

CORBA 기반의 전자상거래 시스템 구축 기법 및 전략

김수동*, 김철진**
*송실대학교 전자계산학과, **오브젝소프트(주)
Soo Dong Kim*, Chul Jin Kim**
Department of Computer Science, **ObjectSoft Corporation
*Soong Sil University

요 약 문

여러 산업 분야에서 인터넷의 채택으로 인터넷을 기반으로 하는 전자상거래 시스템 구축이 일반화되고 있으며, 전자상거래에 연관된 표준들이 제기되고 있다. 전자상거래가 인터넷의 네트워크를 고려한다면, 전자상거래 시스템은 클라이언트/서버나 분산 아키텍처를 요구한다. 즉, 객체지향 패러다임이 새롭고 효율적인 개발 방법으로써 널리 받아들여져 왔다. OMG의 CORBA가 객체지향 분산 아키텍처를 위한 실제적인 표준으로 잡음으로써, 진보된 인터넷 상거래 시스템들 구축하는데 있어 CORBA를 이용하기 위한 기술을 요구한다. 본 논문에서는 CORBA 기반의 전자상거래 시스템의 아키텍처를 제시하며, CORBA가 인터넷 상거래 시스템의 성능과 다른 특성들을 어떻게 향상시킬 수 있는지를 제안한다.

1. 서 론

1.1 연구 배경

인터넷의 보급은 산업계에 상당한 영향을 끼치고 있다. 90년대 들어서면서 많은 기업들이 인터넷을 기반으로 한 소프트웨어 개발에 박차를 가하기 시작했으며, 회사 조직내의 환경을 인트라넷(Intranet)으로 구축하고 있다.

95년 가을 Java 언어가 소개되기 이전까지 대부분의 회사들은 인터넷 기반의 웹 어플리케이션을 개발하는 데 있어서 CGI와 같은 형태를 채택하였다[1]. 즉, 사용자 요청에 대한 결과는 Web Server로부터 HTML 문서 형태로 생성되며, CGI 프로그램을 통하여 계산 및 데이터베이스 인터페이스를 제공하는 형태이다. CGI와 같은 방식의 웹 어플리케이션은 많은 문제점과 제약사항을 지니고 있다. 클라이언트 컴퓨터는 모든 정보의 검색과 처리를 서버에 요청해야 하고, 서버는 CGI 프로그램을 실행하여 그 결과를 클라이언트 컴퓨터에 보내게 된다. 그 결과는 클라이언트 컴퓨터의 웹 브라우저를 통해 출력 된다. 따라서, 클라이언트 컴퓨터는

단지 그래픽 터미널 역할에 불과한 단순 기능만 수행하게 되므로 효율성이 저하되게 된다. 서버는 여러 클라이언트들의 요청들을 받아서 처리해야 하는 모든 기능을 담당하게 되므로 네트워크 트래픽(Network Traffic)의 집중화와 프로세싱(Processing)의 과중으로 인한 성능 저하의 문제를 안고 있다. 또한 CGI와 같은 방식은 네트워크를 지속적으로 연결해 주지 못하기 때문에 웹 클라이언트의 트랜잭션 관리를 효율적으로 처리해 주지 못한다.

이와 같은 전통적인 클라이언트/서버 모형의 문제점들을 해결하기 위한 대안책으로 여러 기술들이 지난 2~3년 간에 걸쳐 소개 되고 있다. 예를 들면, Java 언어의 도입으로 클라이언트 컴퓨터에서의 프로세싱 기능이 가능하게 되었고, OMG(Object Management Group)에서 분산 객체 컴퓨팅의 표준으로 제시한 CORBA(Common Object Request Broker Architecture)를 이용하여 데이터와 프로세싱 분산이 가능하게 되었다.

Java와 CORBA 등의 신기술은 모두 객체지향 패러다임(Object-Oriented Paradigm)을 취하고 있으며 이러한 신기술의 보급으로 최근에는 객체지향 방식의 웹 어플리케이션 개발이 보편화되고 있다. 따라서, 객체지향 기술을 적용한 클라이언트/서버 모형은 기존의 절차적 클라이언트/서버 모형이 지니고 있는 문제점이나 제약 사항들을 해결할 수 있으며, 성능, 효율성 및 재사용성 측면에서 소프트웨어 품질 향상에 크게 기여할 수 있다.

본 논문에서는 객체지향 클라이언트/서버 및 분산 아키텍처를 기반으로 전자상거래의 시스템을 구축하기 위한 기법으로 다양한 지불 방식을 제공할 수 있는 지불 프레임워크 구축 기법, 객체지향 분산 트랜잭션을 관리할 수 있는 기법, 그리고 효율적으로 데이터를 분할할 수 있는 기법을 제시한다.

II. CORBA 기반 전자상거래 구축 기법

2.1 Payment System

전자 상거래에 있어서 전자결재는 매우 중요한 부분을 차지하고 있다. 전자결재에 대한 완벽한 지원이 이루어지지 않는다면 신뢰할 수 있는 전자 상거래가 이루어 질 수 없기 때문에 이에 대한 완벽한 해결책이 제공 되어야 한다. 현재까지 10여가지 이상의 전자거래 프로토콜이 소개 되었으며, 앞으로도 많은 프로토콜이 선보일 것으로 예상된다. 현재 관련 제품은 대부분 특정 프로토콜만 지원하거나, 개방형 구조를 가지고 있더라도 프레임워크 방식의 접근은 하지 않고 있다. 따라서 특정 프로토콜만 지원하는 것보다 모든 프로토콜을 지원하고 향후 어떤 프로토콜이 소개 되더라도 유연하게 대처할 수 있는 프레임워크 구조를 가지고 있는 제품이 필요하다. 현재까지 소개된 여러 가지 전자지불 프로토콜은 다른 프로토콜과 맞물려서 전자거래가 이루어지는 방식을 지원하지 않고 있다. 따라서 이를 중간에서 변환시켜주는

Adapter가 존재한다면 다양한 사용자가 전자거래를 하려고 할 때 특정 프로토콜이 지원되지 않아서 거래를 하지 못하는 등의 문제는 해소 시킬 수 있다.

전자지불 시스템은 Java로 구현하여 모든 플랫폼에 변형 없이 적용 가능하며, 프레임워크 형태로 구현되어 전자 상거래 시스템 구축 시 Plug-in 형태로 인터넷 전자 상거래 지불 프로토콜인 SET, Mondex, DigiCash, E-Cash를 기본적으로 제공할 수 있다.

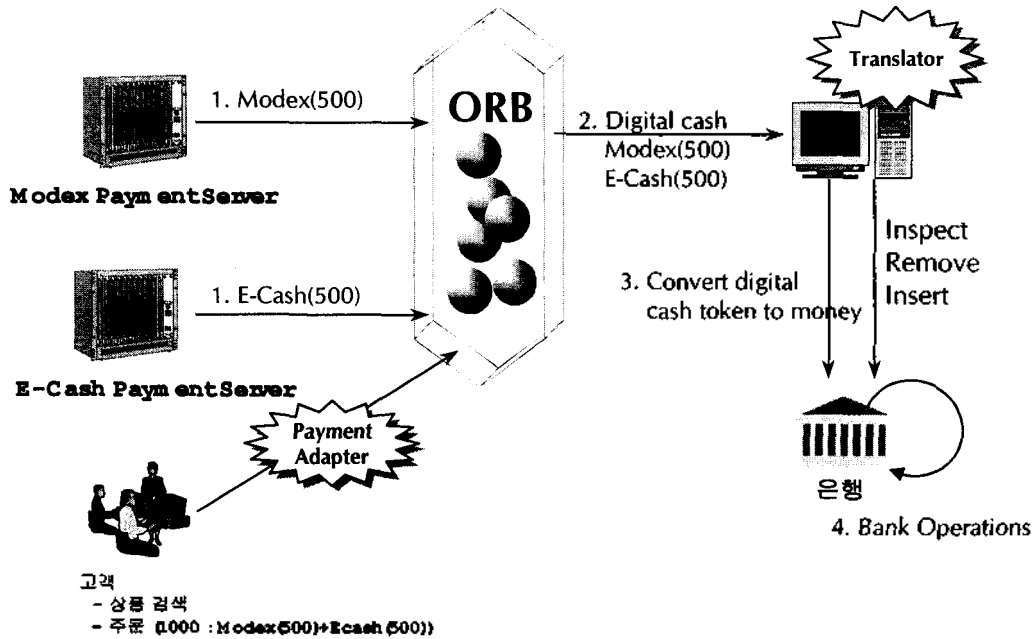


그림 1. ORB 상에서 Multiple Payment 처리 흐름

그림 1에서와 같이 다중의 전자지불 방식을 가지고 있는 고객이 제품에 대한 지불 방식을 Modex와 E-Cash로 할 경우, 해당하는 ORB를 통해 다중 지불 서버(Modex, E-Cash)로부터 전자 화폐를 서로의 지불 화폐와 연결될 수 있도록 변환기를 통해 토큰으로 변경시켜 준다. 이렇게 변경된 전자 화폐의 토큰은 은행서버를 통해 은행 처리를 한다.

○ Adapter 기능

블랙박스 프레임워크 형태로 내부적으로 SET, E-Cash, DigiCash, Mondex를 기본적으로 제공하고 프레임워크 사용자가 원하는 방식을 선택하여 사용할 수 있는 제공한다. 지불 Adapter는 여러 지불 방법이 가지고 있는 공통된 기능을 추상 클래스화하여 트랜잭션이 이 추상클래스에 정의된 동일 메소드 이름을 이용하여 여러 지불 방법에 접근할 수 있도록 하였다. 그림 2 처럼, Payment 추상클래스는 하부 Bank Payment(은행 계좌 이용 지불 방법), Credit Card Payment, SET 기반 Payment 방식, E-Cash, DigiCash, Mondex Payment의 공통 기능들인 인증, 지불 요구, 지불 반환 요구, 총계 확인, 현재 상태(State) 확인, Abort 등을 RequestAuthorization(), RequestPayment(), RequestReversePayment(), RequestState(), RequestAmount(), Abort() 메소드들로 표현하고 있다. 여기서 E-Cash, DigiCash, Modex Payment 지불 방법은 다른 3가지 지불 방법과

달리 인증을 얻기 위해 재정 기관(은행)을 접근하지 않고 전자화폐의 서명(Signature) 인증 대행 기관의 인증을 받는다.

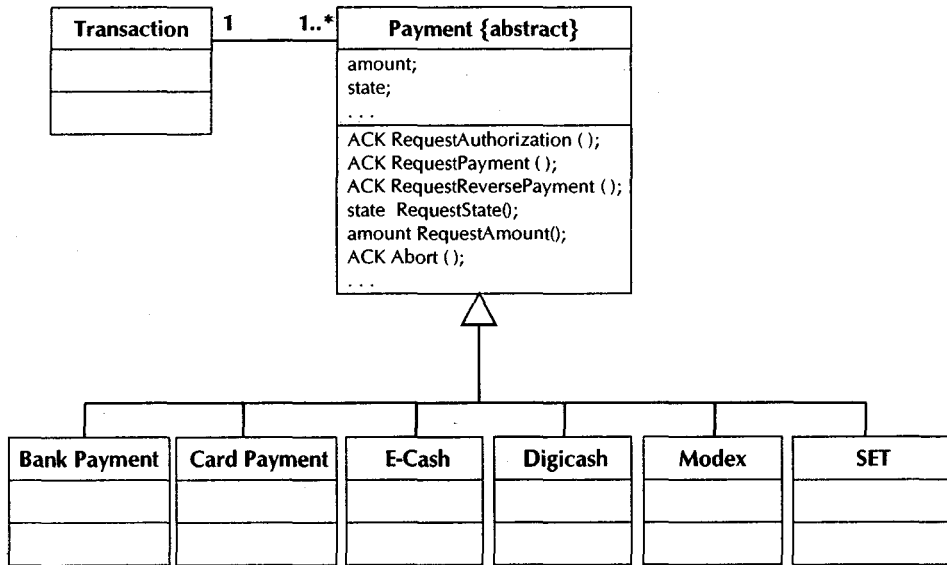


그림 2. Payment Adapter 구조

제시된 Payment 추상 클래스를 이용하여 새로운 지불 방법이 추가되더라도 Payment 클래스를 상속하여 쉽게 지불 방법을 구현, 적용할 수 있으며 구축 시 동일한 메소드 이름을 사용하여 여러 지불 방법에서 동일한 기능을 수행하게 함으로써 코드의 이해성을 높이고, 구축을 쉽게 할 수 있다[2][3][4][5][6].

2.2 Object Transaction Processing Service

전자상거래의 트랜잭션 처리는 상이한 네트워크에서 비정형적으로 분산되었거나, 다중의 객체들 간에 트랜잭션 처리를 수행하기 위한 CORBA 스펙인 OTS(Object Transaction Service)를 기반으로 개발할 수 있다[7][8].

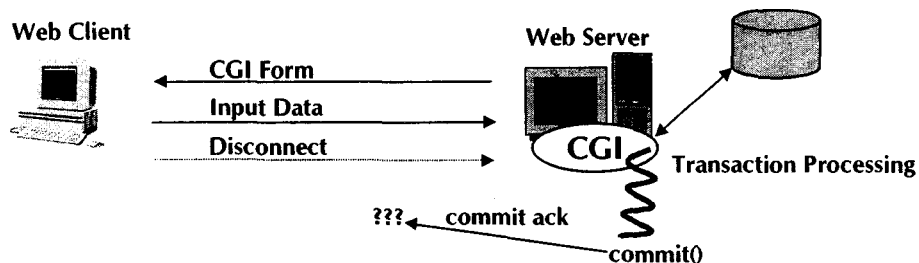


그림 3. Connectionless 방식에서의 트랜잭션 처리

그림 3에서와 같이 기존의 네트워크가 지속되지 않는 연결 방식에서의 트랜잭션 처리는 클라이언트의 트랜잭션 처리를 관리할 수 없다. 예를 들면, 전송된 CGI 폼에 고객이 원하는 상품을 주문하고 주문 정보들이 서버의 CGI 프로그램에서 트랜잭션을 처리할 수 있다. 이때 commit이나 Rollback 되었을 때, 이러한 트랜잭션 처리 결과를 클라이언트로 보내야 하지만, 네트워크 연결이 끊긴 상태이므로 클라이언트는 제대로 트랜잭션이 이루어졌는지 알 수 없다.

이와 같이 기존의 connectionless 방식의 트랜잭션 처리와는 다르게 분산 환경에서 어플리케이션의 트랜잭션을 관리할 수 있는 방식으로 웹 클라이언트에서 발생하는 트랜잭션을 관리할 수 있는 트랜잭션 방식을 객체지향 언어인 자바를 이용하여 가능하게 할 수 있다. 웹 상에서 클라이언트의 트랜잭션 관리는 제한적이거나 불가능하지만 OTS에 기반을 둔 JavaSoft의 JTS(Java Transaction Service)를 이용하면 클라이언트의 트랜잭션 관리가 용이하도록 할 수 있다. 그러나 아직까지 JTS를 이용하여 클라이언트의 트랜잭션 처리를 관리해 줄 수 있는 제품이 없다. 따라서 CORBA의 ORB 핵심 기능인 인터셉터(Interceptor)[기]를 이용하여 클라이언트의 트랜잭션 상태 정보를 저장하여 실패가 일어났을 때, 클라이언트에 저장된 트랜잭션 상태 정보를 이용해 rollback 할 수 있다.

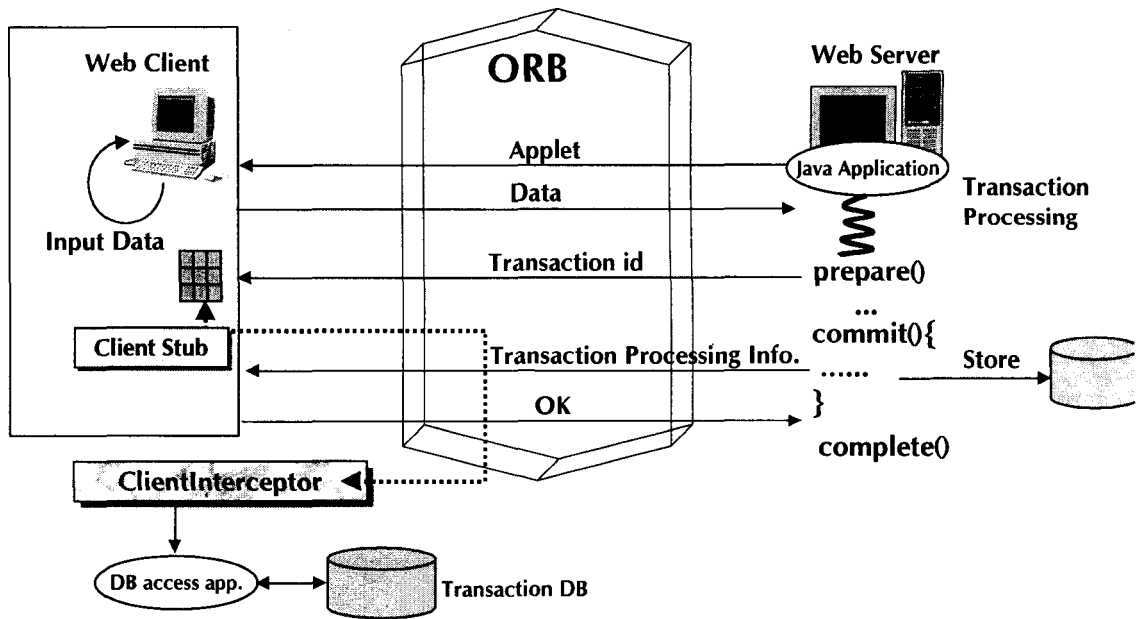


그림 4. 웹 상에서의 트랜잭션 관리

그림 4는 웹 상에서 클라이언트의 트랜잭션을 관리할 수 있는 기법이다. 서버에서 트랜잭션 처리 중 prepare가 완료되었을 때, 그 시점에서 서버는 트랜잭션을 나타낼 수 있는 정보(transaction Id)를 클라이언트로 전송한다. 클라이언트는 이 트랜잭션 정보를 지역적으로 저장하여 현지점의 트랜잭션 상태를 파악할 수 있다. 이 때 애플릿의 security 제약 사항으로 인해 클라이언트 호스트에 트랜잭션 상태 정보를 저장할 수 있는 방법이 없으므로 웹 클라이언트의

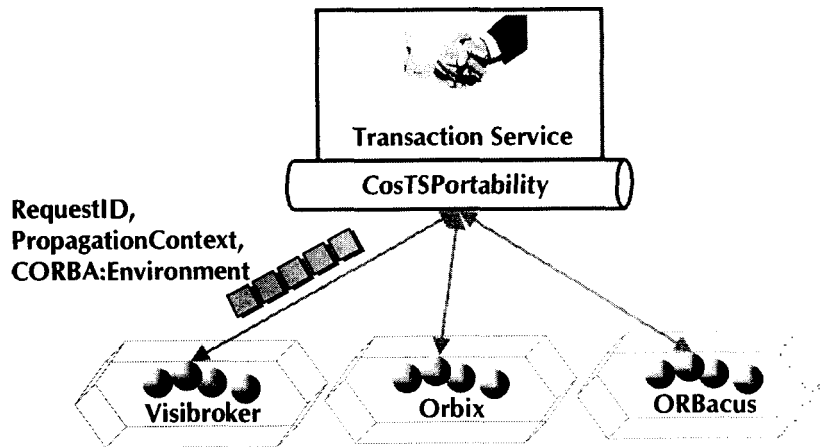


그림 6. 트랜잭션 서비스의 Portability

그림 6에서와 같이, 트랜잭션 요청이 처음 전송되기 전에, 트랜잭션 콜백(transaction callback)을 설정할 수 있도록 트랜잭션 서비스를 특정 ORB 에 명시한다.

2.3 Data Balancing

자바 언어를 통해 클라이언트는 서버의 프로세싱을 분담할 수 있으며, CORBA 를 기반으로 한 분산 환경에서 클라이언트 호스트는 다중의 서버로부터 필요로 하는 정보를 요구할 수 있으며, 지역적으로 클라이언트에서 데이터를 관리할 수 있으므로 데이터의 분할을 이룰 수 있다. 또한 데이터 요청을 통한 잦은 네트워크 트래픽을 줄일 수 있다.

전자상거래에서 고객은 웹 브라우저에서 주문한 상품을 장바구니에 관리하고 계산 처리를 서버로 보내서 처리할 필요가 없이 지역적으로 처리하여 실제로 주문을 원할 경우에 계산된 처리를 서버로 보내므로 서버의 부담과 네트워크 트래픽을 감소시킬 수 있다. 그러나 웹 상에서 서버에 등록된 고객의 정보(예를 들면 고객의 주민번호, 성별, 거래 은행, 계좌번호, 등)를 애플릿이 초기화 될 때마다 재전송해야 하므로 서버에 많은 부하를 준다.

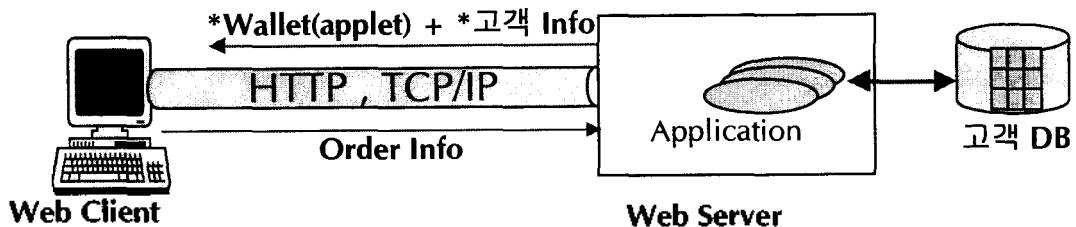


그림 7. Wallet 애플릿을 이용한 데이터 관리

그림 7은 전자상거래에서 자바 애플릿을 이용해 데이터를 분산시키기 위한 기법으로 전자상거래에서 고객 정보나 상품 정보가 네트워크를 통해 자주 전송되므로 네트워크 트래픽으로 인한 전자상거래 서비스의 성능을 저하시킨다. 따라서 자주 전송될 필요가 없는 정보, 즉 상품 정보나 고객 정보를 지역적으로 관리하여 성능을 향상시킬 수 있다.

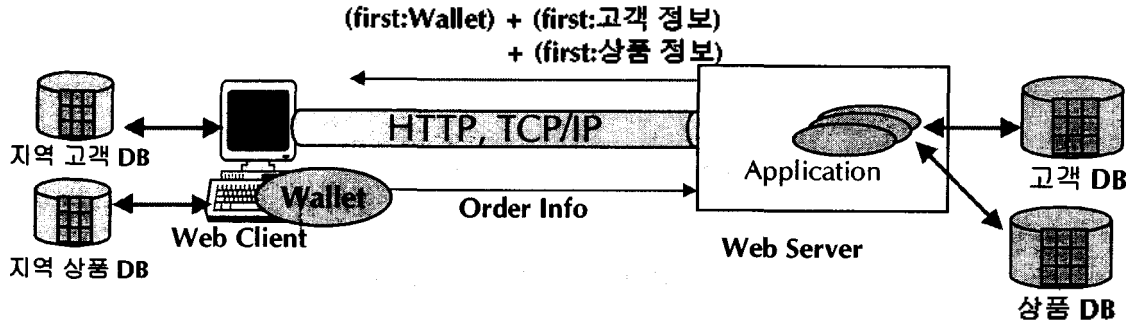


그림 8. Wallet 어플리케이션을 이용한 지역 데이터 관리

그림 8은 전자상거래 Wallet 을 클라이언트 호스트에 설치하는 방식으로 고객이 처음 등록할 때에서 서버로부터 Wallet 어플리케이션을 전송 받아 클라이언트에 설치한다. Wallet 자체는 자바의 애플릿이 아니라 어플리케이션이므로 애플릿의 Security 제약 사항으로 인한 지역 데이터의 접근의 문제를 극복할 수 있다. 이렇게 지역적으로 Wallet 어플리케이션이 설치될 때, 초기 고객 정보를 지역 데이터베이스에 저장하고, 실제 주문이 이루어 질 때 고객 데이터베이스 이용해 거래를 한다. 또한 초기에 상품 정보를 Wallet 과 함께 설치하고, 후에 갱신되는 상품 정보는 상품별로 개별적으로 전송 받아 지역 데이터베이스를 최신의 데이터로 유지할 수 있다. 이러한 고객의 호스트에 Wallet 을 설치하는 것은 네트워크 트래픽을 줄이기 위한 방법이지만 클라이언트가 전자상거래를 하기 위해 대용량의 Wallet 어플리케이션을 설치해야 한다는 부담을 준다.

그림 9는 ORB 에서 클라이언트와 서버 사이에서 IIOP(Internet Inter-ORB Protocol) 메시지 레벨에서 전송 메시지를 모니터링할 수 있는 인터셉터(Interceptor)를 이용하여 지역 데이터를 관리할 수 있는 기법이다.

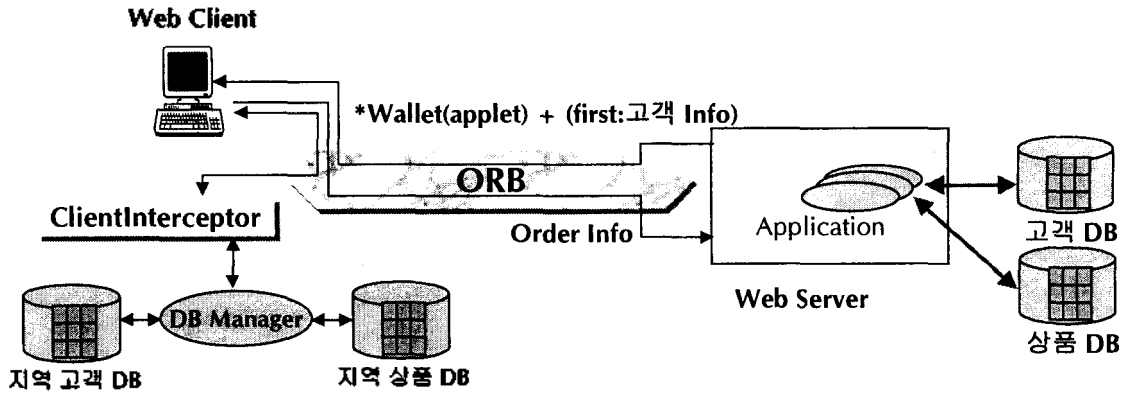


그림 9. ORB Interceptor를 이용한 지역 데이터 관리

그림 8에서 처럼 지역적으로 대용량의 Wallet 어플리케이션을 설치하지 않으며, 가벼운 애플릿을 서버로부터 전송하여 거래를 할 수 있다. 그리고 데이터베이스를 관리할 수 있는 관리 어플리케이션을 애플릿이 전송될 때 함께 클라이언트로 전송된다. 애플릿에서 지역적으로 고객이나 상품 데이터베이스를 접근하기 위해 인터셉터를 이용해 Security 제약 사항 없이 접근할 수 있다. 최신의 상품 정보가 생성될 때 마다 상품별로 애플릿이 전송될 때 지역 상품 데이터베이스를 수정한다.

III. 결론 및 향후 연구과제

본 논문에서는 CORBA 환경에서 전자상거래를 구축하기 위한 기법으로 다양한 지불 방식을 제공할 수 있는 지불 프레임워크 구축 기법과 인터넷 상에서 서버뿐만 아니라 웹 클라이언트의 트랜잭션도 관리할 수 있도록 CORBA의 OTS (JTS)를 이용한 트랜잭션 서비스를 이용하여 전자상거래 트랜잭션을 처리한다. 그리고 네트워크 트래픽을 감소시켜 전자상거래 시스템을 향상시킬 수 있는 분산 환경에서 데이터 분할 방법을 제시하였다.

향후 연구 과제로 전자상거래 시스템을 구축할 수 있는 DB 관리 모듈, 검색/주문 모듈, 지불 모듈, 그리고 광고계재 모듈을 개발하여 통합된 전자상거래 구축 모듈을 개발한다.

[참 고 문 헌]

- [1] R. Orfali, D. Harkey, J. Edward, *Client/Server Programming with JAVA and CORBA*, Jon Wiley&Sons: New York, NY, 1997.
- [2] Roger S. Pressman, *Software Engineering - A Practitioner's Approach*, Fourth Edition, McGraw-Hill, 1997.
- [3] J. Rumbaugh et. al., *Object-Oriented Modeling and Design*, Prentice Hall, 1991.
- [4] Grady Booch, *Object-Oriented Analysis and Design with Applications*, Benjamin/Cummings, 1994.
- [5] Martin Fowler, *UML Distilled Applying the Standard Object Modeling Language*, Addison-Wesley,

1997.

[6] Fayad, M. and Schmidt, D., "Object-Oriented Application Frameworks", CACM, Oct. 1997.

[7] Visigenic, "VisiBroker for Java", Reference Manual Version 3.0, Visigenic Software, Inc. 1997.

[8] Object Management Group. The Common Object Request Broker: Architecture and Specification. Revision 2.2 .

저자 소개

김 수 동

1984, 전산학 학사, Northeast Missouri State University

1988, 전산학 석사, The University of Iowa

1991, 전산학 박사, The University of Iowa

1991 - 1993 한국통신 연구개발단 연구실장/선임연구원

1993 - 1994 The University of Iowa, 교환교수

1994 - 1995 현대전자 S/W 연구소 책임연구원

1995.9 - 현재 송실대학교 컴퓨터학부 조교수

주요관심분야

객체지향 개발방법론 (UML, Java)

웹 기반 분산 객체 컴퓨팅 (CORBA, Intranet, Client-Server Web, Web Agents)

인터넷 상거래 시스템 기술

김 철 진

1996. 전산학 학사, 경기대학교

1998. 전산학 석사, 송실대학교

현재. 오브젝소프트(주) 연구개발실장

송실대학교 컴퓨터학부 대학원 박사 과정

주요관심분야

객체지향 사용자 인터페이스

객체지향 개발방법론 (OMT, UML)

객체지향 재사용 기법 (Framework)

웹 기반 분산 객체 컴퓨팅 (CORBA, Client-Server Web)