

# CORBA 기반의 분산 실시간 응용을 지원하기 위한 실시간 이벤트 필터링

윤은영\*, 윤용익\*

\*숙명여자대학교 컴퓨터과학과

e-mail:eyyoon@cs.sookmyung.ac.kr

## A CORBA-Based Real-Time Event Filtering for Supporting Distributed Real-Time Applications

Eun-Young Yoon\*, Yong-Ik Yoon\*

\*Dept of Computer Science, Sook-Myung Women's University

### 요 약

이벤트 서비스는 멀티미디어 통신, 전자상거래 등과 같은 대용량의 분산 응용 시스템에서 발생하는 비동기적 이벤트의 전송을 제어하기 위한 서비스이다. 하지만 기존의 이벤트 서비스는 분산 환경의 실시간 응용 시스템을 지원하기 위해 필요한 실시간 지원 기능들이 결여되어 있어 이를 지원할 수 있는 실시간 이벤트 서비스 처리 방안이 요구된다. 본 논문에서 제안하는 RTEF(Real-Time Event Filtering) 서비스는 기존의 ORB 구조에 실시간 이벤트 필터링, 실시간 모니터 및 QoS 저장소를 포함시킨 실시간 이벤트 서비스 미들웨어이다. 특히, RTEF는 이벤트 사용자의 실시간 요구사항을 분산 실시간 응용시스템에 반영시키기 위하여 이벤트 필터링 기능을 실시간 처리에 맞게 강화 시켜 기존의 시스템에서 지원하지 못한 사용자의 실시간 요구사항을 지원하는데 중점을 두었다. 분산 실시간 응용 시스템에 RTEF를 적용하면 사용자에게 자신이 원하는 실시간 요구조건에 맞게 필터링된 이벤트 처리 결과를 제공하게 되므로 궁극적으로 보다 높은 서비스(QoS)를 제공하게 된다. 또한 이 과정을 통해 불필요한 이벤트가 필터링 됨으로써 전체적인 네트워크 트래픽(traffic)을 감소시키는 효과를 가져온다.

### 1. 서 론

최근 급속한 인터넷의 보급으로 인터넷 사용자 급증하면서 사용자의 요구사항을 반영 할 수 있는 안정되고 질 높은 서비스에 대해 관심이 모아지고 있다. 분산 환경에서 사용자에게 질 높은 서비스를 제공하기 위해서는 통합된 환경과 사용자의 요구사항을 반영할 수 있는 시스템이 요구된다.

지금까지 이 기종의 분산 환경 시스템들을 통합된 환경처럼 제공하기 위해 많은 연구들이 진행되고 있으며 이를 위해 분산환경의 표준화 구조인 OMG의 CORBA가 정의되었다. CORBA는 분산 환경에서 객체 지향으로 개발된 애플리케이션들의 서로 다른 하드웨어 플랫폼, 이종의 통신 프로토콜을 통합하여 객체들간의 통신을 단일한 사용자 환경처럼 지원한

다[4]. CORBA 객체 서비스 중에 하나인 이벤트 서비스는 멀티미디어 통신, 전자상거래 등과 같은 대용량의 분산 응용 시스템에서 발생하는 비동기적 이벤트의 전송과 하나 이상의 이벤트 공급자가 하나 이상의 이벤트 소비자에게 메시지를 전송하는 것을 지원한다[2][3]. 그러나 기존의 이벤트 서비스는 실시간 응용시스템에서 사용자가 요청하는 실시간 요구사항을 지원하기 위해 필요한 실시간 지원 기능들이 결여되어 있어 이를 지원 할 수 있는 실시간 이벤트 서비스가 요구된다.

본 논문에서는 이러한 요구사항들을 지원하기 위한 CORBA 기반의 실시간 이벤트 필터링 (RTEF : Real-Time Event Filtering) 서비스 처리방안을 제안한다. 특히, RTEF는 이벤트 사용자의 실시간 요구사항을 분산 실시간 응용시스템에 반영시키기 위

하여 이벤트 필터링 기능을 실시간 처리에 맞게 강화 시켜 기존의 시스템에서 지원하지 못한 사용자의 실시간 요구사항을 지원하는데 중점을 두었다.

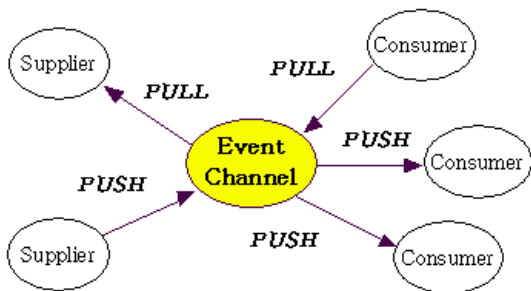
본 논문의 구성은 다음과 같다. 2장에서 관련연구로 CORBA 이벤트 서비스와 객체 시간 서비스를 분석하고, 3장에서 이벤트 필터링과 QoS 저장소를 포함한 RTEF의 전체적인 구조를 설계하고 구성요소의 기능을 설명하며 본 논문에서 제안한 RTEF를 분산 실시간 응용시스템에 적용하는 방안을 제시한다. 4장에서 RTEF 테스트 프로그램 구현에 대해 설명한 후, 5장에서 본 연구의 결론 및 향후 연구 과제를 제시한다.

## 2. 관련 연구

본 절에서는 CORBA 객체 서비스 중에 하나인 이벤트서비스에 대해서 기술하고, 실시간 서비스를 지원하기 위한 객체 시간 서비스에 대해 기술한다.

### 2.1 CORBA Event Service

기존의 CORBA 애플리케이션들 사이에서 발생하는 통신의 형태는 상대방의 인터페이스를 이용해서 직접 전달하는 방법을 사용한다[7]. 그러나, CORBA의 이벤트 서비스는 ORB상에 존재하는 애플리케이션들의 간접적인 통신을 위한 모델을 정의하고 있다[5]. 다시 말해서 클라이언트는 특정 서버에 있는 객체의 연산을 직접 호출해서 데이터를 전달하는 대신, 불특정 다수의 객체들에 전달될 수 있는 이벤트를 전달한다. CORBA 이벤트 서비스는 (그림 1)과 같이 공급자와 소비자, 이벤트 채널의 요소로 이루어져 있는데 이벤트를 전달하는 애플리케이션을 공급자(Supplier)라 하고 이벤트를 전달받아서 처리하는 애플리케이션을 소비자(Consumer)라고 한다. 이들의 통신형태로 크게 Push 모델과 Pull



[그림 1] CORBA 이벤트 서비스 모델

모델이 있는데 Push 모델에서 공급자는 이벤트가 발생하면 소비자에게 이벤트를 전달함으로써 통신을 시작하고, Pull 모델에서는 소비자가 이벤트를 요구함으로써 이벤트 통신이 시작된다. 여기에서 이벤트 채널(Event Channel)은 공급자와 소비자 사이에서 발생하는 이벤트 통신을 중재하는 역할을 수행하는 CORBA 이벤트 서비스의 핵심이다. 그러나, 표준 CORBA 이벤트 서비스에서는 실시간을 지원하기 위해 필요한 실시간 이벤트 필터링과 QoS 강요(enforcement), 우선순위, 마감시간과 같은 시간적 제약을 가진 실시간 이벤트 지원 기능이 결여되어 있어 이를 지원하는 실시간 이벤트 서비스가 요구된다. 세계적으로 실시간 이벤트를 연구하여 개발중인 제품으로는 워싱턴 대학에서 개발한 고성능의 표준 CORBA와 호환되는 실시간 ORB인 TAO[8]와 Rhode Island 대학과 미해군 NRaD 연구소, 그리고 MITRE사의 공동 작업으로 개발된 NRaD 등이 있다.

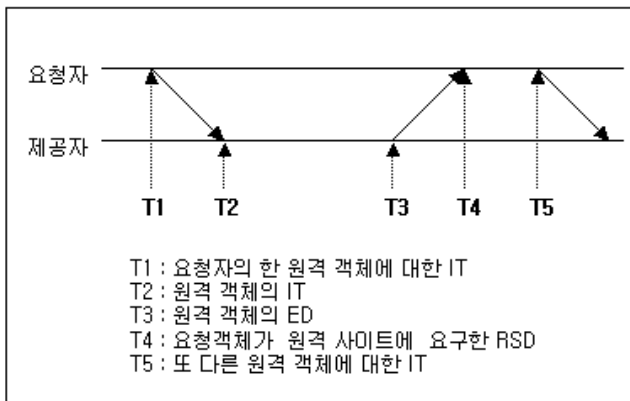
### 2.2 객체 시간 서비스

OMG는 ORB를 기반으로 하는 분산환경에서 특정 시간 순서를 필요로 하는 이벤트의 처리를 위해 시간을 동기화 시키는 서비스가 필요함에 따라 COSS3(Common Object Service Specification3)에 객체 시간 서비스의 명세를 규정하고 있다[5]. 여기에서 규정한 객체 시간 서비스는 크게 기본 시간 서비스와 타이머 이벤트 서비스로 나누어진다. 기본 시간 서비스는 애플리케이션들의 클라이언트에 대해 시간의 취득, 시간의 예러폭의 취득, 두 가지 시간의 비교 및 시간간격의 계산, 오버랩 시간의 계산 등의 기능을 제공한다. 타이머 이벤트 서비스는 시간을 설정 혹은 해제하기 위한 인터페이스로 구성되어 있다. 시간에는 절대시간, 상대시간 또는, 주기적인 시간의 어느 쪽을 설정할 수 있다. 여기서 설정된 시간이 되면 등록된 이벤트가 이벤트 채널을 경유로 애플리케이션에 송신된다. 또한 이벤트의 송신에는 이벤트 서비스의 Push 모델을 사용하고 있다.

분산 환경에서 실시간 이벤트 처리를 요구하는 화상회의, 원격강의, 전자상거래 등과 같은 실시간 응용 시스템들의 시간적인 정확성을 보장하기 위해서는 시간적인 제약 조건들을 명확하게 제시하여야 한다. 분산처리를 요구하는 시스템의 기본 흐름은 원격처리요구 단계, 서비스 처리 단계, 결과 전달 단계 등 크게 3단계로 구분된다. 이들 각 단계마다

필요한 시간적 제약조건이 있으며 이를 명확하게 응용 시스템에서 정의하여 적시성을 보장받도록 해야 한다. 이를 위하여 분산 처리 환경 하에서 적시성을 위한 시간적 제약 조건들을 다음과 같이 크게 4가지로 정의한다.

- 생성 시간 (IT : Invocation Time)
  - 실행 시간 (CT : Computing Time)
  - 이벤트 데드라인 (ED : Event Deadline)
  - 원격 실행 데드라인 (RSD : Remote Service Deadline)
- IT는 원격 사이트로 서비스를 요구하는 시간을 의미하며, CT는 요청자가 원격 서비스를 포함하여 자신의 수행에 필요한 시간을 의미한다. ED는 이벤트의 수행이 반드시 완료 되어야 하는 시점을 의미하며, RSD는 이벤트가 원격 사이트로 서비스를 요구한 후에 수행 결과가 반드시 환원되어야 하는 시점을 의미한다. RSD가 분산 실시간 처리 시스템에서 보장되어야 하는 제약조건이다. 이들 간의 관계는 (그림 2)에서 보여주고 있는 것과 같이 RSD는 ED 내에 이루어져야 한다. 즉, 이를 위해서는 서비스를 제공하는 서버내의 스케줄러를 포함한 시스템에게 요청자의 요구사항이 반영되도록 하여야 한다. 분산 실시간 시스템 내에서 RSD의 보장은 적시성 보장의 기준이 된다.



[그림 2] 분산처리과정에서 시간적 제약조건

### 3. RTEF(Real-Time Event Filtering) 구조 설계

#### 3.1 요구사항

CORBA 통합 분산 기술이 많은 관심을 받으면서 CORBA를 기반으로 분산 응용 프로그램을 구현하려는 연구가 계속되고 있다. 그러나 분산 응용 프로그램에서 중요한 실시간적 요구사항들을 고려한 연구들은 미흡한 실정이다.

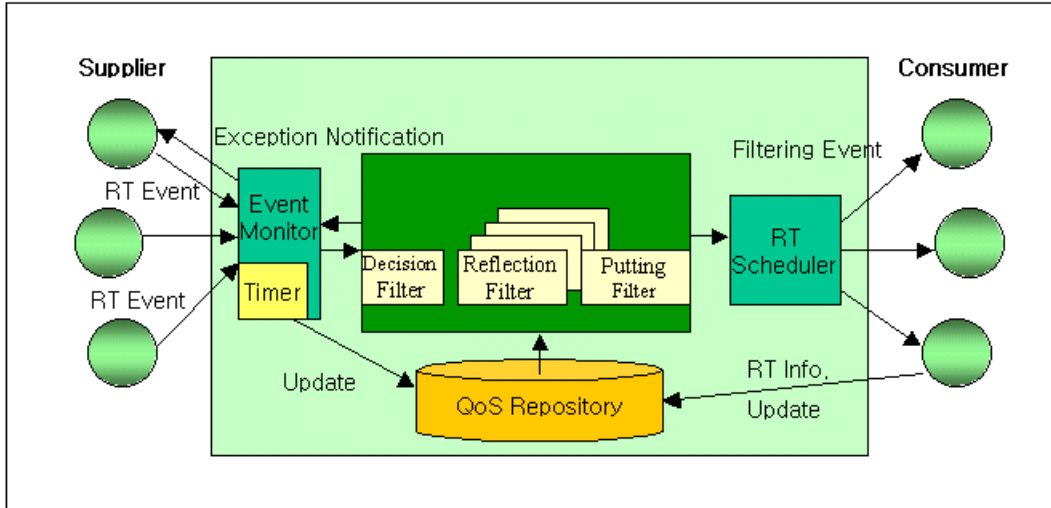
기존의 CORBA 이벤트 서비스에서는 공급자가 이벤트를 발생시키면 이벤트 채널을 통해 소비자에게 이벤트를 밀어 보낸다[5]. 그러나 소비자가 단지 공급자로부터의 발생하는 이벤트 중 일부분에만 관심이 있다면, 소비자에게 불필요한 이벤트를 받지 않게 하는 이벤트 필터링이 필요하다. 또한, 사용자가 입력한 실시간 이벤트가 마감시간과, 우선순위 등의 실시간 요구조건을 반영하여 이벤트를 계속 진행시킬 것인지 진행 전에 중단시킬 것인지를 결정하는 실시간 이벤트 필터링이 필요하다. 이러한 이벤트 필터링의 효과는 QoS를 강요하는 역할을 하게 되며 또한, 분산 시스템에서 발생하는 불필요한 이벤트의 전송을 제어함으로써 네트워크 전송 트래픽(traffic)을 감소시킬 수 있다.

#### 3.2 이벤트 필터링

일반적으로 이벤트는 사용자가 응용 프로그램의 GUI 환경에서 시스템 처리 중에 데이터의 상태를 변경시키기 위해 입력하는 모든 행위(action)라고 말할 수 있으며, 다양한 이벤트 타입이 존재하게 된다. 예를 들어, 사용자가 원격에 있는 메소드의 결과 값을 요청하는 결과 요청 이벤트와 자신이 입력한 이벤트가 원격에 있는 이벤트 소비자에게 전달되는 것을 목적으로 하는 전달 이벤트가 있으며, 이들 이벤트에 사용자가 원하는 시간적 제약을 포함시킨 실시간 이벤트가 있다. 실시간 이벤트는 분산 환경에 존재하는 많은 응용 프로그램들 중에서 시간적 제약(time constraint)을 요구하는 이벤트를 말한다. 예를 들어, 사용자가 이벤트를 요청할 때 이벤트의 결과에 대해 사용자가 원하는 시간(deadline)과 우선순위(priority)를 입력한, 시간적 제약을 가진 이벤트를 실시간 이벤트라 할 수 있다.

필터는 크게 선택(Selective), 변환(Transforming), 혼합(Mixing) 등의 세 가지 기능이 있으며 멀티미디어 통신에서 광범위하게 사용되는 기술이다[2]. 선택 필터는 몇 가지 주어진 정보를 기반으로 데이터를 계속 진행시킬 것인지 아니면 중단시킬 것인지를 선택하며, 변환 필터는 선택된 데이터를 주어진 조건에 따라 변환시키는 필터이다. 혼합 필터는 두 개의 데이터를 혼합하여 새로운 데이터를 생성한다.

본 논문에서 제안하는 RTEF는 이러한 필터링의 기능을 실시간 이벤트 처리에 적용하여 사용자가 원하는 시간적 제약조건에 만족하는 이벤트만을 필터링하여 전송한다. 이를 처리하기 위해 필터링 기능



[그림 3] RTEF 구조

의 선택 기능과 변환 기능을 적용하였는데, 선택 기능은 이벤트가 입력되면 몇 가지 기준을 적용하여 필요한 이벤트를 선택하는 기능으로 본 논문에서는 실시간 이벤트의 전송을 기존에 저장된 시간적 정보를 기반으로 하여 중단시킬 것인지 진행시킬 것인지를 결정하는데 적용하고, 변환 기능은 입력된 실시간 이벤트의 요구사항과 기존의 사용자로부터 입력된 QoS 요구사항을 반영하여 사용자가 원하는 이벤트로 변환하는데 적용하였다.

### 3.3 RTEF 모델링

(그림 3)과 같이 RTEF의 구조는 이벤트 소비자, 공급자, 이벤트 모니터, 이벤트 필터링, QoS 저장소 등으로 구성되어 있으며 각각의 기능은 다음과 같다.

· 이벤트 공급자(Supplier)

이벤트를 발생시키는 이벤트 공급자로 이벤트 소비자가 요청한 이벤트를 발생시킬 수도 있고, 자신이 발생시킨 이벤트를 자신이 원하는 소비자에게 보낼 수 있다.

· 이벤트 소비자(Consumer)

이벤트 공급자가 발생시킨 이벤트를 제공받게 되는 소비자로 자신이 관심 있는 이벤트를 선택하고, 원하는 이벤트를 요청할 때 이벤트 결과에 대한 마감시간과 우선순위 등과 같은 실시간 정보를 입력할 수 있다.

· 이벤트 모니터(Event Monitor)

실시간 이벤트에 대한 정보와 전송시간, 우선순위와 같은 실시간 조건을 모니터하여 이에 대한 정보를 QoS 저장소와 이벤트 필터링에 전달하며 예외상황

발생 시 사용자에게 실시간으로 예외상황 메시지를 전송한다.

· 실시간 스케줄러 (Real-Time Scheduler)

이벤트 필터링에서 필터링된 이벤트들을 우선순위와 작업시간을 반영한 알고리즘으로 스케줄링하여 이벤트 소비자에게 전달한다.

· 결정 필터(Decision Filter)

결정 필터는 실시간 이벤트가 입력되면 이벤트를 전송시키기 전에 여러 조건을 비교하여 이벤트를 계속 진행시킬 것인지 아니면 중단시킬 것인지를 결정하는 필터이다. 예를 들어, 마감시간(deadline)을 요청하는 실시간 이벤트가 입력되면 소비자가 요청한 이벤트에 대한 마감시간(deadline)을 QoS 저장소에 저장된 이벤트 실행시간 (worst case time)과 비교하여 입력된 실시간 이벤트를 계속 진행시킬 것인지 중단시킬 것인지를 결정하여 서버로 전송하기 전에 이벤트를 여과시켜 주는 기능을 제공한다.

· 반영 필터(Reflection Filter)

반영 필터는 다양한 실시간 이벤트의 요구사항을 반영하여 필터링한다. 분산 환경에서 다양하게 발생하는 실시간 이벤트의 요구사항 중 소비자가 자신이 원하는 자신만의 이벤트를 제공받기를 원할 경우와 같이 QoS를 강요하는 이벤트는 고정된 필터링 기능으로 제공하는 것에는 한계가 있다. 예를 들어, 소비자가 공급자가 제공하는 이벤트 중 자신이 관심 있는 이벤트만을 제공받기 위하여 자신이 원하는 필터링 정보를 QoS 저장소에 입력한 경우, 반영 필터는 이를 지원하기 위해서 소비자가 원하는 자신만의 이벤트 필터링을 지원할 수 각각의 필터를 선택하여 RTEF에 Plug\_In 구조로 필터링을 제공한다.



· 발송 필터(Putting Filter)

발송 필터는 결정 필터와 반영 필터를 거쳐 필터링된 이벤트를 이벤트 소비자에게 발송하는 필터로 실시간 스케줄러를 이용하여 우선순위와 마감시간을 반영한 이벤트를 발송한다. 예를 들어, 마감시간을 요청한 이벤트가 발생한 경우, 발송 필터는 입력된 실시간 이벤트의 마감시간이 QoS 저장소에 저장된 소비자 우선순위를 반영하여 분배되도록 필터링한 스케줄에 따라 이벤트를 발송한다.

· QoS 저장소 (Repository)

이벤트를 공급받게 되는 소비자(consumer)는 자신이 입력한 실시간 이벤트에 대해 QoS를 제공받기 위해서 자신이 필요한 QoS 정보를 입력하게 되며, 이러한 QoS 정보는 QoS 저장소에 저장된다. QoS 정보는 실시간 이벤트에 대한 각각의 실행시간, 소비자에 대한 우선순위 정보, 소비자가 입력한 이벤트 필터링 정보 등을 말한다.

3.4 RTEF 적용방안

본 논문에서 제안한 RTEF는 기존의 표준 ORB 구조에 실시간 이벤트를 필터링 할 수 있는 기능을 확장시킨 실시간 ORB 미들웨어이다. RTEF는 대표적인 분산 실시간 응용 프로그램인 멀티미디어 통신, 전자상거래, 실시간 증시 서비스 등에 적용될 수 있다. 실례로 전자상거래의 기반 서비스, 응용 서비스, 데이터베이스 서비스 등을 객체지향 개념으로 개발하여 RTEF를 적용시킴으로써 제품 공급자는 객체지향 개념을 이용하여 개발한 자신의 클라이언트 객체를 등록하여 전자상거래 플랫폼과 원활한 정보교환을 할 수 있으며, 향후 다른 공급자나 다른 전자상점과도 쉽게 연동 할 수 있고, 소프트웨어 모듈별 업그레이드 및 재사용이 가능하게 된다. 다시 말하면, 클라이언트 및 서버 오브젝트를 객체지향 개념을 이용하여 개발하고 이를 실시간 ORB 표준에 따라 개발하면 구매자, 판매자, 전자상점 운영자가 프로그램을 활용할 수 있다. 또한 전자상거래의 구매자, 판매자, 인증기관, 브로커, 카드은행 등 많은 거래 요소들이 실시간 ORB를 이용해 전자상거래에 접속하게 함으로써 다단계 클라이언트/서버 구조로 구현하면, 현재 CGI로 구현되어진 전자상거래의 병목현상을 피하고, 서버들을 연결된 구조로 부하를 분산시킬 수 있으며, 상호운용성과 이벤트 필터링, 우선순위, 마감시간, QoS 보장 등과 같은 실시간 지원이 가능해진다.

4. RTEF 테스트 프로그램 구현

ORB로 RTEF 테스트 프로그램을 구현하기 위해서 ORB 미들웨어인 ObixWeb을 사용하였고, 언어는 Java (JDK 1.1.5)와 DataBase 미들웨어로 MS SQL을 이용하여 개발하였다. ORB로 응용 프로그램을 구현하기 위해서는 ORB 표준 아키텍처를 따르는 개발 업체에서 개발한 ORB 구조를 지원할 수 있는 개발 툴을 이용하여 구현하여야 한다. 실시간 오퍼레이션은 실시간 이벤트 발생 시 생성되는 오퍼레이션으로 우선순위(priority), 기간(period), 실행시간(execution) 등과 같은 실시간 정보를 말한다 [2][3]. ORB 내에서 실시간 이벤트 처리를 위해서는 실시간 오퍼레이션에 대한 인터페이스가 먼저 정의되어야 한다. [그림 4]는 실시간 필터 모듈 IDL (Interface Definition Language)을 나타낸 것이다.

```

module RT_Filter
{
    typedef TimeBase :: TimeT Time;
                                // 100 nanoseconds
    typedef long Period; // 100 nanoseconds
    typedef long Preemption_Priority;
    typedef long Decision_Filter;
    typedef long Reaction_Filter;
    typedef long Putting_Filter;
    :
    :
    struct RT_Info
    { // Execution times.
        Time worstcase_execution_time_;
        Time typical_execution_time_;
        Period period_;
        Preemption_Priority;
        Decision_Filter;
        Reaction_Filter;
        Putting_Filter;
        :
        :
    }
}
    
```

[그림 4] 실시간 필터 모듈 IDL

RTEF 테스트 프로그램은 ORB를 이용하여 원격의 클라이언트와 구현객체를 연결하고, 실시간 이벤트 발생 시 클라이언트와 서버의 위치와 연결상태를 실시간으로 응답한다. 이 프로그램을 기반으로 필터링 기능을 Plug\_In 구조 개발하여 테스트하였다.

5. 결론 및 향후 연구 과제

본 논문에서 제안한 RTEF는 기존의 ORB 구조에 실시간 이벤트를 처리하는 이벤트 필터링과 실시간 스케줄러, 모니터 및 QoS 저장소를 포함시킨 실시

간 ORB 미들웨어이다. 특히, 본 논문은 사용자의 실시간 요구사항을 분산 시스템 환경에 반영시키기 위하여 이벤트 필터링 기능을 실시간 처리에 맞게 강화시켜 기존의 시스템에서 지원하지 못한 사용자의 QoS 강요와 실시간 요구사항을 지원하는데 중점을 두었다.

RTEF는 사용자가 원하는 시간적 제약조건과 이벤트 필터링 정보를 입력하면 이 정보가 QoS 저장소에 저장되고, 실시간 이벤트가 발생하면 이 정보를 반영하여 필터링이 이루어진다. 이로써 사용자는 자신이 원하는 실시간 요구조건에 맞게 필터링된 이벤트 처리 결과를 얻게 되어 궁극적으로 보다 높은 서비스를 제공받게 된다. 이 과정을 통해 불필요한 이벤트가 필터링 됨으로써 전체적인 네트워크 트래픽이 감소되는 효과를 얻을 수 있다.

본 논문을 초석으로 하여 향후에는 분산 환경에서 통합된 실시간 멀티미디어 미들웨어 플랫폼 구축을 연구하고자 한다. 이를 위해서는 본 논문에서 제안한 실시간 이벤트 서비스 시스템의 기능 확장 연구가 필요하며 확장된 실시간 모듈을 표준 ORB 기능에 포함시키고 실시간 데이터베이스 시스템, 실시간 멀티미디어 서비스, 자바 등과 연계하여 개발할 수 있는 연구가 필요하다.

#### 참고문헌

[1] S. Yau, C. Gao and F. Karim, "Object-Oriented Real-Time Event Service for Large-scale Distributed Systems", Proceedings of IEEE Workshop on Middleware for Distributed Real-time Systems and Services, December, 1997.

[2] N. Yeadon, F. Garria, D. Hutchison and D. Shepherd, "Filters: Qos Support Mechanisms for Multipeer Communications", IEEE Jour on Selected Areas In Communications, Vol. 14, No. 7, September 1996, pp. 1245.

[3] G. Liu and A. Mok, "An Event Service Framework for Distributed Real-Time Systems", Proceedings of IEEE Workshop on Middleware for Distributed Real-time Systems and Services, December, 1997.

[4] Rosemary Rock-Evans, "Middleware the key to Distributed Computing", Ovum, 1995.

[5] OMG, "CORBA Service : Common Object

Service Specification", <http://www.omg.org/corba/sectran1.htm>, 1997.

[6] E. Nett, M. Gergeleit, and M. Mock, "An Adaptive Approach to Object-Oriented Real-Time Computing", IEEE Workshop Proceedings of ISORC'98 20-22 April, 1998.

[7] Object Management Group, The Common Object Request Broker : Architecture and Specification, 2.2 ed., Fed. 1998.

[8] T. H. Harrison, C. O'Ryan, D. L. Levine, and D. C. Schmidt, " The Design and Performance of a Real-Time CORBA Event Service", Proceedings of IEEE Journal on Service Enabling Platforms for Networked Multimedia Systems, 1999.

[9] D. Schmidt, D. Levine, and S. Mungee: "The Design of the TAO Real-Time Object Request Broker", to appear in the Computer Communications Journal, summer, 1997.

[10] M. Gergeleit, M. Mock, E. Nett, J. Reumann: "Integrating Time-Aware CORBA Object into O-O Real-Time Computations", Proceedings of WORDS'97, February 5-7, 1997.