

서블릿의 EJB 변환 기법

신정은*, 최일우, 류성열
숭실대학교 컴퓨터학과

e-mail : {kikiwa*, lucifer}@selab.ssu.ac.kr, syrhw@computing.ssu.ac.kr

Translation Technique from Servlet to EJB Component

Jung-Eun Shin*, Il-Woo Choi, Sung-Yul Rhew
School of Computing, Soongsil University

요 약

인터넷 사용자의 빠른 증가는 인터넷 환경 발달을 요구한다. 웹의 아키텍처를 N-계층으로 변환하는 이유도 웹 서버의 과부하를 줄여 속도증가와 성능향상을 얻기 위함이다. 그러나 여전히 네트워크를 통한 업무처리 관련 분산 시스템은 계속 되는 개발 단계에 있다. 사용자들의 각기 다른 시스템 환경과 그에 적합한 다양한 어플리케이션의 요구, 늘어나는 새로운 프로그램들과 개발 시스템들은 계속되는 소프트웨어의 유지보수를 필요로 한다. 컴포넌트는 급속하게 변화되는 소프트웨어를 효율적으로 관리하고 활용하기 위해 새롭게 대두되고 있으며, 재사용 가치가 높은 기존 소프트웨어에 적용하여 관리하는 것도 필요하다. 본 논문에서는 인터넷에서 사용되고 있는 서블릿 게시판을 컴포넌트화 하여 웹상의 소프트웨어로서 재사용성을 높이며, 서블릿을 EJB 로 컴포넌트화 하는 변환 과정을 제안하고자 한다

1. 서 론

서블릿은 플랫폼에 독립적이고, 서버에서 작동되어 속도가 빠른 장점 때문에 웹 어플리케이션 개발 기술로 사용되고 있다[1]. 그러나, 인터넷 보급 확대에 따른 e-Business 소프트웨어 수요의 급속한 증가와 대규모 시스템 개발에는 좀 더 향상된 재사용성을 필요로 한다. 새로운 소프트웨어 개발 패러다임으로 제기된 컴포넌트는 사용자에게 인터페이스만을 제공하여 내부적인 어플리케이션 코드를 다시 수정 하지 않고 독립적으로 사용하고 할 수 있도록 하여 재사용성을 높인다.

특히, EJB 컴포넌트는 자바의 특성을 유지하여, 하드웨어나 DB, 컨테이너에 관계없이 독립적으로 운영되기 때문에 이식성이 높은 장점이 있다.

본 논문에서는 웹상에 40,000 개 이상이 존재하는 서블릿으로 개발된 게시판을 EJB 로 변환하는 과정을 제시한다. 이 논문에서 제시한 변환 방법을 통하여 기존의 소프트웨어를 최대한 재사용하고 컴포넌트를 개발하여, 사용자의 사양에 관계없이 게시판 성능을 유지하며, 서버 내에 많은 객체들의 인스턴스를 각각

효율적으로 관리할 수 있도록 한다.

2. 관련 연구

2.1. 분산 객체 아키텍처

분산 객체 시스템은 객체 지향 기술을 근본으로 한다. 분산 객체는 주로 3-계층 아키텍처에 기반을 둔다. 첫번째는 사용자의 인터페이스와 관련된 프레젠테이션, 두번째는 비즈니스 로직을 포함한 웹서버, 세번째는 데이터 베이스와 같은 자원을 제공하는 데이터로 나누어진다.

분산객체는 서로 같은 기본구조의 프로토콜을 가지고 네트워크상 분산된 객체간의 통신을 지원하며, 객체 서버, 스킴렉톤, 스텝 이라는 세가지 요소로 구성된다.

그림 1 에서 스텝은 클라이언트내에 존재하며, 객체 내에 존재하는 비즈니스 메소드들의 인터페이스를 구현한다. 클라이언트의 요청은 스텝을 통해 Remote 서버의 스킴렉톤으로 직렬화된 스트림 객체로 전달된다.

스켈렉톤은 서버 내에 있는 객체의 메소드를 호출하고 그 결과값을 다시 스텝으로 전달하여

클라이언트에 응답할 수 있도록 한다. 이를 통해 멀리있는 객체를 로컬에 있는 것처럼 쓸 수 있다[2].

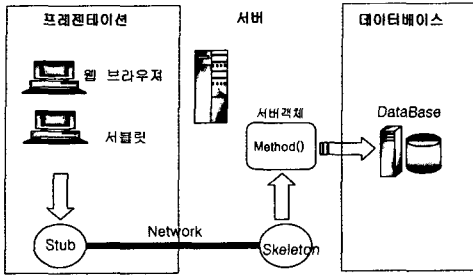


그림 1 분산 객체 아키텍처

2.2. 컴포넌트 기반 소프트웨어 개발(Component Based Software Development)

CBSD 는 독립적인 기능을 담당하는 다양한 컴포넌트 소프트웨어의 집합에서 해당 업무의 수행에 필요한 기능을 담당하는 하나 이상의 컴포넌트를 결합하여 해당업무를 위한 소프트웨어를 개발하는 기술이다[3]. CBSD 절차는 다음의 6 단계를 갖는다.

- 1) **요구분석:** 컴포넌트 명세를 인터페이스와 그 서비스 기능으로 추출한다.
- 2) **컴포넌트 기반 설계:** 해당 응용 소프트웨어를 구현하는데 필요한 컴포넌트의 인터페이스와 적용 컴포넌트를 추출한다.
- 3) **소요 컴포넌트의 획득:** 적합한 컴포넌트를 찾지 못한 경우 해당 컴포넌트를 구현한다.
- 4) **소프트웨어 조립:** 획득된 컴포넌트의 집합에 대해 전체적으로 원하는 기능을 수행 할 수 있도록 통합한다.
- 5) **수행 및 평가:** 실제 수행을 통해 성능과 기능을 평가한다.
- 6) **유지 보수:** 각 단계의 산출물을 관리하며, 재사용성 향상을 위해 품질 매트릭스를 통한 컴포넌트 품질 관리를 수행한다[4].

3. 서블릿의 EJB 변환

3.1 서블릿의 EJB 변환 절차

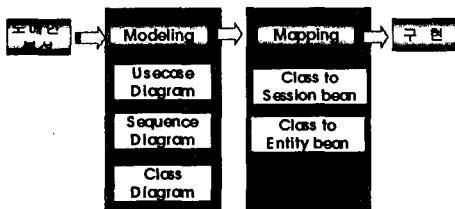


그림 2 EJB component 변환 과정

본 논문에서 서블릿을 EJB 로 Component 하는 절차는 그림 2 와 같다. 먼저 기존의 서블릿에서 제공하는 시스템의 도메인을 분석하고 UML 을

이용하여, 같은 기능을 제공하는 시스템을 모델링을 한다[6]. 모델링한 Diagram 들을 바탕으로 클래스의 속성과 메소드를 기준으로 Entity Bean 과 Session Bean 으로 나누어 매핑하는 과정을 거쳐 구현한다.

3.2. 서블릿 분석

3.2.1. 서블릿 아키텍처

서블릿 구조는 그림 3 과 같이 브라우저, 웹서버, DB 서버 3-계층으로 되어있다. 클라이언트가 웹브라우저에서 HTTP 를 통해 서버에 원하는 서비스 요청을 하면, 서버측에서는 invoke 를 생성해서 서블릿 서비스를 독립적으로 수행하게 한다. 필요에 따라, 서블릿이 데이터베이스 트랜잭션과 함께 비즈니스 로직을 모두 담당하기 때문에 서블릿 게시판은 비즈니스 로직의 재사용성을 높이지 못한다.

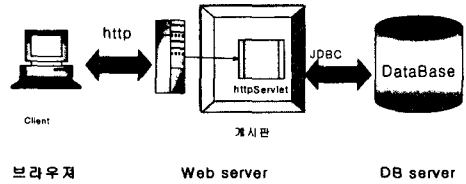


그림 3 서블릿 게시판 아키텍처

3.2.2. 서블릿 게시판 기능 분석

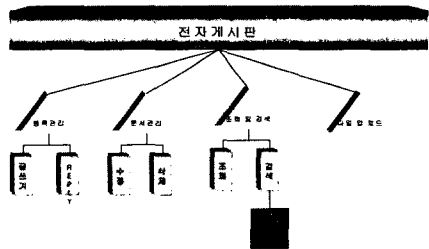


그림 4 서블릿 게시판 기능 분석

자료 공유, 교환을 위한 공간으로 서블릿 게시판은 그림 4 와 같이 조회, 쓰기, 수정, 삭제, 파일 업로드 기능이 제공된다. 쓰기는 날짜 순서로 등록되어 최근의 글을 먼저 볼 수 있도록 하며, 해당 자료에 대한 답변 글이 등록 되기도 한다. 특정 자료 검색을 위해 제목, 이름으로 검색, 조회가 가능하다. EJB 게시판도 서블릿 게시판의 기능을 그대로 제공한다.

3.3. 컴포넌트화를 위한 설계

3.3.1. EJB 아키텍처

그림 5 를 살펴보면, EJB 구조는 클라이언트 서버, 웹 서버, DB 서버로 3-계층 웹 아키텍처를 이루지만, 내부적으로 서블릿 게시판과 다르게 서블릿을 클라이언트에 포함한다. 서블릿은 물리적으로 서버 내에 존재하지만 클라이언트측의 UI 관련 로직만을 담당하여 클라이언트 서버 역할을 하도록 한다. 서블릿에 포함 되었던 비즈니스 로직은 EJB 의 session bean 에서 관리하고, 서블릿의 데이터베이스 로직은

entity bean 에서 담당하여 논리적으로 3-계층 아키텍처를 이루도록 한다. EJB 게시판은 여러 사용자가 동시에 DB 에 접근을 요청하는 경우 각 객체의 인스턴스를 효율적으로 관리할 수 있으며, 비즈니스 로직을 구별하여 관리하므로 비즈니스 로직의 재사용성을 높일 수 있다.

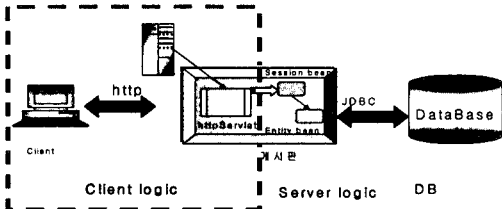


그림 5 EJB 게시판 아키텍처

3.3.2. UML 을 이용한 Component 모델링

1) Use-Case Diagram

서블릿 게시판의 기능 분석을 바탕으로 같은 기능을 제공하는 컴포넌트를 설계한다. 게시판의 Use-Case Diagram 은 아래 그림 6 과 같다.

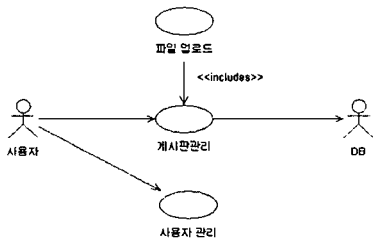


그림 6 게시판 Use-Case Diagram

2) Sequence Diagram

UI 를 담당하는 서블릿 클라이언트에서 클라이언트의 요청에 대한 서비스를 수행하기 위해 session bean 인 SessionBoard 를 호출한다. SessionBoard 는 서비스 처리에 대한 DB 데이터가 필요한 경우 entity bean 인 EntityBoard 의 함수를 호출한다. EntityBoard 에서는 DB 에 직접적으로 접근하여 데이터를 저장하고 가져온다. 그림 7 은 게시판 관리의 Sequence Diagram 이다.

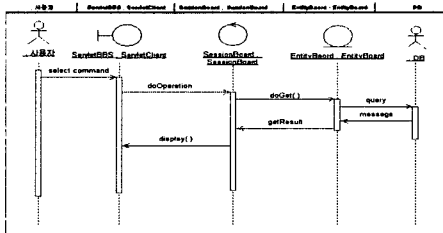


그림 7 게시판 관리 Sequence Diagram

3) Class Diagram

게시판 클래스는 사용자의 UI 를 담당하는 boundary class 인 ServletClient class, DB 의 내용을 나타내는 entity class 인 EntityBoard, entity class 와 boundary class 를 연결하며 business logic 을

관리하는 control class 인 SessionBoard 로 구별할 수 있다. 게시판 관리 Class Diagram 은 아래와 같다.

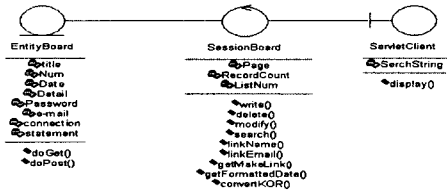


그림 8 게시판 관리 Class Diagram

3.4. 서블릿에서 EJB 로의 매핑

모델링 단계를 마친 후, EJB 로 매핑하기 위해서는 표 1 과 같이 서블릿 클래스의 데이터 속성과 그 데이터를 사용하는 메소드를 기준으로 세션 빈과 엔티티 빈으로 나누어 추출한다.

표 1 서블릿의 EJB 로 매핑 기준

서블릿		EJB	
Attribute A	- 영구적인 객체 데이터 속성 - DB 데이터를 기술	Entity Bean	Bean-managed Entity Bean - 프로그래머가 DB 데이터를 코드로 관리
Method a()	- Attribute A 를 참조하여 수행 - DB 와 관련된 직접적인 트랜잭션 처리		Container-managed Entity Bean - 컨테이너가 자동적으로 DB 내의 데이터를 관리
Attribute B	- 일시적인 객체 데이터 속성	Session Bean	Stateful Session Bean - 데이터의 속성이 다른 함수에서 참조
Method b()	- Attribute B 를 참조하여 수행 - DB 트랜잭션과 관련없는 workflow		Stateless Session Bean - 데이터의 생명주기가 짧고 하나의 메소드에서 참조

3.4.1. Entity bean 으로 매핑

표 1 에서 우선 객체의 데이터 속성이 영구적 속성을 지니고 있거나, DB 의 데이터를 나타내는 attribute A 가 존재하는 경우, 이 데이터와 직접적으로 수행하는 메소드 a()을 entity bean 으로 매핑한다. 서블릿 내에서 DB 에 저장된 게시판 자료를 읽고, 저장 하는 메소드인 doGet(),doPost()가 entity bean 으로 매핑된다.

1) Container-Managed Persistent Entity

EJB 컨테이너에 의해 자동적으로 DB 내의 데이터를 관리 할 때 쓰인다.

2) Bean-Managed Persistent Entity

개발자가 데이터와 DB 의 대응을 코드로 관리하게 하는 경우 사용한다.

3.4.2. Session bean 으로 매핑

표 1 에서 객체의 데이터가 임시적인 속성을 지닌 attribute B 인 경우, 이 데이터를 참조하거나 사용하는 workflow 부분의 메소드인 b())를 추출하여 session bean 으로 매핑한다. DB 와 관련된 직접적 트랜잭션을 제외한 메소드를 매핑한다. Session bean 은 DB 검색과 조회에 사용되며, 재사용하여 다른 시스템에서 연동할 수 있도록 한다.

1) Stateful Session

데이터의 속성이 다른 함수에 의해서 참조되는 대화식 속성인 경우 사용한다.

2) Stateless Session

데이터의 생명주기가 짧고 하나의 함수에서만 쓰여 질 때 해당 메소드들을 stateless session 으로 매핑한다.

3.5. 구현

아래 그림 9 의 EntityBoard.java 는 entity bean 의 remote 인터페이스이다. EntityBoard 에서는 DB 의 Data 를 직접 읽고, 저장하는 트랜잭션을 정의한다. EntityBoardBean.java 에서는 EntityBoard 에서 정의된 메소드를 doPost(PrintWriter out)과 doPost(PrintWriter out)을 이용해 구현한다.

```

Public interface EntityBoard extends EJBObject{
    public void loadBBS() throws RemoteException;
    public void saveBBS() throws RemoteException;
}
    
```

그림 9 EntityBoard.java

그림 10 은 Session Bean 의 remote 인터페이스 코드이다. Session bean 에는 DataBase 와 직접 관계하지 않는 workflow 를 정의하고, SessionManageBoardBean 에서 정의된 함수에 대해 구현한다.

```

import javax.ejb.*;
import java.rmi.RemoteException;
import java.rmi.Remote;
public interface SessionManageBoard extends EJBObject{
    public void write() throws java.rmi.RemoteException;
    public void delete() throws java.rmi.RemoteException;
    public void modify() throws java.rmi.RemoteException;
    public void search() throws java.rmi.RemoteException;
    public String getLinkName() throws java.rmi.RemoteException;
    public String getLinkEmail() throws java.rmi.RemoteException;
    public String getFormattedDateTime(Date date) throws java.rmi.RemoteException;
    public String getHTMLTitle(String title) throws java.rmi.RemoteException;
    public String getHTMLTitle(String title) throws java.rmi.RemoteException;
}
    
```

그림 10 SessionManageBoard.java

다음은 게시판에 등록된 글 중 글쓴이로 해당 목록을 검색한 결과이다.

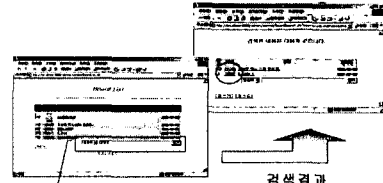


그림 11 게시판 검색 Interface

4. 결론 및 향후 연구과제

본 논문에서는 서블릿을 EJB 로 매핑 하는 과정을 살펴 보았다. 서블릿의 EJB 의 변환은 기존의 소프트웨어를 재사용하였기 때문에 유지 보수 비용을 절감 할 수 있는 장점이 있다. 또한 EJB 를 통한 컴포넌트화는 소프트웨어를 서버측의 비즈니스 로직과 데이터를 캡슐화 하여 서버측에 컴포넌트로 제공한다. 따라서 개발자는 컴포넌트화된 게시판을 독립적으로 다른 시스템에 조립하여 쓸 수 있다. 한편, 사용자는 게시판의 내부적 비즈니스 로직을 구현하지 않고도 인터페이스만 유지하며 계속 사용할 수도 있다.

향 후 연구에서는 서블릿 이외의 소프트웨어를 EJB 로 변환하여 재사용 할 수 있는 방법과 효율적인 컴포넌트 추출 방법론에 대해 연구하고자 한다.

5. 참고 문헌

[1] 최재영, 최충명, 유재우, "프로그래머를 위한 Java 2," 2000.
 [2] Richard Monson-Haefel, Enterprise JavaBeans, O'REILLY, 1999.
 [3] Butler Group, Component-Based Development: Application Delivery and Intergration Using Componentised Software, 1998.
 [4] 서동수, 홍기형, "컴포넌트 기반 소프트웨어 개발 프로세스," 제 12 권 제 3 호, 1999, 9.
 [5] Ed Roman, Mastering Enterprise JavaBeans, WILEY, 1999.
 [6] Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, Addison Wesley, 1999.