

전자상거래를 지원하기 위한 실시간 웹 미들웨어 프레임워크

윤 은 영[†] · 윤 용 익^{††}

요 약

본 논문에서는 전자상거래 시스템에서 요구하는 상호운용성과 실시간 요구사항들을 지원할 수 있는 실시간 웹 미들웨어(RTWM Real-Time Web Middleware) 프레임워크를 제안한다. RTWM 시스템은 전자상거래 시스템의 실시간 요구사항들을 지원하기 위해 CORBA 객체 서비스 구조를 확장하여 이벤트 모니터링, 실시간 스케줄러, 실시간 이벤트 서비스 등의 객체 컴포넌트를 포함하는 구조를 가진다. 특히, 본 논문에서는 전자상거래 사용자의 실시간 요구사항을 지원할 수 있는 실시간 이벤트 필터링을 제공하는 것에 중점을 두었다. 사용자로부터 시간적 제약조건과 원하는 이벤트 경로를 입력받아 QoS 저장소에 저장하고, 실시간 이벤트가 발생하면 RTFA(Real-Time Filtering Agent)를 통해 소비자가 원하는 이벤트를 필터링하여 제공하게된다. 이로써 사용자는 보다 높은 서비스를 제공받게 되고 이 과정을 통해 전체적인 네트워크 트래픽이 감소되는 효과를 얻을 수 있다.

Real-Time Web Middleware Framework for Supporting Electronic Commerce

Eun-Young Yoon[†] · Yong-Ik Yoon^{††}

ABSTRACT

This paper proposes a RTWM (Real-Time Web Middleware) framework for real-time EC (Electronic Commerce) systems. RTWM system is extended the existing COS (CORBA Object Service) model added to the event monitoring, real-time scheduler, real-time event filtering for supporting real-time concept of EC systems. Especially, this paper is concentrated on providing suitable event filtering function for EC system in order to meet various user time requirements under distributed system environment. It stores time constraint requirements and interesting event information input from users into QoS repository, then processes the data through appropriate RTFA (Real-Time Filtering Agent) module when real-time events occur. From this method, users can get the filtered event result reflected their requirements about real-time filtering. It means this system provides high QoS to users. In addition, it results in decreasing network traffic as unnecessary event information is filtered from network.

1. 서 론

최근 웹(World Wide Web)기술은 단순하고 편리한

사용자 인터페이스로 인하여 인터넷상에 분산되어 있는 정보 및 서비스를 접근하는 매체로 폭 넓게 보급되면서 이러한 웹을 기반으로 상거래를 할 수 있는 전자상거래(Electronic Commerce)가 많은 관심을 받고 있다.

※ 본 연구는 정보통신부에서 지원하는 대학기초연구지원사업으로 수행되었음

† 을 회 원 숙명여자대학교 대학원 컴퓨터과학과

†† 중신회원 숙명여자대학교 정보과학부 교수

논문접수 : 2000년 3월 27일, 심사완료 : 2000년 5월 8일

웹을 기반으로 전자상거래 시스템을 구축함으로써 기업과 사용자는 쉽고 편리하게 서비스를 이용할 수

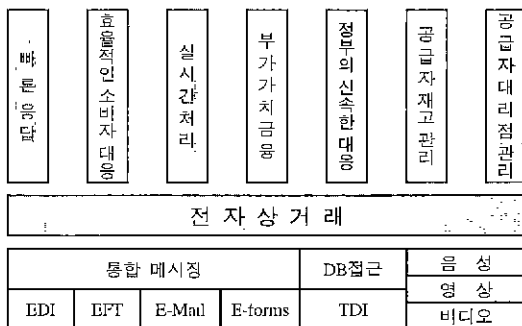
있게 되었지만 상호운용성(interoperability), 이동투명성, 사용자의 실시간 요구, 유지 보수의 어려움 등과 같은 웹 기반 전자상거래 시스템의 문제점을 인식하게 되었다. 또한 지금까지 개발된 전자상거래 시스템들의 대부분은 데이터베이스 정보를 CGI(Common Gateway Interface)와 통신 프로토콜로 HTTP(Hyper Text Transfer Protocol)을 사용하여 간접적으로 데이터베이스를 연결하고 있다. 이는 구축의 편리함과 서버의 부하를 줄여준다는 장점이 있으나 웹과 데이터베이스 연동 상에서 실시간 지원을 제공하지 못하며 올바른 자원의 참조가 이루어지지 않을 가능성이 있다.

본 논문에서는 이러한 문제점들을 개선하기 위해서 분산 객체 기술을 이용한 실시간 웹 미들웨어(RTWM : Real-Time Web Middleware) 프레임워크를 제안하여 웹 기반 전자상거래 시스템에 적용하고자 한다.

2. 관련연구

2.1 전자상거래

전자상거래(EC, Electronic Commerce)에 대한 개념은 약간씩 차이가 있으나, 상거래에 있어서 전자적인 매체 특히, 인터넷을 활용하여 소비자와 공급자를 연결해 주는 시스템이라 할 수 있다[11]. 전자상거래를 구현하기 위해서는 제공하고자 하는 서비스에 따라 다양한 기술을 적용할 수 있다. 넓게는 모든 정보기술이 포함될 수 있으며, 좁게는 구매자와 판매자간의 구매/지불을 위한 홍보, 보안 지불 등의 요소기술이 해당될 수 있다. 전자상거래 구현을 위한 응용분야 및 기술적 참조모델로서 (그림 1)과 같이 정의할 수 있다[11].



(그림 1) 전자상거래 기술 및 응용분야

2.2 미들웨어

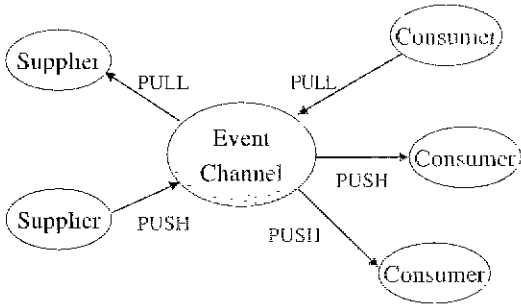
2.2.1 ORB 미들웨어

미들웨어 관련 대표적인 표준으로는 OMG(Object Management Group)에서 제정한 CORBA(Common Object Request Broker Architecture)가 있다. CORBA는 분산 환경에서 여러 시스템을 통합하는데 필요한 기능으로써, 분산 시스템들 간의 운영체제나 프로그래밍 언어에 관계없이 CORBA 객체 서비스(COS : CORBA Object Service)를 제공하여 독립적이고 표준화된 환경을 제공하며 객체간의 통신을 담당한다. CORBA는 객체 지향 개념을 바탕으로 하이 객체의 합수를 호출함으로써 서비스가 이루어진다. CORBA에서는 IDL (Interface Definition Language)이라는 표준언어를 제공하며 클라이언트 객체와 구현 객체 사이의 인터페이스를 정의한다. ORB(Object Request Broker)는 클라이언트와 구현 객체 사이의 증계자 역할을 수행하는 여러 개의 컴포넌트와 인터페이스로 구성되어 있으며 분산된 객체들을 클라이언트/서버 관계로 만들어주는 객체 미들웨어이다.

COS는 광범위하게 CORBA 객체를 조작하는데 필요한 모든 서비스를 지원하기 위한 인터페이스 명세이다[13]. CORBA가 지원하는 여러 객체 서비스 중에서 특정 객체에게 이벤트를 전달하여 새로운 서비스를 요구하거나 수행시키기 위해 제공되는 서비스로 이벤트 서비스가 있다. CORBA 이벤트 서비스는 분산 객체들 사이에서 일어나는 비동기적인 이벤트 전송과 그룹 통신을 유연성 있게 제공하며, 하나 또는 그 이상의 공급자가 하나 이상의 소비자에게 이벤트를 전달할 수 있도록 지원하고 있다. CORBA 이벤트 서비스에는 이벤트를 생성하는 역할을 하는 공급자(supplier)와 객체와 공급자로부터 생성된 이벤트를 받아 처리하는 소비자(consumer), 그리고 공급자와 소비자의 중간 매개 역할을 하는 이벤트 채널(Event Channel)이 존재한다. 공급자로부터 생성된 이벤트는 ORB를 통해 소비자에게 전송된다. CORBA 이벤트 서비스는 멀티캐스트 기능을 제공하여 복수의 이벤트 공급자의 소비자의 이벤트 전송을 지원한다.

CORBA 이벤트 서비스는 (그림 2)와 같이 푸시(push)모델과 풀(pull) 모델 두 가지 형태로 제공된다

[13]. 푸쉬 모델은 이벤트 공급자가 주도적으로 소비자에게 이벤트를 전달한다. 공급자는 이벤트 채널의 push 함수를 실행함으로써 이벤트 채널에 이벤트를 전송한다. 이벤트를 받은 이벤트 채널은 공급자로부터 전송된 이벤트를 원하는 소비자에게 전송한다. 풀 모델에서는 소비자가 주도적으로 공급사의 이벤트를 이용하는 방식이다. 풀 모델에서는 소비자가 이벤트 채널 객체의 pull 함수를 실행함으로써 이벤트 채널에게 이벤트 전송을 요구한다. 소비자로부터 이벤트 전송을 요구받은 이벤트 채널은 공급자 객체에게 이벤트를 요구하고, 이벤트가 있을 경우 이벤트 채널은 통하여 소비자에게 전달된다.



(그림 2) CORBA 이벤트 서비스 모델

2.2.2 웹과 데이터베이스 연동 미들웨어

미들웨어 기술 중 웹과 데이터베이스 연동에 사용되는 기술은 크게 CGI(Common Gateway Interface)를 이용하여 데이터베이스를 간접적으로 연결하는 방법과 JDBC(Java Database Connectivity)를 사용하여 웹과 데이터베이스를 직접적으로 연결하는 방법 등이 있다. 먼저, CGI를 이용한 웹과 데이터베이스 연동 방법을 살펴보면, CGI는 웹에서 외부 시스템을 연결할 때 사용하는 방법의 표준이다. CGI를 사용하면 HTML, 폼 등을 사용해 사용자의 질의를 환경변수를 통해 데이터베이스에 전달 할 수 있다. CGI를 이용한 방법은 쉽고 빠르게 기존의 데이터베이스 프로그램을 웹에 연동 할 수 있기 때문에 초기 많이 이용되었다. 그러나 사용자의 질의를 처리할 때마다 새로운 프로세스를 생성해야 하고, 생성된 프로세스는 데이터베이스 연결과 로그인 과정을 거쳐야 하기 때문에 성능에 많은 문제가 발생한다. 즉, 웹 브라우저와 데이터베이스간의 연결이 지속되지 않고 상태정보가 유지되지 않음. 간접 연결이라

할 수 있다. 초기 대부분의 전자상거래는 간접 연결 방법인 CGI를 이용해 많이 개발되었는데 상태정보를 계속 유지해야하는 전자상거래 특성상 간접 연결 방법은 적합하지 않다고 볼 수 있다. 두 번째로 JDBC를 이용한 웹과 데이터베이스 직접 연결방법을 살펴보면, JDBC 미들웨어는 JDBC를 multi-tier 차원에서 관리하고 다룰 수 있게 해준다. 클라이언트 측에서는 웹서버에서 다운로드 받은 후 JDBC에 직접 연결해 작동하는 JDBC 애플릿이 위치한다. JDBC 애플릿은 데이터베이스에 지속적으로 연결하면서 상태정보를 유지하는 직접 연결 방법을 사용한다. 직접 연결 방법을 이용하면 웹브라우저의 애플리케이션과 데이터베이스간의 연결이 지속되고 상태정보가 유지된다. 따라서 사용자들의 지속적인 대화형 애플리케이션을 작성할 수 있다. 직접 연결 방법의 특징은 웹 브라우저 내에 Java 애플릿이나 액티브X와 같이 다운로드 가능한 동적인 애플리케이션이 필요하다.

3. 실시간 웹 미들웨어(RTWM) 프레임워크

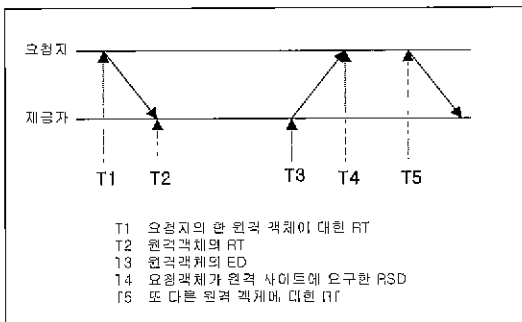
3.1 시간 서비스

분산 환경에서 클라이언트와 서버는 현재 전역 시간을 얻기 위해 터인 서버의 오퍼레이션을 호출하여, 각 시스템을 시간편차 내에서 동기화 시켜야 한다. 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[15]. 여기에서 규정한 객체 시간 서비스는 크게 기본 시간 서비스와 타이머 이벤트 서비스로 나누어진다. 기본 시간 서비스는 애플리케이션 등의 클라이언트에 대해 시간의 취득, 시간의 에러폭의 취득, 두 가지 시간의 비교 및 시간간격의 계산, 오비랩 시간의 확인 등의 기능을 제공한다. 타이머 이벤트 서비스는 시간을 설정 혹은 해제하기 위한 인터페이스로 구성되어 있다. 시간에는 절대시간, 상대시간, 혹은 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.

분산처리를 요하는 시스템의 기본 흐름은 원격적 리요구 단계, 서비스 처리 단계, 결과 전달 단계 등 크

게 3단계로 구분된다. 이들 각 단계마다 필요한 시간적 제약조건이 있으며 이를 명확하게 응용 시스템에서 정의하여 적시성을 보장받도록 해야 한다. 이를 위하여 분산 환경에서 적시성을 보장하기 위한 시간적 제약 조건들을 다음과 같이 크게 4가지로 정의한다.

- 요청 시간 (RT : Request Time)
- 서비스 시간 (ST : Service Time)
- 이벤트 데드라인 (ED : Event Deadline)
- 원격 실행 데드라인 (RSD : Remote Service Deadline)



(그림 3) 시간적 제약조건

RT는 원격 사이트로 서비스를 요청하는 시간을 의미하며, ST는 요청자가 원격 서비스를 포함하여 자신의 수행에 필요한 서비스 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

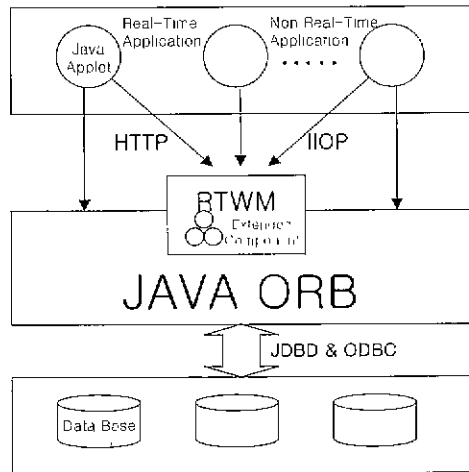
RSD가 분산 실시간 처리 시스템에서 보장되어야 하는 제약조건이다. 이들 간의 관계는 그림에서 보여주고 있는 것과 같이 RSD는 ED 내에 이루어져야 한다. 즉, 이를 위해서는 서비스를 제공하는 서버내의 스케줄러를 포함한 시스템에게 요청자의 요구사항이 반영 되도록 하여야 한다. 분산 실시간 시스템 내에서 RSD의 보장은 적시성 보장의 기준이 된다

3.2 RTWM 프레임워크

RTWM 프레임워크는 전자상거래 시스템의 상호운용성, 웹과 데이터베이스 연동 상에서의 실시간 지원 및 사용자 실시간 요구사항을 지원하기 위하여 웹과 JavaORB 연동 미들웨어를 적용하였으며 이를 기반 환경으로 전자상거래 시스템에서 사용자의 실시간 요구

사항을 지원하기 위해 COS 구조를 확장하여 이벤트 모니터, 실시간 스케줄러, 실시간 이벤트 서비스 등의 객체 컴포넌트를 포함하는 구조를 가진다.

RTWM 프레임워크는 (그림 4)와 같이 세 부분으로 구성된다. 첫째는 클라이언트들로 실시간을 요구하는 애플리케이션과 그렇지 않은 애플리케이션들이 Java 애플릿, HTTP, IIOP 등을 통해 JavaORB로 연결한다. 둘째는 CORBA의 구조를 제공하는 Java ORB이다. Java ORB는 CORBA의 자원 및 객체 서비스를 제공한다. 또한, 애플리케이션들은 애플릿으로써 CORBA의 자원을 직접 활용할 수 있다 셋째는 JDBC와 ODBC 등으로 연결된 데이터베이스로서 웹과 데이터베이스 연동 상에서 실시간 지원을 제공한다. 넷째는 COS 서비스를 확장하여 실시간 지원을 가능하게 하는 RTWM 시스템 구조로서 이벤트 모니터, 실시간 이벤트 서비스, 실시간 스케줄러, 실시간 예외처리 등의 객체 컴포넌트들로 구성되어 있다.



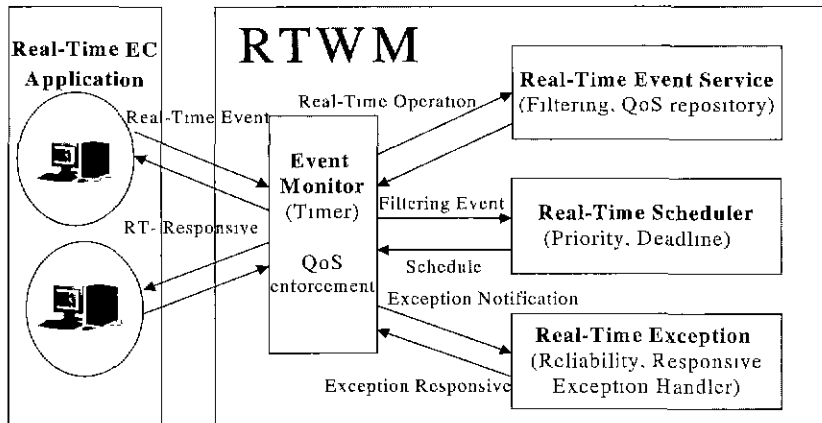
(그림 4) RTWM 프레임워크

3.3 RTWM 구성요소와 기능

사용자의 실시간 요구사항을 지원하기 위해 CORBA 객체 서비스를 확장하여 (그림 5)와 같이 실시간 웹 미들웨어 구조를 설계하였다. 확장된 객체 컴포넌트들은 이벤트 모니터, 실시간 이벤트 서비스, 실시간 스케줄러, 실시간 예외처리 등으로 구성되어 있다.

3.3.1 이벤트 모니터(Event Monitor)

이벤트 모니터는 클라이언트에서 이벤트 컴포넌트 시



(그림 5) RTWM 구성요소

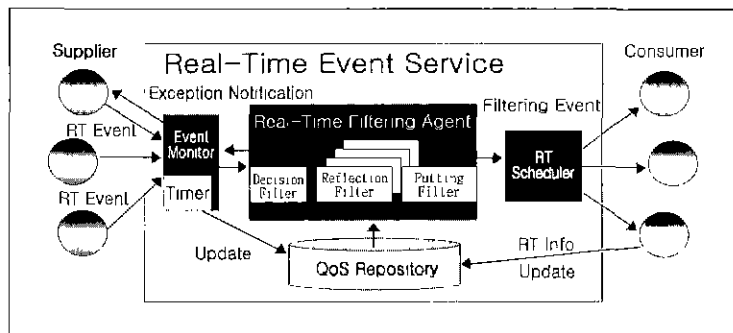
문 역할을 수행하는 컴포넌트이다. 모니터는 이벤트의 참조가 이루어지는 동안에 클라이언트를 인식한다. 즉, 임의의 이벤트에 대해서 참조되는 참조의 수를 인식하여 카운터 작업을 수행한다. 이 카운터는 기존의 CGI로 작성된 카운터와 다르게 실시간에 처리가 이루어진다. 즉, 어떤 자원에 대해서 참조중인 클라이언트의 참조의 수가 증대하면 값이 증대하고 참조가 끝나면 참조의 수가 실시간에 감소된다. 이 과정을 통해 자원이 참조 중인 상태일 경우 수정이나 이동을 방지하여 항상 올바른 자원의 참조가 이루어지도록 한다. 또한, 실시간 이벤트에 대한 정보와 전송시간, 우선순위와 같은 실시간 조건을 모니터링하여 이에 대한 정보를 QoS 저장소와 이벤트 필터링에 전달하며 예외상황 발생 시 사용자에게 실시간으로 예외상황 메시지를 전송한다.

실시간 요구사항을 반영하여 사용자에게 관심 있는 이벤트만을 전송하기 위하여 이벤트 필터링(filtering)기법을 적용하였다. 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[2]. 이러한 필터의 기능을 이벤트 처리에 적용하여 사용자가 입력한 실시간 정보를 바탕으로 이벤트를 필터링하여 원하는 이벤트를 사용자에게 제공하며, 불필요한 이벤트는 사용자가 제공받지 않도록 하였다.

사용자의 실시간 요구사항을 지원하기 위해서는 먼저 사용자로부터 자신이 원하는 정보를 입력할 수 있도록 해야하며, 이 정보를 저장하고 사용자가 원하는 이벤트를 제공하기 위해 이벤트 필터링 과정과 실시간 스케줄링 과정을 거쳐야한다. (그림 6)은 이러한 과정을 나타내는 실시간 이벤트 서비스 구조도이다.

3.3.2 실시간 이벤트 서비스

실시간 이벤트 서비스에서는 전자상거래 사용자의



(그림 6) 실시간 이벤트 서비스 구조

이벤트 필터링 에이전트는 사용자로부터 입력된 실시간 요구사항 및 관심 있어 하는 이벤트 정보를 분석하여 사용자가 원하는 이벤트를 원하는 시간에 제공 가능한지 여부를 결정하고 원하는 정보를 반영하여 변환시킨 이벤트를 전송하는 중계자 역할을 한다. 필터링 에이전트에는 사용자의 요구에 따라 여러 필터가 제공될 수 있으며, 대표적인 필터는 다음과 같다

1) 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 마감시간을 QoS 저장소에 저장된 이벤트 실행시간(worst case time)과 비교하여 입력된 실시간 이벤트를 계속 진행시킬 것인지 중단시킬 것인지를 결정하여 서버로 전송하기 전에 이벤트를 여파시켜 주는 기능을 제공한다.

2) 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 Plug_In 구조로 필터링을 제공한다. (그림 7)은 실시간 필터 모듈 IDL의 일부이다.

3) 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.

```

module RT_Filter
{
    typedef TimeBase :: TimeT Time;

    typedef long Period;
    typedef long Preemption_Priority;
    typedef long Decision_Filter;
    typedef long Reaction_Filter;
    typedef long Putting_Filter;

    struct RT_Info
    { // Exccution times
        Time worstcase_execution_time_,
        Time typical_execution_time_,
        Period period_,
        Preemption_Priority,          Decision_Filter,
        Reaction_Filter,
        Putting_Filter;
        :
    }
}
    
```

(그림 7) 실시간 필터 모듈 IDL

QoS저장소는 이벤트 필터링에 필요한 소비자와 공급자의 정보와 실시간 정보가 저장된다. 이벤트를 공급받게 되는 소비자는 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.

3.3.3 실시간 스케줄러 (Real-Time Scheduler)

전자상거래의 실시간 요구사항을 지원하기 위해서 본 논문에서는 실시간 스케줄링 기법 중에서 동적 스케줄링 기법의 EDF(Earliest Deadline First) 스케줄링 알고리즘을 적용하였다. EDF 스케줄링 알고리즘은 새로운 작업이 도착할 때마다 현재 실행 중인 작업들과 새로운 작업의 마감시간을 서로 비교하여 새로운 작업의 우선순위를 결정한다 이 때 마감시간이 짧을수록 더 높은 우선순위를 부여받게 된다. EDF 스케줄링에서 스케줄 가능성 분석을 위한 조건을 제시하였는데 전체 프로세서 이용률 U를 구하는 식은 다음과 같다

$$U = \sum_{i=0}^N U_i = \sum_{i=0}^N \frac{C_i}{T_i} \leq 1 \tag{1}$$

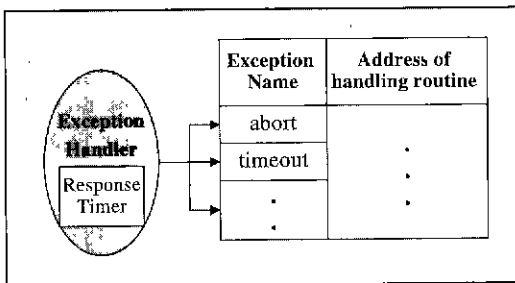
식 (1)에서 U는 프로세서의 이용률, T_i는 작업의 주기 그리고, C_i는 실행시간을 나타낸다. 이 조건식은 필

요 충분 조건으로서 정확한 스케줄 가능성 분석 방법을 제공하며, 이는 또한 EDF 스케줄링 기법이 우선 순위 선점형 스케줄링 기법 중에서 최적의 스케줄링 방법임을 의미한다고 할 수 있다

3.3.4 실시간 예외 처리(Real-Time Exception)

분산된 실시간 객체 처리 중에는 마감 시간을 넘거나 네트워크 오류 등으로 예외 상황이 발생하게 된다. 실시간 예외 처리 서비스는 예외 상황 발생 시 이를 처리하여 신뢰성을 보장할 수 있도록 지원한다.

예외 상황이 발생하였을 때 오류가 발견된 객체는 원자성(atomicity)을 보장하기 위하여 실행을 중지(abort)하게 된다. 원자성을 유지하기 위해서는 중지된 객체가 데이터베이스 상태나 프로그램 수행 상태에 영향을 주지 않도록 해야 한다. 프로그램을 중지시키는 가장 일반적인 방법은 중지된 프로그램을 취소하고 그 이전 단계로 되돌리는 undo operation을 제공하는 것이다. (그림 8)은 실시간 예외처리의 구조를 나타낸다.



(그림 8) 실시간 예외 처리 구조

예외 핸들러는 예외 상황 메시지가 통보되면 그 예외 상황을 처리할 수 있는 예외 처리 루틴에 매핑하는 역할을 한다. 또한 응답 타이머는 예외 처리에 대한 결과를 빠른 시간에 클라이언트에게 제공하기 위하여 시간을 조절하는 기능을 한다.

4. 전자상거래 실시간 웹 미들웨어 적용방안

4.1 RTWM을 적용한 전자상거래 플랫폼

CORBA는 객체지향 프로그램으로 구현된 분산환경에서 클라이언트와 서버간에 객체를 주고받을 수 있도록 해주는 미들웨어이다. 전자상거래 플랫폼에서는 다양한 소비자의 공급자가 존재하게 되는데 이러한 분산환경

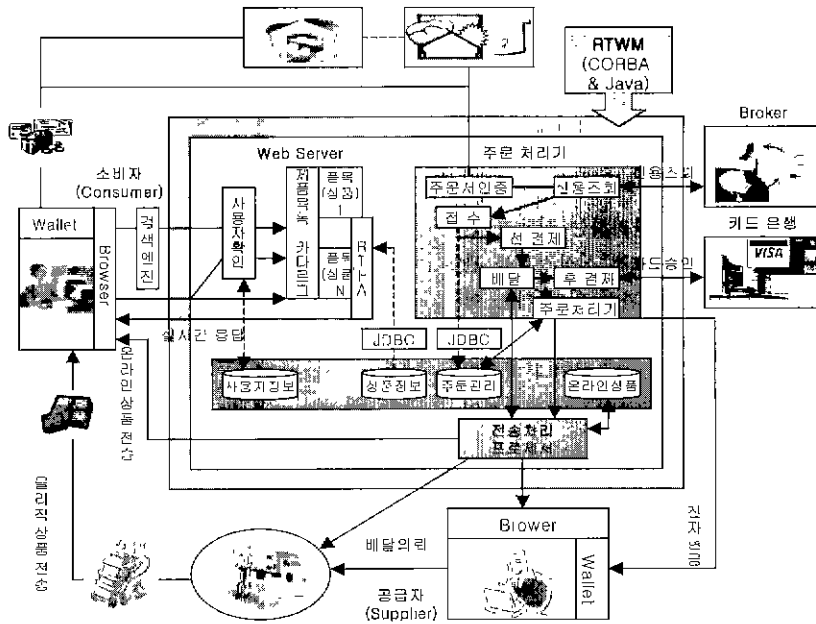
에서는 Plug & Play 형태로 활용할 수 있는 CORBA의 이용이 효과적이다. 전자상거래 구현 시 CORBA 미들웨어를 적용하여 구현하면 플랫폼 독립성 (Platform Independence), 상호운용성(Interoperability), 재사용성, 이식성(Portability), 위치 투명성(Location Transparency), 유연성(Flexibility) 등의 장점이 있다.

RTWM은 전자상거래의 각 과정에 적용될 수 있는데, (그림 9)에서의 같이 전자상거래의 구성요소인 소비자와 공급자, 은행, 인증기관 등이 CORBA-Java 기반으로 구축하면 각 구성요소들이 하나의 통합된 환경에서와 같은 사용 환경을 제공받게 된다.

전자상거래의 기반 서비스, 응용 서비스, 데이터베이스 서비스 등을 객체지향 개념으로 개발하여 RTWM을 적용시킴으로써, 제품 공급자는 객체지향 개념을 이용하여 개발한 자신의 클라이언트 객체를 등록하여 전자상거래 플랫폼과 원활한 정보교환을 할 수 있으며, 향후 다른 공급자나 다른 전자상점과도 쉽게 연동할 수 있으며, 소프트웨어 모듈별 업그레이드 및 재사용이 가능하게 된다. 다시 말하면, 클라이언트 및 서버 객체를 객체지향 개념을 이용하여 개발하고 이를 CORBA 표준에 따라 개발하면 구매자, 판매자, 전자상점 운영자가 프로그램을 활용할 수 있다 또한, 전자상거래의 구매자, 판매자, 인증기관, 브로커, 카드은행 등 많은 거래 요소들이 RTWM을 이용해 전자상거래의 병목현상을 피하고, 서버들을 연결된 구조로 부하를 분산시킬 수 있으며, RTFA를 통해 소비자는 실시간 이벤트 필터링과 이에 따른 실시간 응답을 제공할 수 있게된다. 또한, JDBC 미들웨어를 이용하여 실시간 데이터베이스 연동 등과 같은 실시간 지원이 가능해진다. 다음은 전자상거래 각 기능을 살펴보고 RTWM 적용방안을 제시한다.

4.1.1 광고 및 쇼핑

구매자는 웹 브라우저를 사용하여 전자상거래에 접근하게 되고 웹 브라우저 상에 나타난 텍스트, 이미지, 사운드를 비롯한 동영상 등 다양한 형태로 제공되는 제품 카다로그를 이용하여 쇼핑을 하게된다. 제품을 홍보하기 위한 이러한 정보는 판매자가 웹을 이용하여 전송함으로써 적절한 전자상거래 내부 절차를 거쳐 상품으로 등록되게 되며, 웹과 데이터베이스를 연결해주는 미들웨어를 통해 소비자에게 제공된다.



(그림 9) RTWM을 적용한 전자상거래 플랫폼

이 과정에서 실시간 필터링 에이전트(RTFA)는 구매자가 쇼핑 시 자신이 원하는 제품정보만을 요청해서 제공받을 수 있고, 구매자가 입력한 실시간 요구사항에 맞추어 이벤트형 필터링하여 제공하며, 마감시간을 지킬 수 없는 등의 예외상황이 발생하면 구매자에게 예외상황을 실시간 응답한다. 또한 구매자의 구매성향을 분석하여 추천 상품 정보를 제공하는 등의 사용자에 높은 서비스를 제공하는 것에 적용될 수 있다. RTFA는 구매자뿐만 아니라 판매자에게도 자신의 상품정보를 원하는 시간에 구매자에게 전송하는 서비스를 제공하며, 상품정보를 필요로 하는 소비자에게만 상품정보를 전송하게 하는 필터링 서비스도 적용될 수 있다.

4.1.2 주문

구매자가 쇼핑을 하다가 구매하고 싶은 상품을 선택하였을 때 장바구니 기능을 하는 임시 저장 장소에 주문할 상품과 수량을 입력하고 최종 주문서를 작성하기 전까지 구매자는 장바구니에서 주문 내역을 조회하여 변경할 수 있다. 구매자가 주문을 한 경우 주문 내용에 이상이 없을 경우에는 신용조회를 한다. 신용조회에는 대금을 카드로 결제하는 경우 브로커를 통해 카드회사로부터 구매자의 카드사용 관련 신용도에 대한

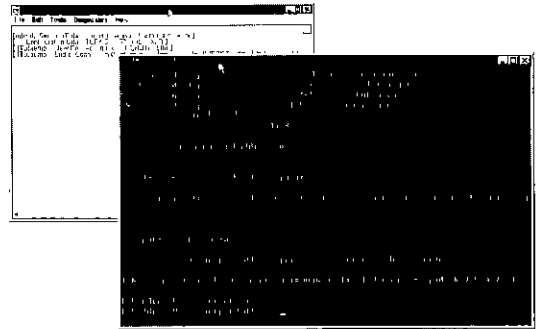
응답을 받는 것과, 전자현금을 사용하는 경우 계정을 이유 없이 반송하거나 계약을 취소하는 등의 불량거래자인지를 소비자 정보관리 데이터베이스를 통해 확인하는 것 등이 포함된다. 주문의 내용이 이상 없고, 주문을 의뢰한 구매자의 신용이 이상 없을 경우에만 구매자의 제품 주문내용은 접수되어 주문관리 데이터베이스에 저장된다. 이 과정에서 데이터베이스 연동 이상내경보를 계속적으로 유지할 수 있는 JDBC 미들웨어를 적용하는 것이 바람직하다.

4.2 RTWM 테스트 프로그램 구현

본 논문에서 제안한 RTWM 프레임워크의 검증을 위해 전자상거래에 적용한 테스트 프로그램을 구현하였다. 개발환경은 다음과 같다. CORBA의 핵심이 되고 있을 수 있는 IDL은 Java 매핑방법을 이용하여 컴파일 하였으며, OrbixWeb3.1을 사용하였다. 구현체와 클라이언트는 웹 프로그래밍 언어인 Java를 이용하여 구현하였으며 JDK1.x.x를 이용하였다. 요청한 상품에 대한 데이터 저장을 위해 DBMS는 MS SQL Server를 사용하였으며, 이때 ODBC드라이버를 통해 DBMS에 연결할 수 있게 하였다. 또한 Java 구현체에서 ODBC에 연결하기 위해 JDBC를 사용하였다. 서버와 DBMS는 Pentium-500

Windows NT 환경에서 실행시켰으며, 클라이언트는 Pentium-500 Windows 95 환경에서 테스트해 보았다. (그림 7)은 Java로 작성된 전자상거래의 고객입력 화면이다. 데이터베이스 서버는 Windows NT 4.0 환경의 MS SQL Server를 사용하였으며 JDBC-ODBC로 연결하였다.

(그림 10)은 전자상거래 고객을 관리하는 사용자 인터페이스 화면으로 고객이 입력한 고객 신상정보를 관리한다. (그림 11)은 고객이 자신이 관심 있어 하는 이벤트 정보와 실시간 요구사항을 입력하는 화면이며 이 정보를 바탕으로 이벤트 필터링이 이루어진다. (그림 12)는 클라이언트/서버 컴퓨터가 CORBA를 이용하여 등록되고, 연결된 것을 보여주고 있다.



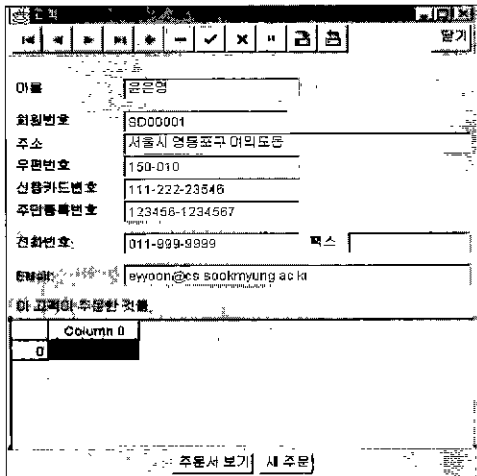
(그림 12) 서버 연결 화면

이 테스트 프로그램에서는 분산된 웹 기반 전자상거래 시스템의 클라이언트와 서버를 JavaORB 미들웨어 및 JDBC-ODBC 미들웨어를 이용하여 연결하였으며 고객이 입력한 이벤트 정보를 반영하여 필터링되는 과정을 테스트하였다.

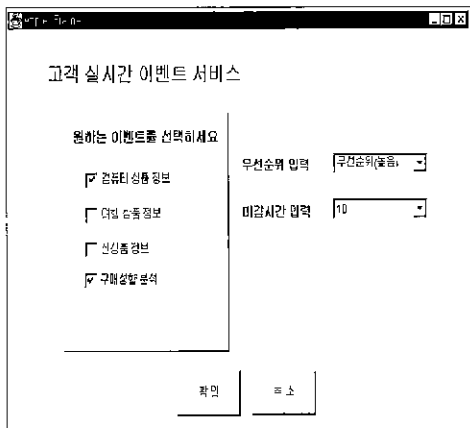
5. 결 론

사용하기 편리한 인터페이스를 제공하는 웹 기술의 발전으로 인터넷을 비즈니스 측면에서 활용하는 전자상거래가 활성화되면서 전자상거래 시스템을 보다 효율적이고 체계적으로 구현하기 위한 연구가 세계적으로 진행되고 있다. 전자상거래와 같은 웹 기반의 분산 실시간 애플리케이션들은 상호운용성을 지원하는 분산 객체 환경과 웹과 데이터베이스 연동 상에서 실시간 지원 및 사용자의 실시간 요구사항을 반영할 수 있는 실시간 웹 미들웨어 프레임워크가 요구된다.

본 논문에서 제안한 RTWM 시스템은 웹 기반의 전자상거래 시스템들의 상호운용성과 데이터베이스 연동 상에서의 실시간 지원을 위해 웹 미들웨어인 Java ORB와 JDBC를 이용하였으며, 사용자의 실시간 요구사항을 지원하기 위하여 CORBA 객체 서비스를 확장하여 실시간 스케줄러와 실시간 이벤트 서비스 등을 포함하는 구조를 제안하였다. 특히, RTWM 시스템은 사용자의 실시간 요구사항을 웹 기반 분산 시스템 환경에 반영시키기 위하여 이벤트 필터링 기능을 실시간 처리에 맞게 강화시켜 기존의 시스템에서 지원하지 못한 사용자의 QoS 강요와 실시간 요구사항을 지원하는 데 중점을 두었다. 이벤트 필터링 기능을 제어하는 RTPA는 사용자로부터 입력된 시간적 제약조건과 이벤트 필터링 정보를 QoS 저장소에 저장하고 실시간 이벤트가 발생하면 이 정보를 반영하여 이를 지원할



(그림 10) 고객 관리 화면



(그림 11) 실시간 이벤트 입력화면

수 있는 필터를 선택하여 적용함으로써 사용자가 원하는 맞춤 이벤트만을 필터링하여 제공하게 된다. 이로써 사용자는 자신이 원하는 요구사항에 맞게 필터링된 이벤트 처리 결과를 얻게 되어 궁극적으로 보다 높은 서비스를 제공받게 된다. 또한 이 과정을 통해 불필요한 이벤트가 전송 전에 필터링 됨으로써 전체적인 네트워크 트래픽이 감소되는 효과를 얻을 수 있다.

본 논문을 초석으로 하여 향후에는 웹 기반 분산 환경에서 실시간 멀티미디어 미들웨어 프레임워크를 연구하고자 한다 이를 위해서는 본 논문에서 제안한 RTWM 시스템의 기능 확장 연구가 필요하며, 멀티미디어 객체들에 대한 특성 분석과 멀티미디어 서비스 기능에 대한 연구가 필요하다

참 고 문 헌

[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 Multiplex 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.

[11] David L. Taylor. "Electronic Commerce Strategies : Five Year Scenario," Gartner Group, 1995

[12] G. P. A. Fernandes and I. A. Utting, "An Architecture for Scheduling of Services in a Distributed System," 1996.

[13] OMG, Event Service Specification, 1995, <<http://www.omg.org>>



윤 은 영

e-mail : eyyoon@cs.sookmyung.ac.kr
 1997년 숙명여자대학교 전산학과 졸업(이학사)
 1999년 숙명여자대학교 대학원 전산학과 졸업(이학석사)
 1999년~현재 숙명여자대학교 대학원 컴퓨터과학과 박사과정 재학 중

관심분야 : 전자상거래, 분산 실시간시스템, 분산 데이터베이스 시스템, 분산 미들웨어 시스템, 멀티미디어 통신



윤 용 익

e-mail : yiyoon@sookmyung.ac.kr
 1983년 동국대학교 통계학과 (이학사)
 1985년 한국과학기술원 전산학과 (공학석사)
 1994년 한국과학기술원 전산학과 (공학박사)

1997년~현재 숙명여자대학교 정보과학부 교수
 관심분야 : 정보통신, 멀티미디어통신, 분산시스템, 실시간 처리 시스템, 분산 미들웨어 시스템, 분산 데이터베이스 시스템, 실시간 OS/DBMS