드의 이름을 갖는다. TMO 구조를 갖는 실시간 통신 컴퓨터 시스템들은 항상 분산된 네트워크의 형태를 가지고 있어야 하며, 각각의 실시간 통신 시뮬레이터의 설계자는 객체의 서비스 능력에 대한 시간 보장을 제공하여야 한다. 실시간 시뮬레이션 설계자는 보장된 실행 시간 윈도우내에 각각의 서비스 메소드에 대한 출력 결과를 산출하고, 각각의 시간 구동 메소드는 서비스 메소드로부터 서비스 실행을 요구하며, 서비스 메소드에 대한 완료 시간을 보장하도록 서비스 메소드의 명세서에 나타낸다. 각각의 서비스 메소드에 대한 절차는 클라이언트 객체 설계자들에게 통보되며, 서버 객체 설계자는 각각의 클라이언트 객체에 대한 실행 엔진의 이용이 가능하도록 시간 윈도우 명세서를 결정한다. 그러므로 서버 객체는 각각의 클라이언트 객체의 출력 동작이 시간 윈도우 내에 수행될 수 있도록 항상 서비스 메소드를 실행한다. 기본적인 동시성 제약이 시뮬레이션 설계자에게 부과된 부담을 줄일 수 있다.

## IV. 실시간 통신 시뮬레이션 프로그래밍 구현

### 4.1 TMO로 구현한 DHS

실시간 통신 시뮬레이션 프로그래밍을 하기 위해 TMO 구조를 이용하여 DHS(Distributed High-Precision Simulation)라는 응용 환경에 적용하였다. 그림 4-1은 TMO 구조를 이용하여 DHS를 나타낸 것으로서 객체 데이터 멤버들은 각 팔호없이 이루어지며, DHS에서 유지되고 있는 정보는 객체 데이터 저장영역내에 객체 멤버에 대한 정보를 나타낸다. 객체 데이터 저장 영역은 다음 세 가지 환경 구성 요소에 대한 상태를 나타내는 객체 멤버들로 구성되어 있다.

- ▶ Flying Airplane Group Container Information (Environment)
- ▶ Flying Object Tracking Information (Reporter)
- ▶ DHS Space(Sky and Land)

DHS를 구성하고 있는 객체 데이터 멤버는 SpM을 이용하여 주기적으로 정보를 갱신한다. DHS TMO내의 SpM은 계속해서 동작하고 각각 그들의 실행을 순간적으로 완료한다. 그러므로 SpM 영역은 객체 멤버의 상태 변화를 계속적으로 나타내고, 이는 DHS 환경 구성 요소인 ODSS 세그먼트 부분의 객체 멤버 데이터를 계속적으로 SpM 영역을 참조하여 주기적으로 갱신된다. 또한 다중 SpM 영역은 응용 환경 구성 요소들 사이에 존재하는 병렬적인 특성을 지속적으로 나타내기 위해 사용되는 것으로서 동시에 동작될 수 있다.

<b>DHS (Distributed High-Precision Simulation)</b>
Access Capability (to other TMOs) None
Object Data Store <div style="border: 1px solid black; padding: 5px; text-align: center;">                     DHS Space(=Sky+Land space)                      Flying Airplane Group Container Information(Environment)                      Flying Object Tracking Impormation(Reportr)                      [Radar on Land]                      [Land_Launcher on Land]                 </div>
SpM" Update the state descriptors in ODS" Update the state of Target in Land [Update the state Flying Airplane Group Container] [Update the state of Reporter on Land] [Update the state of Radar on Land] [Update the state of Land_Launcher on Land]
SvM Receive Flying Airplane Information From FAGC Receive Request From Radars Receive Land_LauncherInformation Receive Flying Airplane Information From ASPACE Receive Report From Radars Receive LandLauncher From Reporter

그림 4-1 DHS에 대한 TMO 명세서  
Fig. 4-1 High-Level Specification of the DHS TMO

#### 4.2 분산된 다중노드에서 실시간 제어절차

DHS에서 컴퓨터를 토대로 한 제어 시스템을 어떻게 적용하느냐 하는 것이다. 제어 시스템의 비행체 추적 기능은 사용되어지는 논리적인 알고리즘의 이론에 의해 결정되며, DHS의 시뮬레이션 환경에서는 Reporter와 같은 하나의 제어 시스템 명세서를 구체적으로 작성함으로써 실시간 제어를 수행할 수 있다.

그림 4-2은 Reporter 제어 시스템을 포함하여 시뮬레이션을 수행함으로써 발생하는 Reporter의 출력에 대한 다른 TMO의 컴퓨터 제어 시스템과의 처리과정

을 단순 명세서로 표현한 것이다. 그림 4-2의 단순 명세서는 그림 4-1에서 나타난 명세서보다 구체적으로 추적 명세서를 나타내었다.

Reporter
Access Capability (to other TMOs): Land_Launcher.ReceiveReportFromReporter
Object Data Store -Radar_Reports_Received
[Initialization] -Radar_Reports_Received=Empty
SpM: [SpM] 1. Compare current Radar_Reports_Received to previous one. If there are objects which reasonably close to others regarding time and position, remove them based on time. (An object data received earlier should be removed first) 2. Send Reports to display based on comparison. Each object should be noted as radar or danger. 3. Send the information of the flying airplane (ID, position) to Land_Launcher.
SvM [ReporterReceiveReportFromRadars] 1. Receive data from Radars

그림 4-2 Reporter 제어 시스템의 단순 명세서  
Fig. 4-2 Simple Specification of Reporter Control System

그림 4-3는 Reporter 제어 컴퓨터 시스템의 상세 명세서를 나타낸 것으로서 Reporter 제어 컴퓨터 시스템은 시간 구동 메소드에 대한 자동 동작 조건을 가지고 있고, 각각의 시간 구동 메소드와 서비스 메시지 메소드에 대한 입력과 출력 절차를 가지고 있으며, 또한 서비스 메소드에 대한 초기 조건도 나타내고 있다. 메소드에 대한 입력 절차는 호출 변수 또는 객체 데이터 저장 영역과 입력장치로부터 들어오는 데이터를 나타내며, 외부 클라이언트 시간 구동 메소드와 서비스 메소드로부터 입력되는 실행 시간 동안의 데이터를 생성하는 동작이다.

Reporter 제어 컴퓨터 시스템의 출력 명세서에서 나타난 사항을 보면 데이터를 다른 객체 TMO 구조 또는 출력 장치에 보내기도 하며, 객체 데이터 저장 영역 속으로 저장 한다. 서비스 메소드에 대한 초기화 과정은 서비스 메소드가 클라이언트로부터 호출된 후에 초기화시킴으로서 동시성을 만족한다.

실시간 통신을 필요로 하는 분산된 객체 노드에 TMO 구조를 이용하여 객체 노드간의 실시간 통신 메소드를 SpM 영역과 SvM 영역으로 명확하게 분리하여 적용함으로써 실시간 통신시스템을 구축함으로써 비행체 추적시간을 줄일 수 있었으며, 또한 TMO

구조에서 이용되는 시간 구동 및 메시지 구동 메소드는 충분한 실시간 통신 서비스를 제공하기 위한 데드라인을 보장함으로써 데드라인 시간을 설정하기 위한 실시간 통신 시뮬레이션에 의해 비행체 추적 시간을 줄일 수 있었다. 분산된 객체 지향 실시간 통신시스템에서 TMO 구조를 이용한 단단계 프로그래밍 설계는 객체에 요구된 단순 절차, 상세 절차에 따라 실시간 프로그래밍을 유연하고도 쉽게 작성할 수 있다.

범을 이용한 TMO 구조에서 추적 프로그래밍을 하나의 통합된 실시간 통신에서 이동체 추적에 적용할 수 있도록 제안하였다. 실시간 통신 객체 모델은 시스템 복잡도의 증가에 따라 객체지향 요소, 실시간 통신 요소, 신뢰성 요소 등을 적용한 실시간 통신시스템의 다양한 요구사항을 잘 반영할 수 있는 실시간 통신시스템 모델과 실시간 통신시스템 개발 기간을 단축시킬 수 있는 통합된 개발환경을 연구하였다.

Control computer system in Reporter
<b>Access Capability (to other TMOs):</b> Radar (<Accept_spot_check_request>)
<b>Object Data Store</b> Radar data received, Flying object tracking information
<b>SvM:</b> SvM1 Radar Data Processing step "Process all the radar data received since the last processing cycle, update the flying object tracks" AAC for T=from TMO_START+WARMUP_DELETE_SECS to TMO_START + SYSTEM_LIFE_HOURS every PERIOD start_during(T,T+START_WINDOW) finished-by T+DEADLINE InputSpec: Radar data received in the object in the object data store OutputSpec: <deadline xxxx msec> Reflect changes onto the object data store, i.e., Radar data received, Flying airplane tracking information. <deadline xxxx> Send spot-check radar request to Radar if ...
<b>SvM</b> SvM1 Receive_form_Radar_On_Land(post_list) <Accept-with-Delay_bound-ofACCEPTANCE_TIME_LIMIT> under MAX_REQUEST_RATE finish-within EXECUTION_TIME_LIMIT "Receive From Radar-on Land the information on all recent detections" InitiationCond: Other SvM1 in vacations are not in place. InputSpec: pos-list = array of (return_Type(=scan_search/Spot_check), position, time, predicted_time) OutputSpec: <deadline yyyy msec> Deposit the radar data received in the object data Store SvM2 Accept A device from ... <Accept-via-...>

그림 4-3 Reporter 제어 컴퓨터 시스템의 상세 명세서  
 Fig. 4-3 Detailed design Specification of Reporter Control Computer System

### V. 결론

실시간 통신에 의한 이동체 추적 시스템에서는 계산 결과, 논리적 정확성의 결과가 산출되는 추적시간의 정확성을 요구한다. 만일 시스템이 시간 제약 조건을 만족하지 못하면, 시스템 실패가 발생한 것으로 간주하며, 또한 실시간 통신시스템은 정확히 수행할 뿐 아니라 외부의 비동기적인 사건에 대하여 주어진 시간 안에 응답할 수 있는 시스템이어야 한다. 실시간 통신 객체 모델이 가지고 있는 기존의 실시간 통신 특성들을 포함하면서 시간 구동 메소드와 서비스 메시지 메소드 객체에 새로운 객체지향 프로그래밍 기

### 참고 문헌

- [1] G. Booch, *Object-Oriented Design*, Benjamin Cummings, CA, 1991.
- [2] T. Bihari, P. Gopinath, and K. Schwan, "Object-oriented design of real-time software", Proc. IEEE CS 10th Real-Time Systems Symp., pp.194-201 1989.
- [3] K. H. Kim, "Object Structures for Real-Time Systems and Simulators", IEEE Computer 30, pp.62-70, 1997.
- [4] A. Attoui and M. Schneider, "An object-oriented model for parallel and reactive systems", Proc. IEEE CS 12th Real-Time Systems Symp., pp.84-93, 1991.
- [5] J. Hernandez and J. A. Sanchez, "RT-MODULA2: An embedded in MODULA2 language for writing concurrent and real time programs", ACM SIGPLAN Notices 27, pp.26-36, 1992.
- [6] K. H. Kim and J. Liu, "Deadline Handling in Real-Time Distributed Objects", Proc. ISORC 2000, Newport Beach, CA, pp.7-15, March 2000.
- [7] M. Champlain, "Synapse: A Small and Expressive Object-based Real-time Programming Language", ACM SIGPLAN Notices 25 pp.124-134, 1990; O. J. Dahl, "hierarchical program structuring", in Structured Programming, eds. Dahl, Dijkstra, and Hoare, Academic Press, NY, 1972.
- [8] Y. Ishikawa, H. Tokuda, and C. W. Mercer, "An object-oriented real-time programming language", IEEE Computer, pp.66-73, 1992.
- [9] J. C. Laprie, "Dependability: a Unifying Con-

cept for Reliable, Safe, Secure Computing", in Information Processing, ed. J. van Leeuwen, pp.585-593, 1992.

[10] H. Kopetz, et al., "Fault-Tolerant Membership Service in a Synchronous Distributed Real-Time System", Proc. IFIP WG 10.4 Conf. on Dependable Computing for Critical Appl., Santa Barbara, pp.167-174, Aug. 1989.

저자 소개



**김광준(Gwang-Jun Kim)**

1993년 조선대학교 컴퓨터공학과 졸업(공학사)

1995년 조선대학교 대학원 컴퓨터공학과 졸업(공학석사)

2000년 조선대학교 대학원 컴퓨터공학과 졸업(공학박사)

2000년~2001년 Dept. of Electrical & Computer Eng. Univ. of California Irvine Postdoc.

2003년~2006년 2월 여수대학교 컴퓨터공학과 조교수

2006년 3월~현재 전남대학교 컴퓨터공학과 조교수

※ 주관심분야 : ATM망, 인터넷 통신, 컴퓨터 네트워크, 실시간 통신 프로그래밍, 영상 처리 및 통신, 프로그래밍 언어(Visual C++, Java), 이동 통신 등



**서종주(Jong-joo Seo)**

2004년 여수대학교 컴퓨터공학과 졸업(공학사)

현재 : 전남대학교 대학원 컴퓨터공학과

※ 주관심분야 : ATM망, 실시간 데이터 통신, 컴퓨터 네트워크, TCP/IP 혼잡제어, 이동 통신 등



**강기웅(Ki-Ung Kang)**

2005년 여수대학교 컴퓨터공학과 졸업(공학사)

현재 : 전남대학교 대학원 컴퓨터공학과

※ 주관심분야 : ATM망, 실시간 데이터 통신, 컴퓨터 네트워크, TCP/IP 혼잡제어, 이동 통신 등



**윤찬호(Chan-ho Yoon)**

1997년 호남대학교 컴퓨터공학과 졸업(공학사)

2000년 조선대학교 대학원 컴퓨터공학과 졸업(공학석사)

현재 : 조선대학교 대학원 컴퓨터 공학과 박사과정

※ 주관심분야 : ATM망, 데이터 통신, 컴퓨터 네트워크, TCP/IP 혼잡제어, 이동 통신 등