EJB 기반 엔티티 컴포넌트 커스터마이제이션

이용원° 장윤정 이경환
중앙대학교 컴퓨터공학과
{ywlee, yjjang, kwlee}@object.cau.ac.kr

EJB-based Entity Component Customization

Yongwon Lee° Yun-Jeong, Jang Kyung-Whan Lee
Dept. of Computer Science and Engineering, Chung-Ang University

요 약

최근 컴포넌트 사용과 컴포넌트 기반 개발 방법론을 이용한 전사적 자원 관리(ERP) 시스템과 정보 관리 시스템(EIS) 개발이 급속히 성장하고 있다. 그 시스템을 구성하는 엔티티 컴포넌트는 데이터베이스 질의 처리를 통해 자료의 영속성 처리를 한다. 그러나, 컴포넌트가 복잡한 질의 처리를 통해서 영속성을 유지해야 하는 경우는 데이터베이스간의 질의 처리가 호환되지 않는 문제점과 그를 해결하기 위해 모든 데이터베이스와의 연동을 지원하기에는 어려운 점을 가지고 있다. 엔티티 컴포넌트에서 각 데이터베이스 질의 연산을 추상 클래스 정의하고 이 추상 클래스와 엔티티 빈 간의 종속성 제거하고 이것을 통한 EJB 기반 엔티티 컴포넌트의 커스터마이제이션을 제시한다.

1. 서론

최근 컴포넌트의 사용과 컴포넌트 기반 개발 방법론을 이용한 시스템 개발이 급속히 성장하고 있다.[1] 컴포넌트 기반 개발 방법론에 의한 전사적 자원 관리(ERP : Enterprise Resource Planning) 시스템이나 정보 관리 시스템(EIS : Executive Information System) 분석에서 다음과 같은 구조가 빈번히 발생한다. 데이터를 생성하고 검색하고 수정하고 삭제하는 연산을 가지는 엔티티 객체가 있다. 이 객체들은 그들간의 의존성에 의해서 하나의 컴포넌트로 이루어질 수 있다. [2]

이러한 엔티티 컴포넌트는 자료의 영속성을 유지하기 위해서 데이터베이스를 사용한다. EJB 컴포넌트 모델에서 자료의 영속성을 위한 엔티티 빈이 데이터베이스의 복잡한 질의 처리를 통해서 자료 처리를 할 때 각 데이터베이스마다 조금씩 다른 질의 처리를 해야한다. 그 때문에, 각 데이터베이스마다 엔티티 빈을 제작해야 하는 문제를 야기 시킬 수 있다.

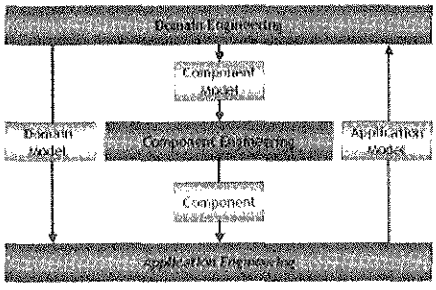
본 논문에서 먼저 기반 연구로서 컴포넌트 기반 개발 프로세스를 정의하고 본론에서 데이터베이스 질의 연산을 추상화한다. 추상화된 클래스와 빈과의 종속성을 제거하기 위한 모델링을 제시하고 이를 통해 엔티티 컴포넌트 커스터마이제이션 방법을 제시하고자 한다. 마지막으로 이 모델을 적용한 사례를 알아보도록 하겠다.

2. 기반연구

2.1 컴포넌트 기반 개발 프로세스

컴포넌트 기반 개발은 아래 [그림 1]과 같은 프로세스를 사용한다. 프로세스는 도메인 엔지니어링, 컴포넌트 엔지니어링, 어플리케이션 엔지니어링의 계층으로 구분한다. 도메인 엔지니어링으로부터 산출된 도메인 모델과 컴포넌트 모델은 어플리케이션 엔지니어링과 컴포넌트 엔지니어링에 반영되고, 컴포넌트 엔지니어링으로부터 산출된 컴포넌트는 어플리케이션 엔지니어링에 반영된다.

도메인 엔지니어링과 컴포넌트 엔지니어링이 어플리케이션 엔지니어링에 제대로 반영되었는지 확인한다. 어플리케이션 엔지니어링에서 산출된 어플리케이션 모델을 통해 검증한다.



[그림 1] CBD 개발 프로세스

2.2 컴포넌트의 진화

컴포넌트의 진화에는 두 가지가 있다. 컴포넌트 설계자가 설계한 소프트웨어 컴포넌트를 변경된 요구사항에 맞게 수정하는 것을 커스터마이제이션(customization)이라 하고, 개발자가 다른 사용 목적으로 서드 파티 컴포넌트를 적용하는 것을 어댑테이션(adaptation)이라고 한다.[3]

2.3 EJB (Enterprise JavaBeans)

EJB에서는 빈을 기본적인 컴포넌트 단위로 정의하고 있다. 빈에는 두 종류가 있는데, 세션 빈(Session Bean)과 엔티티 빈(Entity Bean)이 그것이다. 엔티티 빈에서 자료의 영속성을 보장하는 방법에는 빈 관리 영속성(Bean-Managed Persistence)과 컨테이너 관리 영속성(Container Managed Persistence)으로 나뉠 수 있다.[4]

2.4 DAO(Data Access Object)

DAO는 세션 빈을 간단 명료하고 해 주고, 엔티티 빈 위해 CMP로 쉽게 변경 할 수 있게 하며 서로 다른 데이터베이스와 스키마 처리를 위해 사용하고 객체-관계 매핑 툴을 지원하는 메커니즘을 제공한다.[1]

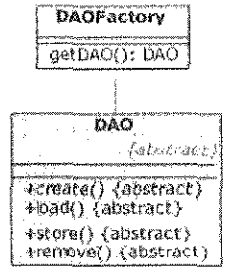
3. EJB 기반 엔티티 컴포넌트 커스터마이제이션

엔티티 컴포넌트는 엔티티 클래스로 구성된다. 엔티티 클래스는 모델을 표현하는 클래스로서 영속성을 유지해야 한다. 객체의 영속성을 유지하기 위해 EJB에서는 엔티티 빈이 사용된다. 엔티티 빈이 데이터베이스와 연동함에 있어서 복잡한 질의를 가지고 영속성을 유지해야 하는 경우는 데이터베이스간의 질의 처리가 호환되지 않는 문제점과 그 문제를 해결하기 위해 모든 데이터베이스와의 연동을 지원한다는 것은 어려운 문제이다.

위와 같은 문제를 해결하기 위해 질의연산을 추상화하여 DAO로 정의하고 그 DAO와 빈 간의 의존성을 제거함으로써 유연하고 확장 가능한 엔티티 컴포넌트의 커스터마이제이션 모델링 기법을 제시하고자 한다.

3.1 DAO(Data Access Object) 설계

질의연산을 위한 추상 클래스 DAO와 엔티티 빈과 DAO의 의존성을 분리하기 위한 DAOFactory 클래스를 정의한다. 아래의 [그림 2]는 데이터베이스 질의 처리를 위해 생성, 검색, 갱신, 삭제의 추상 메소드를 정의한 DAO 추상 클래스와 빈과의 의존성을 제거하기 위한 DAOFactory 클래스의 연관관계를 나타낸 한 예이다.



[그림 2] DAOFactory 와 DAO

3.2 엔티티 빈의 컨테이너 계약(Contract) 메소드 정의

엔티티 빈의 컨테이너 계약 메소드와 DAO의 추상메소드는 [표 1]과 같이 대응된다.

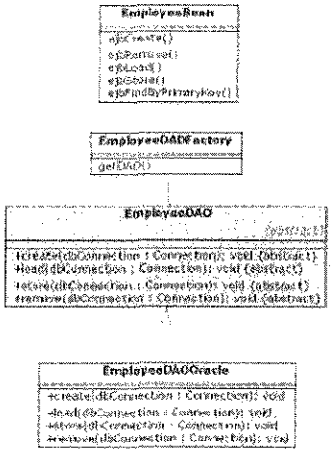
create 메소드는 자료의 삽입에 대한 질의를 정의하고, findByXXX 메소드는 자료의 검색에 대한 질의를 정의한다. load와 store 메소드는 데이터베이스와 빈의 동기화를 유지하기 위한 것으로 검색과 변경에 대한 질의를 정의하고 remove 메소드는 자료의 삭제를 위한 질의 처리를 정의한다.

컨테이너 계약 메소드	DAO 메소드
ejbCreate	create
ejbFind	findByXXX
ejbLoad	load
ejbStore	store
ejbRemove	remove

[표 1] 컨테이너 계약 메소드와 DAO 메소드

4. 적용

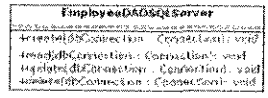
ERP 시스템에서 인사 및 기초자료 관리, 각종 증명서 및 양식의 처리를 위한 인사관리 컴포넌트 중 Employee 객체를 엔티티 빈을 표현한 것이다. EmployeeDAO는 Employee 엔티티 빈의 영속성을 처리한다. 아래의 [그림 3]은 EmployeeBean, EmployeeDAOFactory 클래스와 DAO 추상 클래스의 연관 관계를 클래스다이어그램으로 표현한 것이다.



[그림 3] EmployeeBean

추상 클래스 EmployeeDAO로부터 상속을 받아 정의한 EmployeeDAOOracle 클래스는 자료의 삽입, 검색, 저장, 삭제을 위한 최적화된 질의문와 데이터 변형 처리를 위한 메소드를 정의하고, 빈 클래스와의 종속성을 없애기 위한 EmployeeDAOFactory 클래스를 정의한다. 런타임에 빈 인스탄스가 EmployeeDAOFactory 인스탄스의 getDAO 메소드를 호출하면 getDAO 메소드는 데이터베이스의 메타 정보를 읽어 실제 EmployeeDAO의 인스탄스를 반환하고, 반환된 인스탄스는 빈 인스탄스에 바인딩이 이루어진다.

EmployeeBean 이 다른 데이터베이스에 연동될 경우 [그림 4]와 같이 그 데이터베이스를 위한 DAO 클래스만 정의하여 이전의 연동되던 클래스를 대치시키면 된다.



[그림 4] 대치될 DAO

5. 결론

컴포넌트는 어플리케이션의 한 부품으로 사용된다. 컴포넌트 기반 개발에서 컴포넌트 제작은 어플리케이션 분석, 설계 단계에서 충분한 고려가 이루어져야 그 역할을 충분히 할 수 있다. 또한 컴포넌트의 변경 가능한 부분은 유연성과 확장성이 고려되어야 재사용률을 높일 수 있다. 본 논문에서는 EJB 기반 엔티티 컴포넌트에서 데이터베이스간의 호환되지 않는 질의 처리 문제점을 해결하기 위한 모델을 제시하여, 이전 엔티티 빈을 데이터베이스별로 작성하였던 문제점을 엔티티 컴포넌트 구조의 변경 없이 대치 가능한 모델로써 커스터마이제이션 기능을 제공하였다.

향후 연구과제로 본 논문에 제시한 모델을 적용한 워크플로우의 커스터마이제이션으로 플러그앤플레이한 워크플로우를 변경할 수 있는 모델에 적용하고자 한다.

참고문헌

[1] Nicholas Kassem and the Enterprise Team, "Designing Enterprise Applications with the Java 2 Platform, Enterprise Edition", Sun Microsystems, 2000

[2] Peter Herzen, Oliver Sims, "Business Component Factory", Willy, 2000

[3] George T. Heineman, "An Evaluation of Component Adaptation Techniques", International Workshop on Component-Based Software Engineering, 1999

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

[5] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns", Addison-Wesley, 1995