

CORBA 기반의 컴포넌트 기술과 전자상거래 응용

순천향대학교 백인천*

1. 서 론

지난 수십년간 소프트웨어의 복잡성이 기하급수적으로 증가하면서 위기론이 대두되었고 그에 대한 대안으로 객체 지향개념이 제안되었다. 하지만 이러한 객체지향 소프트웨어도 방대한 양의 일을 적절히 대처하며 급격히 변화해가는 소프트웨어 개발 유지 보수를 감당하기에는 역부족이었다. 실제 개발자들에 의해 객체지향에서 제시하는 재사용성, 대규모 시스템 개발 능력, 확실한 정보 은폐의 능력이 부족함이 대두되면서 컴포넌트 방식의 소프트웨어 설계 기법이 제안되게 되었다. 이러한 컴포넌트 방식의 시스템 설계는 기존의 웹이나 다가오게 될 객체웹의 구축에 쓰이고 있다.

더욱이 최근 10여년간 기하급수적으로 성장한 인터넷 환경은 가상공간(Cyber Space)을 형성하여 전 세계를 시간과 공간을 초월한 단일 세계로 묶고 있고 이 세계에서 무한한 변화와 새로운 가능성을 창출해 나가고 있다. 이러한 세계에서 새롭게 전개되는 전자상거래는 기업의 경영자와 소비자가 누리게 될 초현대적인 편리함과 효율성을 제공해 주게 될 것으로 내다보고 있다. 이러한 흐름에 발맞추어 수많은 관련 종사자들은 이 분야에 막대한 투자와 연구 개발을 진행하고 있고 소프트웨어 컴포넌트 분야는 이곳에 주요한 관심을 갖고 응용되고 있다. 이러한 컴포넌트는 전세계에 퍼져 있는 인터넷과 함께 소프트웨어 세계를 크게 달리할 것이다.

따라서, 본 고에서는 Common Object Request Broker Architecture(CORBA)기반의 분산 컴포넌트 환경을 설명하고 기업에서 사용하는 CORBA기반의 컴포넌트 개발환경과 이의 전자상거래 응용에 대해 설명한다.

2. 자바빈즈 컴포넌트와 CORBA와의 결합

2.1 컴포넌트의 편제

객체지향 프로그래밍을 사용하여 프로그램을 개발해도 서로 다른 기계와 운영체제에서 최종 기계 코드를 연결하는 것은 어렵다. 그래서 서로 다른 시스템간 또한 다른 언어간 객체를 사용하는 것은 곤란하며 또한 상속받을 때도 슈퍼클래스까지 접근해야 한다. 그리고 프로그램 내용이 변화되면 다시 컴파일하고 링크해야 한다. 이러한 문제는 컴포넌트라는 방식을 나오게 했던 배경이 된다.

컴포넌트는 자신의 어플리케이션 환경을 갖고 하나의 독립된 개체로서 활동한다. 이 컴포넌트는 인터페이스를 외부 사용자에게 제공하여 사용자는 이 인터페이스를 갖고 그 컴포넌트를 사용할 수 있다.

컴포넌트는, 전자회로 시스템을 제작할 때, 초기에는 트랜지스터나 다이오드를 일일이 꼽아 제작했고 반도체 회로를 설계할 때도 기본 게이트나 플립플롭부터 하나 하나 제작하여 표준컴포넌트를 새로운 방식이나 더 경제적인 방식으로 결합하는 방식으로 제작하듯이, 최종

*정회원

응용프로그램을 개발할 때 잘 제작된 컴포넌트를 소프트웨어 프레임워크에 맞추어 개발하는 형태로 진행되고 있다.

이러한 컴포넌트를 이용해 어플리케이션을 구성할 때 어플리케이션과 컴포넌트가 동일 주소공간에 있는 것이 기본이고 이때 가장 최적의 성능을 발휘할 수 있다. 그러나 현재의 네트워크 환경하에 클라이언트/서버 환경에서 운영되는 분산 컴포넌트들은 서로 다른 주소공간, 네트워크를 통한 이중의 기계에서 동작되므로 이것을 효율적으로 묶는 인프라가 필요하게 되는데 이러한 컴포넌트간 통신 인프라를 제공하는 모델로 OMG의 CORBA, 마이크로소프트의 DCOM, 썬의 RMI 등이 있다. 본 고에서는 Object Manage Group(OMG)의 CORBA 기반의 컴포넌트 체제를 설명하기로 한다. 특히 OMG는 썬의 자바빈즈를 CORBA 컴포넌트의 기본 사양으로 다음 스펙에서 결정하였다.

2.2 자바빈즈

기존의 소프트웨어 컴포넌트 개념으로 Visual Basic 컴포넌트와 객체지향 파스칼을 이용한 델파이 등을 들 수 있다. 이러한 것은 윈도우 환경하의 것인 반면, 최근 자바언어를 이용하여 기존의 컴포넌트를 발전시켜 개발한 것이 자바빈즈(JavaBeans)이다[5]. 자바빈즈는 자바언어의 플랫폼 독립성을 활용하여 이기종 및 시스템하에서 동작되는 보편적인 컴포넌트로서 성장을 기대하고 있다.

자바빈즈는 자바언어를 사용하여 자바컴포넌트를 구축할 수 있는 시스템 툴이다. 주요 특징으로는 내부 관찰자(Introspection)를 통해서 빈의 속성(Properties), 메소드(Methods), 이벤트(Events)를 찾을 수 있다. 디자인 패턴이나 BeanInfo 클래스를 사용하여 컴포넌트 내부를 관찰할 수 있다. 그리고 속성(properties) 빈은 설계시 노출되어 있어 주문화(Customization)가 가능하다. 또한 다른 빈과 통신하기 위해 이벤트를 사용한다. 이벤트를 받기 원하는 청취자 빈은 이벤트 소스 빈에 등록할 수 있다. 지속성을 위해 객체 직렬화(Object Serialization)를 사용해 빈으로 하여

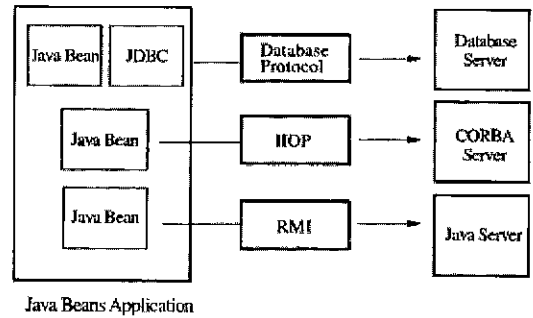


그림 1 자바빈즈 어플리케이션과 서버

금 상태 저장후 이후에 복구할 수 있도록 해준다. 자바빈즈 자체로는 단일한 가상머신 상에서 구현되고 동작되게 되어있다. 이 빈들은 여러 통신 프로토콜을 사용해 서버와 교신하게 된다. 그림 1에 자바빈즈 어플리케이션과 서버의 모습을 보였다.

2.3 CORBA 빈

자바빈즈 자체로는 단일 머현상의 컴포넌트를 주로 고려하고 있으나 실제 컴포넌트는 복잡한 클라이언트/서버 시스템을 만드는 것이 기대되고 이러한 분산 컴포넌트들을 구축하는 것은 객체 웹의 핵심요소이다. 따라서 CORBA 컴포넌트나 Enterprise JavaBeans(EJB)의 구상이 자연스럽게 대두되었다. CORBA 빈은 당초 설계시 단순히 IIOP를 통해 클라이언트 빈에서 호출할 수 있는 CORBA 객체로서 존재했었다. 이러한 CORBA 컴포넌트는 자바빈즈와 함께 결합되었을 때의 상승효과로 현재 OMG에서는 CORBA 컴포넌트의 기본 모델로 자바빈즈를 채택하고 있다.

실제 CORBA는 자바빈즈 컴포넌트에 빈을 위한 분산 서비스 하부구조를 제공하는 것과 자바빈즈는 CORBA에 메타데이터, 이벤트, 패키지화에 관련된 다양한 도구를 제공하는 효과를 제시하며 97년 CORBA 진영의 몇몇 업체들은 OMG에 CORBA 모델을 자바빈즈와 호환이 되게 정의할 것을 제안했다. 실제 CORBA 프로그래밍은 소수의 전문가들에 한정되어 있으므로 일반 개발자들이 비주얼 개발 도구나 스크립트 언어를 이용하여 비즈니스 어플리케이션을 쉽게 제작할 수 있는 환경을 구축

하기 원했고 이를 위해서는 표준 컴포넌트 기반 구조를 정의해야만 했다. 그래서 URL, LDAP(Lightweight Directory Access Protocol)등을 이용하여 객체에 명명할 수 있는 구조, 보안 지원 구조, 기존의 자바나 자바 스크립트와 같은 언어를 통한 객체접근 및 제어 등의 필요성을 제안하고 있다[14].

자바빈즈 컴포넌트를 CORBA 컴포넌트의 기본 모델로 채택하여 지원하고 있다. 이 모델의 요소로 자바빈즈 설계패턴, 이벤트 속성, 패키지, 메타데이터, 도구 등이 있다. 또한 현재의 주요한 개발환경인 비주얼 툴의 패러다임을 그대로 적용하고 있다. 컴포넌트의 설계 및 수행시와 똑 같은 모습으로 캔버스 위에 배치할 수 있게 해준다. 이러한 도구를 가지고 프로퍼티 편집기를 호출할 수 있고 컴포넌트간 상호 연결 패러다임을 제공한다.

CORBA 컨테이너에서 컴포넌트들이 계층적으로 구축될 수 있고 컨테이너는 하나 이상의 컨테이너를 구축할 수 있다. 또한 Component ARchive(CAR)로 CORBA 컴포넌트를 패키지화 한다.

3. RMI/IIOP 기반의 Enterprise JavaBeans 컴포넌트

최근 분산된 컴포넌트에 대한 트랜잭션 서비스에 대한 개발이 큰 이슈가 되고 있다. 대표적인 트랜잭션 서버구조로 마이크로소프트의 Microsoft Transaction Server(MTS), 썬 마이크로시스템의 EJB, OMG의 Object Transaction Service(OTS)가 있다. 이들의 트랜잭션 처리 서비스는 2단계(2 phase) commit 프로토콜을 이용한다. EJB는 자바기반 트랜잭션 처리를 위한 규정으로 썬을 주축으로 IBM, Oracle, Sybase, Netscape, BEA, Novell 등이 제안하여 98년에 사양이 발표되었다.

3.1 엔터프라이즈 자바빈즈(EJB)

EJB란 서버측 자바빈즈와 객체 트랜잭션 모니터(OTM) 혹은 서버측 컨테이너라 부르는 컴포넌트 조정자 사이의 계약을 말한다[7]. 서버측 컨테이너가 그 내부에서 실행하는 EJB

에 제공해야 하는 주요 기능들로는, CORBA RMI/IDL 의미론을 인식하고 CORBA OTS 트랜잭션을 전파할 수 있는 IIOP 기반의 분산 객체 하부구조를 제공하고 세션빈과 엔티티 빈의 작동환경제공, 컴포넌트 패키지화와 배포 기능, CORBA OTS/JTS 서비스에 기초한 2 단계 commit을 사용한 명시적 트랜잭션을 지원하는 선언적 트랜잭션 관리 기능, 엔터프라이즈 빈 전체의 생명주기를 관리하는 팩토리 지원기능, 컨테이너가 빈이 메모리로 로드되거나 비활성화 될 때 빈 호출을 관리하는 EJB플로팅 기능, 세션빈과 지속성을 이용한 엔티티 빈의 관리기능, 컨테이너 메타데이터 제공, 엔터프라이즈 빈에 대한 보안기능, 컨테이너간의 호환성 및 엔터프라이즈 빈에 대한 주문화 기능등이 있다.

엔터프라이즈 빈에는 세션빈과 엔티티 빈이 있다. 세션빈은 첫 사양에서 지원하도록 되어 있던 것으로 일반 비즈니스 로직을 가지며 일반 컴포넌트의 개념과 같은 것이다. 이것은 지속적 저장 장치에 연결되는 것이 아니므로 컨테이너가 재시작되면 이전의 빈은 사라진다. 컨테이너는 JNDI 네이밍 서비스 인터페이스를 통해 위치가 결정된다. 전개 기술어로 세션빈 메소드의 트랜잭션 작업을 결정해준다. 각 메소드는 새 트랜잭션이나 기존 트랜잭션의 일부로 실행되거나 트랜잭션 외부에서 실행될 수 있다. 따라서 시작과 실행을 비롯한 트랜잭션의 범위는 메소드 호출에서 명시적으로 구성될 수 있다. JDBC나 JSQL을 사용해서 데이터베이스를 접근할 수 있으며 각 사용자마다 다른 세션 빈 인스턴스가 존재한다[2].

엔티티빈은 보통 데이터베이스의 데이터를 나타내며 기본적으로 하나의 엔티티 빈은 테이블 하나의 레코드와 연결된다. 이것은 일단 만들어지면 이를 강제적으로 삭제하지 않는한 지속된다. 그러므로 컨테이너가 재시작되더라도 내용은 보존된다. JNDI를 사용하여 빈을 찾을 수 있고 세션빈처럼 전개 기술어를 사용하여 메소드의 트랜잭션 범위를 설정할 수 있다. 데이터 관리는 자체적으로 관리할 수도 있고 컨테이너에서 관리할 수도 있다. 하나의 빈 인스턴스에 대해 복수 사용자의 동시접근을 허용하고 동

기화 문제등을 컨테이너가 직접 해결한다.

EJB는 크게, EJB 서버, 컨테이너, 그리고 EJB 컴포넌트의 세 부분으로 나눌 수 있다. EJB 서버는 많은 컨테이너를 포함할 수 있으며 컨테이너안에는 EJB 컴포넌트가 존재한다. 하나의 컨테이너는 여러 개의 컴포넌트를 가질 수 있다. 서버와 컨테이너 사이에는 계약이 있다. 또한 컨테이너와 컴포넌트 사이에도 계약이 있다. 컨테이너는 각각의 EJB Bean에 인터페이스를 제공한다. 각각의 빈은 그의 컨테이너에 의해 EJB Object와 EJB Home의 두 개의 인터페이스를 개발자에게 제공한다. EJB Object는 원격 객체 호출을 위한 표준 자바 API 를 통해 클라이언트로부터 접근이 가능한 자바언어 객체가 된다. EJB Home은 클라이언트로 하여금 새로운 EJB 객체를 생성, 제거하거나 Enterprise Bean을 위한 EJBMetaData 인터페이스를 얻도록 한다. 이 인터페이스는 어플리케이션 제작 툴로 하여금 엔터프라이즈 빈을 위한 정보를 발견할 수 있게 한다. 그림 2 에 개발자의 관점에서 본 EJB 의 동작 방식을 보였다[2].

EJB 구조는 어플리케이션의 개발이나 배치에 있어서 6개의 구별되는 역할을 정의하고 있다. 엔터프라이즈 빈 제공자, 어플리케이션 구축, 배치, EJB 서버 제공자, 컨테이너 제공자 그리고 시스템 운영자등이 있다. 각각의 역할은 서로 다른 곳에서 진행될 수 있다.

3.2 표준 CORBA 매핑

여러 벤더들사이의 EJB 환경에 대한 상호운용을 위해 EJB 클라이언트 관점에서 CORBA에 대한 매핑 규정을 정하고 있다. CORBA매

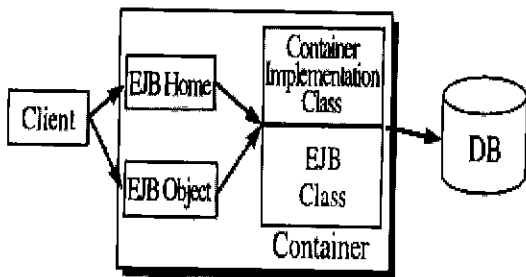


그림 2 EJB 동작 구성도

핑은 CORBA 클라이언트 인터페이스의 CORBA 매핑, 트랜잭션 컨텍스트의 전파, 그리고 보안컨텍스트의 전파를 다루고 있다[3].

EJB의 CORBA 매핑은 서버의 여러 벤더의 구현 사이의 연결상의 상호운용뿐 아니라 자바 클라이언트가 아닌 것들이 표준 CORBA API 를 통한 엔터프라이즈 빈으로 기술되는 서버측 어플리케이션에 접근할 수 있게 해주기 위해서이다. EJB 의 CORBA 매핑은 트랜잭션과 보안 컨텍스트의 전파를 위한 표준 CORBA 객체 서비스 프로토콜에 의존한다.

특히 현재 진행되고 있는 CORBA 스펙 3.0에서는 EJB 를 대폭 수용하고 있다. CORBA 3.0은 기존 CORBA 2.0에 자바 기술이 통합되고 메시징 기능의 추가, CORBA 3.0 어플리케이션을 개발할 수 있는 비주얼 툴의 지원, 비동기방식의 실시간 메시지 큐잉전송기능 및 기존 전용 시스템과의 통합을 위한 스펙이 추가된다. 또한 COM과의 호환성을 부여하며 Portable Object Adapter(POA)를 첨가하여 플러그 앤 플레이 기능이 가능하도록 추가하고 있다.

4. CORBA 기반의 기업용 웹 어플리케이션 서버 환경

웹 어플리케이션 서버는 기존의 웹서버 시스템이 갖고 있는 CGI에 의존한 시스템 부하나 HTTP가 갖고 있는 문제에 대해 해결책을 제시하며, 웹기반의 전자상거래 시스템의 확산과 분산객체 기술의 성숙 그리고 자바기술의 급성장에 부응하여 많은 서버 업체들의 지원과 함께 기업용 어플리케이션 서버 환경 구축에 사용되고 있다.

웹 어플리케이션 서버는 기존의 웹 요구를 받아 처리하는 웹서버, 특정 어플리케이션 요청에 대해 효율적인 자원할당이나 트랜잭션 관리 등을 제공하는 어플리케이션 서버, 비즈니스 로직을 구현하고 있는 어플리케이션, 그리고 웹문서나 컴포넌트 등을 빨리 구축할 수 있게 하는 통합 개발툴로 구성되며[12] 3-계층 클라이언트-서버 구조의 통합 미들웨어 인프라이다. 이것은 웹브라우저와 기업내 기존의

시스템의 중간에 위치하여 웹브라우저의 데이터 호출, 비즈니스로직 실행등을 수행하고 멀티쓰레딩, 커넥션 풀링, CORBA와 EJB지원, 데이터 접근관리의 분산 어플리케이션 개발을 자동화할 수 있게 해준다. 대표적인 어플리케이션 서버는 표 1과 같다.

5. 전자상거래를 위한 Java 컴포넌트와 활용

컴포넌트는 인터넷에서 보안이나 제작의 용이성에서 널리 퍼져나가고 있다. 이제 윈도우 환경하에서는 비주얼 베이직이나 델파이를 사용한 ActiveX 컨트롤이 있고, 자바나 자바빈

즈를 사용한 컴포넌트 제작, 그리고 전술한 MTS 나 EJB 환경하 서버측 컴포넌트들을 들 수 있고 이들의 개발이 활성화 되고 있다. 이제는 이러한 환경하에서 웹 콘텐츠를 제작하고 있으며 서버측 컴포넌트 개발 툴의 등장으로 소수의 전문가만이 담당했던 CORBA나 DCOM 프로그램을, 많은 사람이 GUI 툴을 사용하여 쉽게 서버측 컴포넌트나 어플리케이션을 개발할 수 있게 되고 이를 이용한 전자상거래 컴포넌트나 시스템 개발의 확산이 기대되고 있다. 본 장에서는 클라이언트에서 전자상거래 어플리케이션을 용이하게 개발할 수 있게 하는 쉐사의 Java Electronic Commerce Framework(JECF)와 이를 이용한 사례를 소개한다.

표 1 주요 웹 어플리케이션 서버

서버종류	특징
IBM WebSphere Application Server	<ul style="list-style-type: none"> - Apache HTTP서버를 기초로 SSL 방식의 IBM HTTP서버 제공 - 로터스 도미노 5.0 지원, Tivoli-ready 모듈과 XML 문서구조 서비스 및 Visual Age 와의 통합능력 - 자바빈과 CORBA 지원, EJB와 관계DB 운영과 모니터링을 제공하는 EJB 서버 - 트랜잭션처리 환경을 위한 TXSeries 지원 - 분산객체와 비즈니스 절차통합능력을 갖고 CORBA,ActiveX를 지원하는 컴포넌트 브로커
Inprise Application Server	<ul style="list-style-type: none"> - VisiBroker/ITS CORBA 환경에 기반을 둔 분산 수행 시간 서비스 - EJB와 CORBA 컴포넌트를위한 지속 엔터티 객체와 컨테이너관리의 부가기능 - VisiBroker/ITS 서비스에 기반한 멀티 쓰레딩, 쓰레드 풀, ORB 연결-공유, DB 연결 풀링 지원 - JBuilder,비주얼 자바 어플리케이션 개발툴 지원 - 클러스터링, 부하균등화, 회복, 스마트 에이전트 기술 지원
Oracle Application Server	<ul style="list-style-type: none"> - 카트리지라는 어플리케이션 구축용 Web Request Broker 구조 - PL/SQL, OCI,JDBC,ODBC,SQLJ,XA를 통한 DB 접근 - Java-CORBA 객체라는 카트리지 지원 - OTS에 기반한 Java Transaction Service API 제공 - CORBA/OTS, X/Open TX 인터페이스 지원 - 기본인증을 포함한 6개의 보안메커니즘 제공
SUN NetDynamic Server	<ul style="list-style-type: none"> - 주요DB연결, CORBA, EJB, COM등의 다중 객체 컴포넌트기술 지원 - MQSeries, CICS, LDAP와 같은 리저시 시스템을 위한 코바 기반의 Platform Adapter Component(PAC) 지원을 통한 비즈니스 Portal 어플리케이션 구축 - CORBA,EJB와 COM,MTS,SQL 서버지원 - SMTP,IMAP,POP,SNMP,JDBC 등의 산업 표준 지원
Sybase Enterprise Application Server	<ul style="list-style-type: none"> - CORBA IDL을 사용한 컴포넌트 인터페이스 정의 - PowerBuilder 객체나 ActiveX/COM 컴포넌트, 자바클래스, EJB 등에서 양방향 리엔지니어링 - Jaguar Component Transaction Server(CTS) 제공

5.1 The Java Electronic Commerce Framework (JECF)

JECF는 썬 마이크로시스템에서 인터넷이나 연계된 인트라넷 내에서 비즈니스를 하기 위한 안전하고 확장 가능한 프레임워크이다[6]. JECF의 클라이언트측 어플리케이션으로서 자바환경의 첫 컴포넌트로 개발된 것이 자바 월렛(Java Wallet)이다. 이것은 여러 개의 서브 시스템을 하나로 묶은 것으로서 크게 Java Commerce Client(JCC), Commerce JavaBeans, Java Smart Card API의 세 부분으로 나누어 지고, 더 자세히는 다음의 7개의 서브 시스템으로 나눌 수 있다.

- Java Commerce Client(JCC) : JECF의 핵심인 클라이언트측 솔루션으로 전자상거래 어플리케이션을 수행하기에 용이하도록 플랫폼을 제공한다. JCC는 사용자 인터페이스를 제공하면서 하나의 데이터베이스를 포함하고 있다. JCC가 장착되었던 여기에 제3자에 의해 만들어진 새로운 인터넷 프로토콜들과 Commerce JavaBeans 컴포넌트들을 다운로드해서 장착 할 수 있다. 또한 트랜잭션 로그와 같은 여러 내부서비스, Merchant를 위한 사용자 인터페이스, Java Commerce API를 통한 기본 서비스 등을 제공한다.
- Database : 사용자 정보, 카셋트, 트랜잭션 로그 등을 저장하는 관계 데이터베이스이다.
- 연산(Operation), 프로토콜(Protocol), 수단(Instruments) : 연산은 프로토콜(SET과 같은 지불프로토콜 등)과 수단(Smart 카드, 신용카드 등)을 서로 연결시켜 줌으로서 하나의 태스크를 수행하도록 하는 전체의 과정을 말한다.
- 여러 개의 Cassettes/Commerce JavaBeans : Cassette란 전자적으로 서명된 JAR file들로서 하나나 그 이상의 Commerce JavaBeans 컴포넌트들과 Bean들이 사용하는 여러 자원들을 가지고 있다. Commerce JavaBeans는 특정 인터페이스 요구사항을 갖는 재사용이 가능한 컴포넌트로서 개발자는 이를 이용해서 JCC의 기능을 확장시키거나 상거래 컴포넌트들을 개발할 수 있다.

Commerce JavaBeans는 JavaBeans를 확장함으로 Commerce JavaBeans 인터페이스 타이핑(Commerce JavaBeans의 인터페이스를 몇 가지 유형으로 유형화함)과 게이트웨이 보안 모델(Gateway Security Model)을 가능하도록 했다. Commerce JavaBeans는 현재 7개의 유형화된 인터페이스를 갖고 있고 확장이 가능하기 때문에 개발자는 새로운 타입의 Commerce JavaBeans 컴포넌트를 만들 수도 있다.

- 사용자 인터페이스 : 사용자 인터페이스는 자신이 원하는 모양으로 변화가 가능하며 도구를 수정하거나 트랜잭션 재열람 등을 할 수 있다.
- 게이트웨이 보안 모델 : Gateway Security Model은 JCC의 보안체제로서 Bean들과 JCC사이의 통신을 일정한 신용모델(규약)에 따라 제한한다
- Java Commerce 메시지 : Java Commerce 메시지는 서버들이 JCC와 통신하기 위한 메시지 형태들의 집합이다.

5.2 JCC와 EJB를 이용한 전자상거래용 서버측 검색용 컴포넌트 설계 사례

CORBA를 이용한 전자 상거래 이용 실패로 W3C와 CommerceNet에서 제안한 Joint Electronic Payment Initiative(JEPI) 구조에서는 다중 지불 프로토콜을 수용하는 기본 소프트웨어 통신 버스로 CORBA ORB를 사용하도록 제안되어 있다. IBM, SUN 등 주요 인터넷 업체를 주축으로한 CommerceNet에서는 eCo 프레임워크를 제안하며 플랫폼간 상호호환, 지불과 보안서비스의 호환을 위해 CORBA IIOP를 지원하기로 하는등 기타 다른 네트워크 및 운영체제, 데이터 베이스간 투명성과 상호운용성, 독립성을 보장하기 위한 방안으로 CORBA를 이용한 모델이 제시되고 있고 CORBA기반의 객체 웹상에서 다양한 전자상거래 컴포넌트와 웹 어플리케이션이 개발 중이다. 또한 현재 연구되고 있는, 클라이언트측에서 JCC를 사용하고 서버측에서 EJB를 이용하며 RMI/IIOP 통신을 사용하는 전자상거래용 클라이언트와 서버측 검색용 컴포넌트 시스템

의 구조를 그림 3에 보였다[13]. JCC에는 상품 검색기능이 추가되고 서버측 컨테이너 안에는 상품검색이나 정보 재구축을 위한 컴포넌트가 구축된다. 이 시스템은 사용자 자유롭게 훈련이 가능한 검색엔진을 기초로 구축된다.

6. 결 론

소프트웨어의 발전은 현재 컴포넌트라는 개념에 이르게 되었고 언어에 관계없이 제작되고 연결되던 상황에 자바언어 환경하라는 개념에도 불구하고 자바언어의 단일환경을 수용하여 자신의 RMI와 OMG의 CORBA/IIOP를 기반으로 한 엔터프라이즈 자바빈즈(EJB)의 출현에 이르게 되었다. 업계와 연구자들은 마이크로 소프트웨어의 MTS와의 서버측 컴포넌트 분야의 경쟁에 깊은 관심을 갖고 있다. 본고에서는 CORBA와 자바기반의 컴포넌트에 대해 살펴보았고 서버측 분산 컴포넌트인 EJB에 대해 기술하였다. 또한 자바 컴포넌트 기반의 전자상거래 프레임워크와 서버측 컴포넌트를 이용한 전자상거래 응용 시스템 사례에 대해 제시하였다.

CORBA는 이제 컴포넌트간 통신 인프라의 표준으로 사용되게 되었고 이 컴포넌트들은 어플리케이션 서버 개발환경에서 사용자가 사용하기에 편한 GUI를 갖는 틀로 쉽게 제작되게 되었다. 따라서 이러한 틀은 많은 개발자를 끌어 들이게 될 것이고 이러한 컴포넌트 기반의 많은 클라이언트/서버용 어플리케이션이 제작되리라 예측된다. 실제 많은 전자상거래용 컴포넌트와 어플리케이션이 이를 이용해 제작되고 있다.

전자 상거래는 시간과 공간을 초월한다. 따라서 그 대상은 전 세계의 수십억의 인구이고 상업행위의 시간은 기존의 방식에 비해 거의 순간적이라 할 수 있다. 이러한 분산 컴포넌트 또한 상품이 될 수 있고 대상은 네트워크를 통한 전 세계이고 판매시간은 극히 짧다. 또한 컴포넌트를 사용하므로 전자상거래를 구축하기 위한 가상공간의 양상이 변하고 브라우저이나 쇼핑을 위해 사용하는 틀의 개발도, 객체웹과 같은 다른 환경과 사용자가 직접 개발에 참여

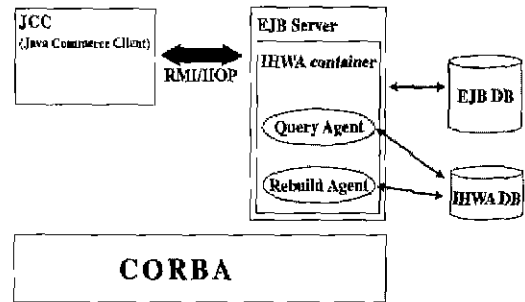


그림 3 JCC와 EJB를 이용한 상품검색 컴포넌트 시스템

하므로 다양한 형태의 모습을 가질 것이고 단축된 시간을 요할 것이다. 이러한 변화에 능동적으로 대처해 나간다면 전자상거래를 위한 기초기술에서부터 문화형성까지 전반적으로 크게 활성화할 수 있는 초석이 될 것이다.

참고문헌

- [1] Sun Microsystems, *Commerce JavaBeans Tutorial*, 1998.
- [2] Vlada Matena & Mark Hapner, *Enterprise JavaBeans*, Sun Microsystems, 1998.
- [3] Rohit Garg, *Enterprise JavaBeans to CORBA Mapping V1.0*, Sun Microsystems, 1998.
- [4] Daniel J. Guinan, *Commerce JavaBeans*, Sun Microsystems, March 1998.
- [5] Graham Hamilton, *JavaBeans version 1.01*, Sun Microsystems, 1998.
- [6] JECF documentation page, <http://java.sun.com/products/commerce/index.html>.
- [7] Robert Orfali, Dan Harkey *Client and Server programming with Java and Corba*, John Wiley & Son, 1998.
- [8] Incheon Paik, T. Han, "Universal Electronic Commerce Framework and Distributed Object Services Based on SET Protocol", *Proceedings of IA-STED Conference*, Oct. 1998.

[9] Sun Microsystems, The Java Wallet Architecture White Paper, March 1998.

[10] IBM WebSphere Application Server, <http://www.software.ibm.com/webse-rvers/>.

[11] Inprise Application Server, <http://www.inprise.com/appserver>.

[12] 한재선, 김동은, 김경백, 박대연, “웹어플리케이션 서버:인터넷 전자상거래를 위한 공용 플랫폼”, '99 EC/CALS 기술 워크샵 발표자료집, 1999. 5.

[13] Incheon Paik, T.W. Han, “Design and Implementation of Electronic Commerce Search Engine Components”, International Conference of

AoM/IAoM, San Diego, 1999.

[14] <ftp://ftp.omg.org/pub/docs/orbos/1997/97-05-25.doc>.

백 인 천



1985 고려대학교 전자공학(공학사)
 1987 고려대학교 대학원 전자공학(공학석사)
 1992 고려대학교 대학원 전자공학(공학박사)
 1993~현재 순천향대학교 컴퓨터학부 조교수

관심분야: 압호 및 멀티미디어 회로설계, 인터넷 검색엔진, 분산객체, 객체지향

소프트웨어 설계, 소프트웨어 컴포넌트, 전자상거래 시스템 및 보안

E-mail: paikic@asan.sch.ac.kr

● 제1차 리눅스 포럼(Linux Forum '99) ●

- 개최일자: 1999년 7월 27일(화)~28일(수)
- 개최장소: 서울 롯데호텔(소공동 소재)
- 주 최: 정보통신부
- 주 관: 한국정보과학회, 한국전산원, 한국전자통신연구원, 전자신문사
- 개최목적: - 리눅스 활성화를 위한 산학연 교류와 협력의 장
 - 리눅스 정보와 기술교류로 리눅스 개발방향 선도
 - 리눅스 현안 이슈들에 대한 이해증진과 공동대처
 - 리눅스 신제품 발표로 리눅스 솔루션 활용 극대화 유도
- 행사내용: - 기초연설: Jon "Maddog" Hall(Linux International)
 - 특별강연: T.L. Kunii 교수(일본 Hosei대학)
 - 컨퍼런스: Tutorials, Track(User, Developer, Business)
 - 전 시 회: 후원기관 중심으로 신제품 발표 및 전시
- 문 의 처: 한국정보과학회 사무국
 Tel. 02-588-9246/7, E-mail:kiss@kiss.or.kr