EJB 기반의 사원 정보 관리 컴포넌트 개발에 관한 연구

이향숙[1], 박진호, 류성열
숭실대학교 컴퓨터
{aromas, jhpark}@selab.ssu.ac.kr, syrhew@computing.ssu.ac.kr

A Study on EJB-Based Employee Information Management Component Development

Hyang-Sook Lee, Jin-Ho Park, Sung-yul Rhew
School of Computing Soongsil University

요약

소프트웨어에 대한 요구사항이 증가하고 그 규모가 점차 커대, 복잡해짐에 따라 소프트웨어의 재사용성은 주목할만한 이슈로 부각되고 있으며 많은 기법이나 지침이 제안되고 있다. 특히 자 예산으로 짧은 개발 기간 동안 기존의 시스템을 버리지 않고 새로운 시스템을 구축하려는 기업들의 요구에 맞게 최근 EJB기반의 컴포넌트 개발은 재사용성을 확보할 수 있는 가장 주목 받는 방안으로 제시되고 있다. 본 논문에서는 사원정보관리 어플리케이션에 EJB를 적용해 컴포넌트로 함으로써 컴포넌트 개발사 실무적인 유용성을 검증해 본다.

1. 서론

기업 시스템의 변화가 증가함에 따라 새로운 시스템에 대하여 현재 시스템의 재개발이 증가하고 있다. 어플리케이션과 데이터가 복잡해지고 분산 시스템의 요구에 대해 기업은 기존의 시스템을 버릴 수 없기 때문에, 기존의 시스템을 저 예산의 짧은 개발 기간동안 새로운 시스템으로 개발하기를 원한다. 이런 이유로 기업 정보 시스템은 빠르게 컴포넌트 기반의 소프트웨어로 옮겨가고 있다. 컴포넌트 기반의 소프트웨어인 EJB 에서 제공하는 컴포넌트 트랜잭션 모니터는 브로커링, 트랜잭션 관리, 보안, 퍼서스턴스, 그리고 동시성 제어등 복잡한 부분들을 자동으로 관리해 줌으로써 분산 개발을 단순화 시켜주는 강력한 컴포넌트 기반 환경을 제공한다. EJB 컨테이너가 트랜잭션, 동시 사용자에 대한 제어, 멀티 쓰레딩 시 데이터 무결성, 보안 등의 문제를 담당하므로 EJB 인터페이스는 컴포넌트의 기능을 규정짓고, 구현이나 플랫폼, 환경 의존등에 관한 것을 드러내지 않으므로 어떤 플랫폼에서도 컴포넌트를 사용할 수 있다[1].

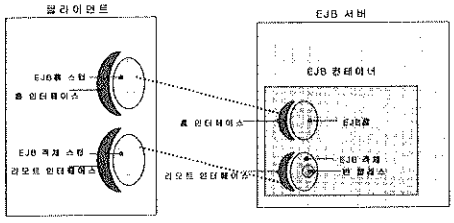
본 논문에서는 컴포넌트 기반의 기법 중 EJB를 사용하여 기업의 독립 실행 방식의 사원정보관리 어플리케이션을 분산환경에 적용되어질 수 있도록 하기 위해 기존의 JAVA 로 구현된 사원정보관리 어플리케이션을 EJB를 사용하여 컴포넌트화하는 설계 기법을 제안하였다.

본 논문의 구성은 다음과 같다. 2 장에서는 EJB 의 기본적인 구조와 트랜잭션 처리에 대한 간략한 소개를 기술하고, 3 장에서는 EJB 를 기반으로 한 사원정보관리 컴포넌트의 설계 과정을 설명하였다. 4 장에서는 제안된 설계 과정을 통해 사원정보관리 컴포넌트를 구현하였고 5 장에서는 향후 연구 방향과 결론을 맺는다.

2. 관련 연구

2.1 EJB

EJB(Enterprise JavaBeans)는 CORBA나 자바 RMI 같은 분산 객체 기술과 서버측 컴포넌트 모델의 개념을 결합하여 어플리케이션 개발이라는 일 자체를 극히 단순화시키는 기술이다. 보안, 리소스 풀링, 퍼서스턴스, 동시 제어, 트랜잭션 무결성 등 비즈니스 시스템이 갖추어야 할 요소들을 자동적으로 해결해 준다. EJB는 최신 Java 기술의 결정체로서 EJB가 발표되기 전에 발표된 여러가지 기술들을 기반으로 한다[1]. [그림1]은 EJB의 아키텍처이다.



[그림1] EJB 아키텍처

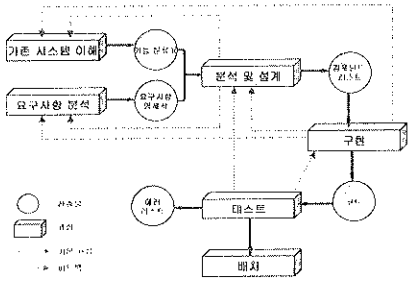
2.2 EJB 트랜잭션

트랜잭션은 함께 수행되어야 할 태스크들의 집합이다. 트랜잭션은 원자성(atomic)을 갖는다. 즉 하나의 트랜잭션이 성공하기 위해서는 해당 트랜잭션 내의 모든 태스크가 완료되어야 한다. EJB서버가 트랜잭션을 자동적으로 관리하므로, 빈 개발자는 트랜잭션과 관련된 빈을 관리하기 위해 별도의 API를 사용할 필요가 없다. 분산 배치할 때, 트랜잭션 속성을 주기만 하면 EJB서버는 런타임시 빈을 어떻게 관리해야 할지를 자동적으로 안다[1].

3. 본론

3.1 개발 과정

개발 과정은 [그림 2]에서 보는 바와 같이 기존에 JAVA로 개발되었던 어플리케이션을 이해하는 과정에서 추출한 기능[2]과 EJB기반의 컴포넌트로 새로운 어플리케이션을 구현하기 위한 요구사항분석을 통한 요구사항 명세서를 가지고 새로 개발할 어플리케이션에 대한 분석 및 설계를 한다. 이 과정에서 아래 <표1>을 적용해서 컴포넌트를 추출하고, EJB 스펙에 따라 컴포넌트를 구현한 후 개발 환경에 맞게 작동하는지 테스트를 한다. 테스트를 완료한 후엔 클라이언트에서 사용할 수 있도록 서버에 배치를 한다. 각 단계는 피드백을 통하여 조정된다[3].



[그림2] 개발과정

3.2 Component 추출 근거

기존의 어플리케이션을 EJB로 변환하기 위해서는 변환에 필요한 부분을 추출하기 위한 추출 기준이 있다. 각 부분이 어떤 기능을 수행하느냐에 따라 EJB 에서 제공하는 세션 빈과 엔티티 빈으로 구분할 수 있는데 세션 빈은 비즈니스 로직을 넣기에 가장 적당한 곳으로 엔티티 빈과 달리 퍼시스턴스를 갖지 않기 때문에 세션 빈 내의 어떤 것도 데이터베이스와 연결되지 않고, 세션들 간에 저장되지 않는다. 세션 빈은 자원과 작업을 관리하지만 데이터를 표현하지는 않는다. 엔티티 빈은 데이터를 객체화하고, 명사로 표현될 수 있는 비즈니스 개념을 표현하는 데에 유용하고 현실 세계의 상태와 동작을 표현하며 개발자가 특정한 개념과 관련된 데이터와 비즈니스 룰을 캡슐화 하도록 한다. 이는 즉 데이타가 퍼시스턴스를 갖는 경우에 데이터베이스와 연결이 되고, 세션들간에 저장되어질 필요가 있는 경우에 엔티티 빈이 되는 것이다. 세션 빈과 엔티티 빈은 세션 빈의 상태 유지 필요에 따라 상태 유지 세션 빈(Stateful Session Bean)과 무상태 세션 빈(Stateless Session Bean)으로 나뉘고 엔티티 빈은 빈의 관리 주체에 따라 컨테이너 관리 엔티티 빈(Container-Managed Entity Bean)과 빈 관리 엔티티 빈(Bean-Managed Entity Bean)으로 나뉠 수 있다. 컨테이너 관리 엔티티 빈은 그 퍼시스턴스가 EJB 컨테이너에 의해서 자동으로 관리된다., 빈 관리 퍼시스턴스를 사용하는 빈은 이런 모든 작업을 개발자가 명시적으로 한다. 빈 개발자는 데이터베이스를 다루는 코드를 반드시 사용해야 한다[4][5]. 컴포넌트 추출 기준을 정리하면 아래의 [표1] 같다

[표1] 컴포넌트 추출기준 표

기준	Session Bean		Entity Bean	
	Stateless	Stateful	Container Managed	Bean Managed
클라이언트와의 직접 통신	✓	✓		
비즈니스 로직 모델	✓	✓		
영속성 유지			✓	✓
대화상태 유지		✓		
데이터베이스 필요성			✓	✓
컨테이너의 자동 관리			✓	
사용자의 직접 코딩 관리				✓
공유 접근			✓	✓
프라이머리 키 필요성			✓	✓
파라미터 매칭		✓	✓	✓

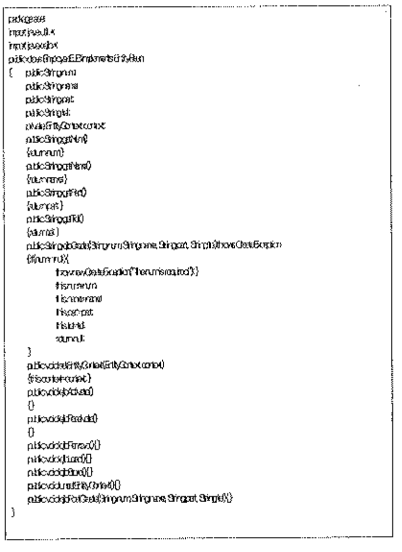
3.3 EJB 기반의 컴포넌트화

사원정보관리 어플리케이션을 반으로 추출할 때 위의 [표1]에서 제시한 컴포넌트 추출 기준을 기반으로 한다. 사원정보관리 어플리케이션은 대량의 사원에 대한 정보를 영속적으로 유지해야 할 필요가 있고, 동시에 여러 사람의 접근을 유지해야 하며, 프라이머리 키로써 사원에 대한

정보를 검색할 수 있어야 하므로 엔티티 빈으로 도출한다[4].

3.3.1 임플로이 엔티티 빈 코드

아래의 [그림3]은 임플로이 빈(EmployeeEJB.java)의 코드로 사원정보를 관리할 수 있는 메소드들을 제공한다.

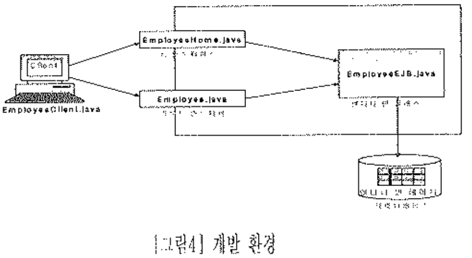


[그림3] 임플로이 엔티티 빈

4. 구현

4.1 개발 환경

개발 환경은 위의 [그림4]과 같다. 클라이언트는 리모트 인터페이스(Employee.java)를통해서 엔티티 빈 클래스(EmployeeEJB.java)의 인스턴스를 할당 받아 데이터베이스에 데이터를 저장하거나 데이터를 읽어 온다. 엔티티 빈 클래스의 라이프사이클은 홈 인터페이스(EmployeeHome.java)가 관리하며, 데이터베이스의 관리는 빈이 해주는 빈 관리 엔티티 빈이다[4].



[그림4] 개발 환경

4.2 구현 사례

EJB 기반의 컴포넌트 플랫폼에서 <표 1> 컴포넌트 추출 기준에 의해서 추출된 부분을 실제 컴포넌트로 구현한다. 서버에 놓여질 빈은 홈 인터페이스(EmployeeHome.java)와 리모트 인터페이스(Employee.java), 기본 키 클래스(EmployeePriKey.java) 엔티티 빈(EmployeeEJB.java)으로 함께 구현하고 , 클라이언트 쪽에 필요한 클라이언트 클래스(EmployeeClient.java)를 구현 한다. 컴포넌트에 대해서 전개 설명서(Deployment Descriptor)를 작성하고 서버에 놓여질 빈에 포함된 모든 클래스를 컴파일한 후 JAR 파일로 묶어준다. 데이터베이스에 엔티티빈에 대한 스키마를 구축하고 빈을 서버에 배치한다. 클라이언트에선 클라이언트 클래스를 컴파일하고, 클라이언트 코드를 통해 최종적으로 서버 쪽 컴포넌트를 실행시킨다.

5. 결론

본 논문에서는 사원정보관리 어플리케이션을 사례 연구 도메인으로 하여 기존의 JAVA 로 구현된 사원정보관리 어플리케이션을 EJB 기반의 시스템으로 변환하는 과정을 구체적으로 제시하였다. EJB 의 특징인 위치 투명성을 통해 하나의 데이터베이스를 두고서 어느 곳에서든 데이터베이스의 정보를 입력, 수정, 삭제, 검색 할 수 있어 저장 공간의 절약이라는 장점을 가지게 되고, 컴포넌트의 재사용성을 확인할 수 있다. 이 과정에서 우리는 EJB 를 적용한 소프트웨어의 실용성과 활용성을 검증할 수 있다.

앞으로 개발한 연구 과제는 인사 관리 어플리케이션의 전반적인 모든 부분을 EJB 로 컴포넌트화 하여 기존의 어플리케이션을 버리지 않고 기업의 시스템을 재개발 할 수 있도록 연구할 것이다.

6. 참고 문헌

[1] Sun Microsystems, Enterprise JavaBeans Specification, Version 2.0, 2000.

[2] Cay s.Horstmann, Gray Cornell,TheSun Microsystems press Java Series,2000.

[3] 인사역, 조현훈, 류성열, "RUP 기반의 웹 마이그레이션 설계 모델 재사용에 관한 연구," 한국정보과학회,2000.

[4] Ed Roman, Mastering Enterprise JavaBeans and the Java2 Platform, Enterprise Edition,Wiley,1999.

[5] Richard Monson-Haefel, Enterprise JavaBeans,Oreilly,1999.