

분산 객체 브로커 내에서의 실시간 이벤트 처리 방안 연구

윤은영(eyyoon@cs.sookmyung.ac.kr), 윤용익(yiyoon@sookmyung.ac.kr)
숙명여자대학교 전산학과

Real-Time Event Service In the Distributed Object Request Broker

Yoon Eun Young , Yoon Yong Ik
Dept. of Computer Science, Sookmyung Women's University

요약

이벤트 서비스는 멀티미디어 통신, 전자상거래, 트래픽 제어시스템 등과 같은 대용량의 분산 응용 시스템에서 발생하는 이벤트의 전송을 제어하기 위한 서비스이다. 하지만 기존의 이벤트 서비스는 분산 환경의 실시간 응용 서비스를 지원하기 위해 필요한 실시간 이벤트 필터링, 우선순위, 마감시간 등과 같은 실시간 QoS 지원 기능들이 결여되어 있어 이를 지원할 수 있는 실시간 이벤트 서비스 처리 방안이 요구된다. 본 논문에서 제안하는 RTES는 기존의 ORB 구조에 이벤트 필터링, 실시간 매니저 및 QoS 저장소를 포함시킨 실시간 이벤트 서비스 미들웨어이다. 특히, RTES는 이벤트 사용자의 실시간 요구사항을 분산 실시간 응용에 반영시키기 위하여 이벤트 필터링 기능을 실시간 처리에 맞게 강화시켜 기존의 시스템에서 지원하지 못한 사용자의 실시간 요구사항을 지원하는데 중점을 두었다. 실시간 이벤트 필터링은 사용자로부터 원하는 실시간 요구사항과 사용자가 원하는 자신만의 이벤트 필터링 정보를 입력받아 QoS 저장소에 저장하고, 실시간 이벤트 발생 시 사용자가 입력한 정보를 반영한 필터링을 통해 이벤트를 처리하는 메커니즘으로써 사용자는 자신이 원하는 실시간 요구조건에 맞게 필터링된 이벤트 처리결과를 얻게 되어 궁극적으로 보다 높은 서비스(QoS)를 제공받게 된다. 또한 이 과정을 통해 불필요한 이벤트가 필터링 됨으로써 전체적인 네트워크 트래픽(traffic)을 감소시키는 효과를 가져온다.

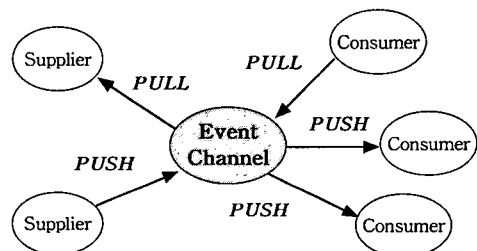
1. 서론

미들웨어(Middleware)는 클라이언트/서버 패러다임(paradigm)에서 클라이언트와 서버 사이의 ' / '에 해당하는 부분으로, 분산 컴퓨팅 환경에서의 클라이언트와 서버를 단일 시스템 환경에서처럼 이 기종의 운영체제, 데이터베이스와 애플리케이션들을 연결해 주는 소프트웨어를 말한다[4]. 분산 환경의 애플리케이션에 적용되어지는 대표적인 미들웨어 기술로 분산된 객체들의 요청을 연결해 주는 ORB(Object Request Broker)가 있다. ORB는 분산 컴퓨팅 환경에서 객체 지향으로 개발된 애플리케이션의 서로 다른 하드웨어 플랫폼, 이종의 통신 프로토콜을 통합하여 객체들간의 통신을 단일한 사용자 환경처럼 지원한다[4]. 이러한 ORB 내에서의 이벤트 서비스는 분산 환경의 시스템들이 요구하는 비동기적 메시지 전송과 하나 이상의 이벤트 공급자가 하나 이상의 이벤트 소비자에게 메시지를 전송하는 것을 지원한다[2][3]. 하지만 기존의 이벤트 서비스는 실시간 애플리케이션들을 지원하기 위해 필요한 우선순위, 마감시간, 실시간 이벤트 필터링 등과 같은 실시간 지원 기능들이 결여되어 있어 이를 지원할 수 있는 실시간 이벤트 서비스 처리가 요구된다.

본 논문에서는 이러한 요구사항들을 지원하기 위한 분산 객체 브로커 내에서의 실시간 이벤트 서비스(RTES : Real-Time Event Service) 처리 방안을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서 관련연구로 이벤트 모델과 CORBA 이벤트 서비스를 분석하고, 3장에서 이벤트 필터링과 QoS 저장소를 포함한 RTES의 전체적인 구조를 설계하고 구성요소의 기능을 설명한다. 4장에서는 본 논문에서 제안한 RTES를 전자상거래에 적용하는 방안을 제시하고, 5장에서 RTES 테스트 프로그램 구현에 대해 설명한 후, 6장에서 본 연구의 결론 및 향후 연구 과제를 제시한다.

2. 관련 연구

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 소비자(consumer)에게 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 중 사용자가 입력하는 이벤트 중 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.



[그림 1] CORBA 이벤트 서비스 구조

기존의 이벤트 서비스 구조로 CORBA 객체 서비스 중의 하나인 CORBA 이벤트 서비스가 있다. COS(CORBA Object Service)이벤트 서비스는 비동기적 메시지 전송을 지원하며, 하나 이상의 이벤트 공급자가 하나 이상의 이벤트 소비자에게 메시지를 전송하는 것을 지원한다. CORBA 이벤트 서비스는 공급자와 소비자, 이벤트 채널의 요소로 이루어져 있는데, 소비자는 공급자에서 발생하는 이벤트를 받는 개체이며, 활성 푸쉬(push) 공급자는 수동적 푸쉬 소비자에게 이벤트를 밀어 보낸다. 마찬가지로, 수동적 풀(pull) 공급자는 활성 풀 소비자가 자신으로부터 이벤트를 끌어 받기를 기다린다. 공급자는 소비자에게 이벤트를 밀어보내기 위해 이벤트 채널을 사용한다. 이벤트 채널은 공급자와 소비자 사이에서 매개체 역할을 수행하는 CORBA 이벤트 서비스의 핵심이다. 그러나, 표준 CORBA 이벤트 서비스에서는 실시간을 지원하기 위해 필요한 이벤트 필터링과 QoS 강요(enforcement), 우선순위, 마감시간과 같은 시간적 제약을 가진 실시간 이벤트 지원 기능이 결여되어 있어 이를 지원하는 실시간 이벤트 서비스가 요구된다. 따라서 본 논문에서는 이러한 요구사항을 지원할 수 있는 실시간 이벤트 서비스 처리 방안을 제시하고자 한다.

3. RTES 구조 설계

3.1 실시간 이벤트 필터링

필터는 입력된 요구 사항에 맞게 데이터를 여과(filtering)해 주는 기능으로, 대표적으로 멀티미디어 통신에서 이미지 데이터의 선택(selective), 변환(transforming), 혼합(mixing) 등의 기능으로 광범위하게 사용되고 있다[1].

기존의 CORBA 이벤트 서비스에서는 공급자(supplier)가 이벤트를 발생시키면 이벤트 채널을 통해 소비자(consumer)에게 이벤트를 밀어 보낸다. CORBA 이벤트 서비스에서 소비자가 단지 공급자로부터의 발생하는 이벤트 중 일부분에만 관심이 있다면, 불필요한 이벤트를 받지 않게 하는 자신만의 이벤트 필터링이 필요하다. 또한 사용자가 입력한 실시간 이벤트가 마감시간과, 우선순위 등의 조건을 반영하여 계속 진행시킬 것인지 진행 전에 중단시킬 것인지를 결정하는 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.

RTES는 이러한 필터링의 기능을 실시간 이벤트 처리에 적용하여 사용자가 원하는 시간적 제약조건에 만족하는 이벤트만을 필터링하여 전송한다. 이를 처리하기 위해 필터링 기능의 선택 기능과 변환 기능을 적용하였는데, 선택 기능은 이벤트가 입력되면 몇 가지 기준을 적용하여 필요한 이벤트를 선택하는 기능으로 실시간 이벤트의 전송을 중단시킬 것인지 진행시킬 것인지를 결정하는데 적용하고, 변환 기능은 입력된 실시간 이벤트의 요구사항과 기존의 사용자로부터 입력된 QoS 요구사항을 반영하여 사용자가 원하는 이벤트로 변환하는데 적용하였다.

3.2 RTES 구성요소와 기능

본 논문에서 제안한 RTES는 이벤트 소비자, 공급자, 이벤트 필터링, QoS 저장소 등으로 구성되어 있으며 각각의 기능은 다음과 같다.

- 이벤트 소비자(consumer)

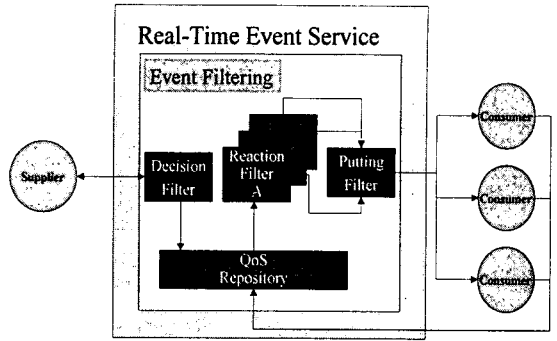
이벤트 공급자가 발생시킨 이벤트를 제공받게 되는 소비자로 자신이 관심있는 이벤트를 선택하고, 원하는 이벤트를 요청할 때 이벤트 결과에 대한 마감시간과 우선순위 등과 같은 실시간 정보를 입력할 수 있다.

- 이벤트 공급자(supplier)

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

- 결정 필터(Decision Filter)

결정 필터는 실시간 이벤트가 입력되면 이벤트를 전송시키기 전에 여



[그림 2] RTES 이벤트 필터링 구조

러 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 것에 한계가 있다. 예를 들어, 소비자가 공급자가 제공하는 이벤트 중 자신이 관심있는 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 RTES에 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.

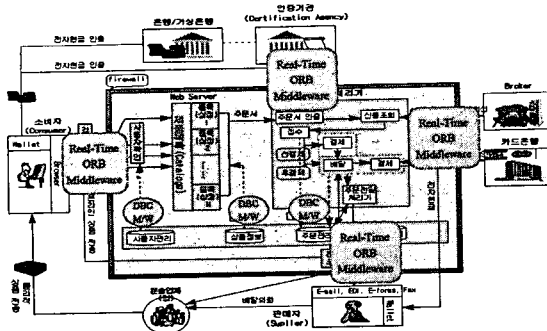
- QoS 저장소 (Repository)

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.

4. RTES를 적용한 전자상거래 구조

[그림 4]는 표준 ORB 구조에 본 논문에서 제안한 RTES를 포함시킨 실시간 ORB 미들웨어를 전자상거래 플랫폼 기능 모델에 적용시킨 구조이다. 전자상거래의 기반 서비스, 응용 서비스, 데이터베이스 서비스 등을 객체지향 개념으로 개발하여 실시간 ORB 미들웨어를 적용시킴으로써 제품 공급자는 객체지향 개념을 이용하여 개발한 자신의 클라이언트 객체를 등록하여 전자상거래 플랫폼과 원활한 정보교환을 할 수 있으며, 향후 다른 공급자나 다른 전자상거래도 쉽게 연동할 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, 전자

와 연결상태를 실시간으로 응답한다. 이 프로그램을 기반으로 필터링 기능을 Plug_In 구조 개발하여 테스트하였다.



[그림 3] RTES를 적용한 전자상거래 플랫폼

상점 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 구현되었던 전자상거래의 병목현상을 피하고, 서버들을 연결된 구조로 부하를 분산시킬 수 있으며, 상호운용성과 이벤트 필터링, 우선순위, 마감시간, QoS 보장 등과 같은 실시간 지원이 가능해진다.

5. RTES 테스트 프로그램 구현

ORB로 RTES 테스트 프로그램을 구현하기 위해서 ORB 미들웨어인 VisiBrokekr for java 3.1을 사용하였고, 언어는 Java (JDK 1.1.5)와 RAD 툴인 JBuilder Client/Server 2.0을 이용해 개발하였다.

ORB로 응용 프로그램을 구현하기 위해서는 ORB 표준 아키텍처를 따르는 개발 업체에서 개발한 ORB 구조를 지원할 수 있는 개발 툴을 이용하여 구현하여야 한다.

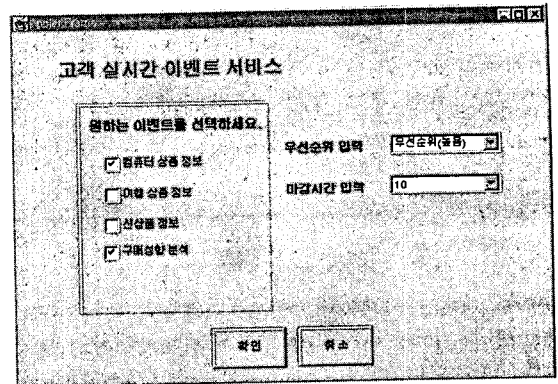
실시간 오퍼레이션은 실시간 이벤트 발생 시 생성되는 오퍼레이션으로 우선순위(priority), 기간(period), 실행시간(execution) 등과 같은 실시간 정보를 말한다[2][3]. ORB 내에서 실시간 이벤트 처리를 위해서는 실시간 오퍼레이션에 대한 인터페이스가 먼저 정의되어야 한다. [그림 4]는 실시간 필터 모듈 IDL (Interface Definition Language)을 나타낸 것이다.

RTES 테스트 프로그램은 ORB를 이용하여 원격의 클라이언트와 원격객체를 연결하고, 실시간 이벤트 발생 시 클라이언트와 서버의 위치

```

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



[그림 5] RTES 테스트 프로그램 실시간 이벤트 입력화면

6. 결론 및 향후 연구 과제

본 논문에서 제안한 RTES는 기존의 ORB 구조에 실시간 이벤트를 처리하는 이벤트 필터링과 실시간 스케줄러, 실시간 매니저 및 QoS 저장소를 포함시킨 실시간 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.

RTES는 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 필터링됨으로써 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] 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.
- [2] 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.
- [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.