

# EJB 기반의 웹 커뮤니티 설계 및 구현

성 경\* · 염희균\*\* · 김석훈\*\*\*

\*목원대학교 · \*\*대전대학교 · \*\*\*한남대학교

## A Design and Implementation of Web Community based on EJB

Kyung Sung\* · Hee-Gyun Yeom\*\* · Seok-Hun Kim\*\*\*

\*Mokwon University · \*\*Daejeon University · \*\*\*Hannam University

E-mail : skyys04@mokwon.ac.kr

### 요 약

기존의 Model 1 웹 프로그램보다 유지, 보수 및 재사용성이 높은 분산객체시스템인 EJB 컴포넌트를 이용하여 Community 웹 애플리케이션 프로그램을 구현하였다. 기존에 있던 웹 프로그램들은 비즈니스 로직과 UI 로직이 같은 층에 존재하여 유지보수 및 재사용을 하기가 어려웠다. 그러나 컴포넌트를 사용하여 프로그램을 구현 함으로써 비즈니스 로직과 UI로직을 서로 다른 머신으로 분리시켜서 비즈니스 로직의 재사용성과 시스템 아키텍처의 유연성을 높였다. 그리고 EJB컴포넌트를 사용 함으로서 트랜잭션 및 패일오버(Failover)를 최소화하며 기존보다 안정된 프로그램으로 구현하였다.

### ABSTRACT

Embodied Community web application program preservation, damage control and resuability uses EJB component that is high breakup object system than existent model 1 web program. Web programs that is in existing hard to exist on course which UI logic is same with business logic and do maintenance and resuability. But, improved resuability of business separating business logic and UI logic by different machine embodying program to use component. Minimizes Teuranjeksyeon and Paeilobeo and embodied in stabler program than existing by using EJB component.

### 키워드

EJB, Software Engineering, Component, Web Community

### 1. 서 론

요즘 인터넷의 급속한 보급으로 많은 인터넷 사용자가 급속하게 늘고 있다. 웹 사이트의 엄청난 증가와 그에 비례한 콘텐츠의 증가로 인해 유지, 보수비용 또한 대폭 증대됐다. 그에 대한 대안으로 등장한 것이 컴포넌트를 이용한 분산 객체 시스템이다. 기존의 웹 프로그램들은 비즈니스 로직과 UI 로직이 같은 층에 존재하여 유지보수 및 재사용이 상당히 힘들었다. 그러나 컴포넌트를 사용하여 프로그램을 구현함으로써 비즈니스 로직과 UI 로직을 서로 다른 머신으로 분리시켜서 비즈니스 로직의 재사용성과 시스템 아키텍처의 유연성을 높였다. 컴포넌트 기반 소프트웨어 시스템 개발은 소프트웨어 업계에서 나타난 표준적인 컴포넌트 기반 구조 기술의 특성과 소프트웨어

생산성 모두를 주목할 만하게 향상시키기 위한 솔루션으로 EJB, CORBA, ActiveX, JavaBeans 등이 등장하였다[2]. 이런 컴포넌트시스템을 사용함으로써 트랜잭션 및 패일오버(Failover)을 최소화하며 기존보다 안정된 프로그램으로 구현 될 수 있었다[1].

인터넷에서는 자기와 서로 친분 있는 사람들과 의견을 나눌 수 있는 Community 사이트들이 많이 존재한다. 그렇지만 게시판이나 사진첩 기능들을 다른 웹 프로그램에서 재사용 하지 못하며, UI와 비즈니스 로직이 같이 있어서 유지, 보수에 상당히 힘들었다. 그래서 재사용성이 상당히 높으며 유지, 보수가 쉬운 CBD 기반의 Community 웹 애플리케이션 프로그램이 필요하다.

본 논문에서는 EJB기반의 Community 웹 애플리케이션 프로그램을 설계 및 구현 하고자 한다.

본 연구의 구성은 먼저 2장에서는 관련연구들을 살펴보고, 3장에서는 Community 컴포넌트의 설계, 4장에서는 구현 및 실험결과에 대해 다루고, 5장에서는 향후연구 및 결론을 맺는다.

## II. 본 론

### 2.1 CBD(Component Based Development)

CBD란 재사용 가능한 소프트웨어 모듈컴포넌트를 생성 및 조립 생산, 선택, 평가 및 통합으로 구성하여 더 큰 컴포넌트를 생성하거나 완성된 애플리케이션 소프트웨어를 구축하는 개발 기법이다. 소프트웨어 컴포넌트를 조립해 새로운 애플리케이션 만들 수가 있으므로, 개발기간을 단축할 수 있고, 기존의 컴포넌트를 재사용함으로써 생산성과 경제성을 높일 수 있다. 컴포넌트로 시스템을 만들 때의 장점은 개발하려는 시스템의 전체를 컴포넌트화 할 수 있는 여러개의 단위로 분리해 개발하므로 복잡성을 단순화시킬 수 있고, 캡슐화가 되어있어 로직 상의 에러나 런 타임 에러 등의 범위를 컴포넌트로 한정할 수 있어 유지보수가 용이하다[4][7].

### 2.2 EJB(Enterprise Java Beans)

선(Sun)사에 의해 제안된 EJB(Enterprise Java Beans)는 대규모 프레임워크를 위한 자바 기술로써 분산환경에서 애플리케이션을 개발, 배포 실행하기 위한 아키텍처이다[3]. 분산 환경하의 애플리케이션들은 트랜잭션 관리, 보안, 데이터베이스 제어들과 같은 시스템레벨의 서비스를 필요로 하며, J2EE(Java2 Enterprise Edition), Web Logic 등의 플랫폼은 이러한 서비스 등을 제공하여 사용자가 비즈니스 로직에만 전념할 수 있도록 다양한 기능을 제공하고 있으며 전 세계 여러 곳의 사이트에 적용되어 운영되고 있다[5].

EJB는 컴포넌트를 특성에 따라 일반적인 비즈니스 로직을 나타내는 세션 빈(Session Bean), 데이터베이스에 저장된 정보와 같은 여러 클라이언트에 의해 공유되며 연속장치에 저장되는 엔티티 빈(Entity Bean) 그리고 JMS 메시지를 처리하는 메시지 기반 빈(Message-driven-Bean)으로 구분한다. 빈 개발자는 EJB를 작성하기 위해 홈 인터페이스, 리모트 인터페이스, 빈 인스턴스를 작성해야 한다.

EJB 클라이언트는 JNDI 네이밍 서버를 통해 홈 인터페이스에 대한 레퍼런스를 얻고, 홈 인터페이스를 통해 리모트 인터페이스에 대한 레퍼런스를 얻을 수 있으며, 이를 이용해 빈 인스턴스에 메시지를 보내게 된다. 이 메시지 바로 빈 개발자가 작성한 빈 인스턴스에 전달되는 것이 아니라 J2EE나 Web Logic 같은 컨테이너가 중간에 가로채 보안 및 트랜잭션 등의 처리를 수행 후, 해당 빈에게 메시지를 전달한다.

## III. Community System 위한 컴포넌트 설계

### 3.1 시스템의 기본구조

현재의 중/대형 규모 소프트웨어 개발의 여러 특징 중의 하나는 아키텍처 기반이라는 점이다. 매우 다양한 네트워크 다양한 컴퓨터 자원들을 포함한 광범위한 분산 컴퓨팅 환경에서 필요한 모든 소프트웨어를 처음부터 개발하는 것은 어려운 작업이 된다. 기존의 라이브러리, 프레임워크, 컴포넌트를 포함한 다양한 리소스를 뿐만 아니라, 여러 프로젝트를 통해 경험/전략까지도 재사용하는 것이 필수적이다[6].

Community 시스템을 EJB 컴포넌트를 기반으로 설계 및 구현하여 컴포넌트의 장점들을 고스란히 이어받았다. 본 시스템의 구조는 (그림1)과 같다. 먼저 클라이언트 즉 WebBrowser로 jsp, html, servlet등으로 이루어진 webserver에 접속을 하여 값들을 입력하면 그 값 들은 비즈니스 로직이 있는 EJB 컨테이너 안으로 전달되고 컴포넌트 들은 로직에 맞게 처리하며 필요한 경우 DBserver에 접속하여 데이터의 삭제 및 수정 읽기가 이루어지게 된다. 그리고 처리한 데이터들은 다시 webserver로 보내서 화면에 출력한다.

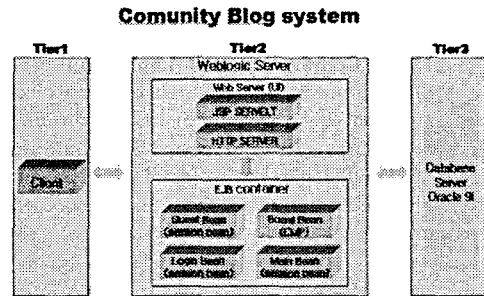


그림 1. Community System Architecture

본 시스템은 세션 빈 3개와 엔티티 빈 1개 즉 4개의 EJB컴포넌트와 UI를 표현해주는 Jsp, Html, Servlet들로 이루어진다. 4개의 컴포넌트의 구체적인 구성은 사용자 정보등록 및 인증을 담당하는 login bean, 게시판, 사진첩의 검색 및 주요기능을 담당하는 boardbean, 방명록의 기능을 담당하는 guestbean, 메인페이지의 이미지수정 및 그 밖의 기능을 담당하는 mainbean으로 구성된다.

### 3.2 주요 컴포넌트의 구성 및 설계

#### 3.2.1 Login 컴포넌트

Login은 ConnectHome, Connect의 두개의 인터페이스와 ConnectBean, ConnectDataBean빈 클래스로 이루어진다. 서로간의 연관관계는 (그림2)를 통하여 살펴본다.

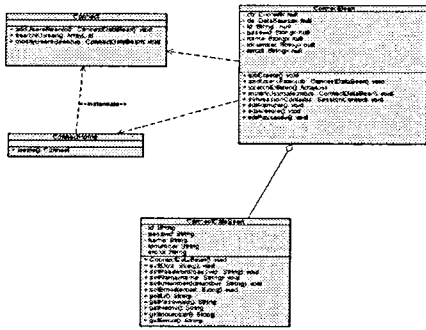


그림 2. Login Component Class Diagram

로그인의 리모트 인터페이스인 Connect 클래스는 회원가입 메소드인 addUsersBean(), ID 검색 메소드인 searchIDBean(), 회원정보 갱신 메소드인 modifyUsersBean() 등의 메소드들로 이루어진다. 로그인인 홈 인터페이스는 create() 메소드를 이용하여 빈을 생성하는 역할을 담당한다.

Connect Data Bean 클래스는 회원정보 한건의 정보를 기억할 수 있는 객체이다. 회원정보각각 데이터에 따른 set, get함수들로 이루어져있다. 해당객체는 원격인터페이스 빈 클래스에서 사용된다. ConnectBean 클래스는 리모트 인터페이스에서 정의한 메소드를 처리하는 비즈니스 로직들이 구현된다.

### 3.2.2 Board 컴포넌트

게시판 컴포넌트는 CMP방식의 엔터티빈으로서 BoardHome, Board 두개의 인터페이스와 Board Bean클래스로 이루어진다.

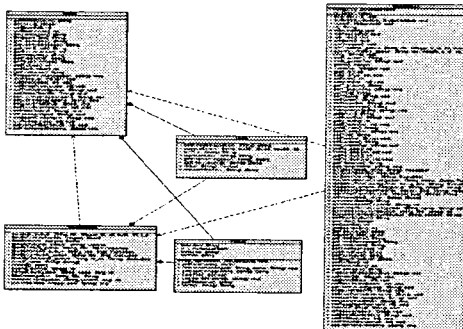


그림 3. Board Component Class Diagram

리모트 인터페이스인 Board클래스는 게시물의 정보 즉, 게시물번호, 글쓴이, contents, 이메일, 홈페이지, password등의 게시물하나의 각각 데이터들을 얻을 수 있는 get메소드들과 객체에 다른 데이터가 저장되어 있다면 원래의 데이터를 제거하고, 새로운 데이터를 저장하는 즉, 수정작업에 사용되는 set메소드들이 정의되어진다.

홈 인터페이스인 BoardHome클래스는Primary key인 게시물의 고유 id값을 통하여 원하는 객체를 찾는 메소드인 findByPrimaryKey(), 제목 및 글쓴이 내용 등을 통하여 원하는 객체를 찾는 메소드인findByTitle(), findByName(), findByContent()등 과, 이름, 본문, 내용으로 검색 했을 때 그에 맞는ID값을가져오는 nameCount(), titleCount(), contentCount()등으로 구성되어져 있다. 하나의 레코드를 가져오면 리턴타입이 Board로 정의하며, 여러 개의 레코드를 가져올 경우 Collection타입으로 정의하였다. 빈 클래스인 Board클래스는 리모트 인터페이스와 홈 인터페이스에서 정의한 메소드들을 실제로 구현하여 주었다. 그리고 ejbSelect는 일단 abstract타입으로 정의한 후 홈 메소드와 매칭 되는 메소드를 사용하여 사용자가 원하는 형식으로 정의해 주었다.

EJB SELECT 메소드들을 살펴보면, 해당게시판에서 가장 큰 ID를 찾아오는 ejbSelectMaxB\_id(), 전체게시판의 COUNT를 담당하는 ejbSelectCount(),이름, 제목, 내용으로 검색할 때 사용되는 ejbSelectNameCount(), ejbSelectTitleCount(), ejbSelectContentCount(), 게시물의 비밀번호를 체크하는 ejbSelectGetPassWord() 메소드 등으로 구성된다. 그리고 그밖에 파일의 업로드를 담당하는 fileup.java Utility.java 클래스가 있다.

### 3.3 UI 설계 및 구성

회원 등록 후 로그인을 한 다음 자기의 커뮤니티 컨텐츠들을 이용하고, 즐겨찾기를 통하여 다른 사용자의 컨텐츠로 이동 할 수 있다. 기본적인 UI(User Interface)흐름은 (그림4)와 같다.

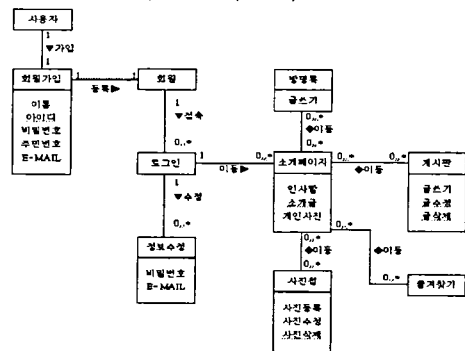


그림 4. Community system UI 흐름도

## IV. Community 시스템의 구현 및 실행

### 4.1 시스템 환경 및 구성

개발환경은 아래와 같다.

- OS : WINDOW XP
- web sever : web logic 8.1.3
- J2SE Ver : 1.4.2 J2EE Ver:1.3
- EJB Ver : 2.0

4.2 실행 화면

Community 시스템의 로그인후 자기만의 메인 페이지 실행화면은 (그림5)와 같다.

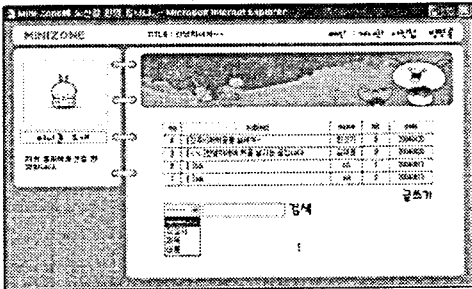


그림 5. 메인페이지 실행화면

손님들이 남기고 가는 인사말들이 나오는 방명록 부분의 실행화면은 (그림 6)과 같다.

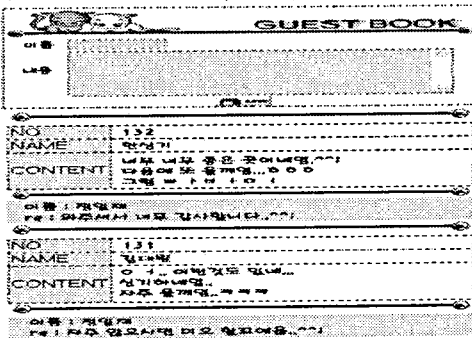


그림 6. 방명록 실행화면

게시판과 같은 컴포넌트를 사용하기 때문에 형식은 비슷하나 이미지를 화면에 출력해주는 사진첩의 실행화면은 (그림 7)과 같다.

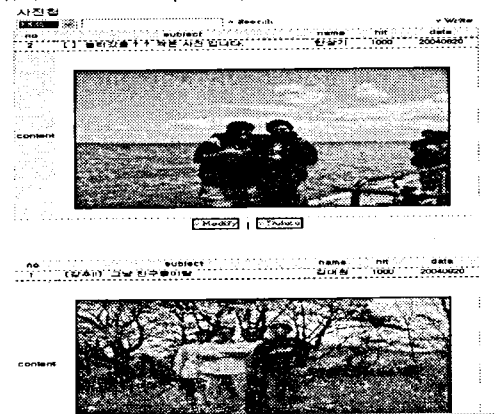


그림 6. 사진첩 실행화면

V. 결론 및 향후연구

EJB 컴포넌트 기반으로 Community 시스템을 만들어본 결과 개발자는 비즈니스 로직만을 신경 쓰고, 보안, 트랜잭션 처리 등은 EJB컨테이너에 Deploy시키면 자동으로 해주어서 더욱 개발에 효율적이었다. 유지 보수에 있어서는 model 1의 웹 프로그램과 비교해서 UI와 비즈니스 로직이 혼합되어 있지 않았기 때문에 쉽게 유지보수를 할 수가 있었다. 재사용성에 있어서는 아직은 소스의 아무런 수정 없이 컴포넌트를 분리하여 조합할 수는 없었고, JSP, JAVA언어에서는 소스와 환경 등을 조금씩 수정하여 재사용을 할 수가 있었다. 앞으로 향후 연구로는 JAVA기반이 아닌 다른 객체지향언어기반의 프로그램에도 이식할 수 있도록 재사용성이 뛰어난 컴포넌트를 구현할 수 있는 연구가 필요하다.

참고문헌

- [1] Gilda Pour, "Enterprise JavaBeans, JavaBeans &XML Expanding the Possibilities for Web-Based Enterprise Application Development", in Proceedings of Technology of Object-Oriented Languages and Systems, pp. 282-291, 1999.
- [2] Mikio Aoyama, "New Age of Software Development : New Component-Based Software Engineering Changes the Way of Software Development," 1998 International Workshop on Component-Based Software Engineering. ICSE, p.124~128, 1998.
- [3] Sandra L. Emerson, Michael Girdley, Rob. "J2EE Application and BEA WebLogic Server." Prentice Hall, 2001.
- [4] 최성, 윤태권 "CBD(Component Based Development)현황과 전망", 정보처리학회지. vol. 10. May. 2003.
- [5] 신동규, 신동일, 차석일, 장철수, 이경호, 김중배. "EJB기반 모바일 전자 상거래 컴포넌트의 설계 및 구현". 정보처리학회 논문지, vol9-D, August, 2002.
- [6] 박지훈 이용원, 김종윤. "EJB 엔터프라이즈 자바 빈즈". 대청출판사, 2001.
- [7] Peter Herzum, Oliver Sims, Business Component Factory : A Comprehensive Overview of Component-Based Development for the Enterprise, OMG press, December,