

# 비즈니스 애플리케이션을 위한 EJB 컴포넌트

## 빈 유형 추출방법

\*김정옥, \*\*박옥자, \*\*\*유철중, \*\*\*\*장옥배

전북대학교 컴퓨터학과

\*kjo3852@hanmail.net

\*\*ojpark@cs.chonbuk.ac.kr

\*\*\*{cjyoo, okjang}@moak.chonbuk.ac.kr

### Extraction Technique of the EJB Component Beans Type for Business Application Domain

Jeong-Ok Kim, Oak-Cha Park, Cheol-Jung Yoo, Ok-Bae Chang  
Dept. of Computer Science, Chonbuk National University

#### 요 약

EJB의 애플리케이션 개발환경은 객체지향 분산처리를 지원하는 컴포넌트를 개발하고 분산 배치를 위한 컴포넌트 구조이다. EJB를 이용하여 개발한 애플리케이션은 컴포넌트 모델의 개념을 결합하여 비즈니스 프로그램의 개발이 쉽도록 단순화시켰으며, 보안성, 리소스 풀링, 영구성, 동시성, 트랜잭션 무결성을 자동적으로 해결해 준다. 본 논문에서는 이러한 개발환경에서 EJB 컴포넌트가 충분한 유연성을 가질 수 있도록 기능별로 효율적인 빈의 유형을 추출하는 방법을 찾고자 하였으며, EJB 빈의 유형별 특성에 따라서 빈 유형을 쉽게 분류할 수 있는 방법을 표준화하여 시스템 분석 및 설계시에 반영할 수 있도록 하였다. 본 논문의 제안모델은 단계별로 전 단계의 모델을 재사용하고, 객체를 쉽게 빈으로 매핑할 수 있도록 표현계층, 비즈니스 로직계층, 데이터계층으로 계층화하여 객체를 배치함으로써 빈의 유형을 쉽게 추출할 수 있는 방법을 제공한다.

#### 1. 서 론

Enterprise JavaBeans(이하 EJB)는 객체지향 분산 엔터프라이즈의 응용프로그램 개발 및 분산 배치를 위한 컴포넌트의 구조이다. EJB는 CORBA와 Java RMI의 분산 객체기술과 서버의 컴포넌트 모델의 개념을 결합하여 비즈니스 프로그램의 개발을 쉽게 단순화 시켰으며 보안성(security), 리소스풀링(resource pooling), 영구성(persistence), 동시성(concurrency), 트랜잭션 무결성(transactional integrity)을 자동적으로 해결해 준다[1, 2]. 본 논문에서는 개발된 EJB 컴포넌트가 충분한 유연성과 성능향상을 갖도록 설계하기 위하여 프리젠테이션 계층(presentation tier), 비즈니스 로직 계층(business logic tier), 데이터 계층(data tier)의 3계층으로 계층화하고 비즈니스 로직층을 상태 세션 빈, 무상태 세션 빈, 컨테이너관리 엔티티 빈, 빈관리 엔티티 빈의 효율적인 유형의 설계를 방법을 제안하고자 한다. 본 논문의 구성은 2장에서는 엔터프라이즈 빈의 유형별 특성을 설명하고, 3장에서는 EJB의 빈 모델링을 위한 빈 추출 단계를 설명했으며, 4장에서는 실제로 모델링에 의한 빈의 추출 사례를 설명하였다. 마지막 5장에서는 결론 및 향후 연구과제를 제시한다.

#### 2. EJB 빈의 특징

##### 2.1 EJB 빈의 종류

EJB에는 두 종류의 빈이 있는데 그 하나는 세션 빈으로 세션 빈은 상태유지 세션 빈과 무상태 세션 빈으로 구분되고, 다른 하나는 엔티티 빈으로 영속성 관리 방법에 따라서 컨테이너관리 엔티티 빈과 빈관리 엔티티 빈으로 구분된다. 빈들의 특성을 비교하면 [표 1]과 같다.

[표 1] EJB 빈의 특성비교

구분	세션빈		엔티티빈	
	무상태	상태유지	컨테이너관리	빈관리
목적	• 하나의 클라이언트의 작업	• 하나의 클라이언트의 작업	• 영속성 저장소에 존재하는 객체표시	• 영속성 저장소에 존재하는 객체표시
영속성	• 영속성을 갖지 않음	• 영속성을 갖지 않음	• 영속성을 가짐	• 영속성을 가짐
공유 접근	• 하나의 클라이언트에서만 접근 (특정 클라이언트에 할당되지 않음)	• 하나의 클라이언트에서 서안 접근 (특정 클라이언트에 할당)	• 하나 이상의 클라이언트에서 접근	• 하나 이상의 클라이언트에서 접근
특징	• 스와핑의 부하를 줄임 • 활성화, 비활성화가 없음 • 하나의 메소드로 전체작업처리	• 빈관리엔티티의 인스턴스변수가 메소드 호출시 특정 클라이언트 데이터를 캐시 • 클라이언트를 대신하여 워크플로우를 집합화하고 관리하는 단순한 인터페이스 제공	• 영속성이 컨테이너에해 자동관리 • DB의 정교한 매핑들이 필요 • 개발라기가 가장 간편	• 영속성 유지틀위한 목적 필요 • 특정 DB 타입과 구조에 종속
			• DB에 독립적으로 정의 가능	

**2.2 EJB 컴포넌트 빈의 구성**

엔터프라이즈 빈즈는 두 개의 인터페이스(리모트, 홈)와 두개의 클래스(빈, 프라이머리키)를 정의하여 구현하며 엔터프라이즈 빈즈는 Java와는 달리 리모트 인터페이스, 홈 인터페이스, 빈 클래스, 프라이머리 키의 4개 부분으로 구성된다. 이는 자바 빈즈의 이벤트 클래스, 정보 클래스 등과 마찬가지로 엔터프라이즈 빈즈에게 컴포넌트 특성을 부여하기 위해 사용되는데 [표 2]는 빈의 부분별 구현내용을 나타낸다[3].

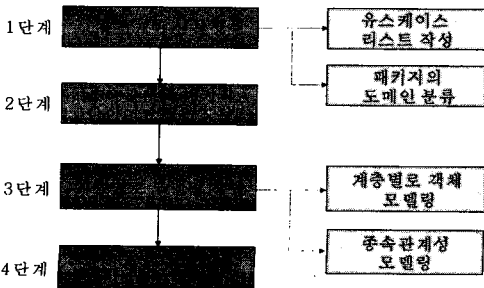
[표 2] 엔터프라이즈 빈즈의 구성요소

구성요소	정의
리모트 인터페이스 (Remote Interface)	- 빈을 수행하기 위해 외부에 제공되는 비즈니스 메소드를 정의
홈 인터페이스 (Home Interface)	- 새로운 빈의 생성, 제거 등의 라이프 사이클에 관련된 메소드 정의
빈 클래스 (Bean Class)	- 실제 빈의 비즈니스 메소드를 정의 - 리모트 인터페이스에 정의된 모든 메소드에 대응하는 메소드 구현 - 홈 인터페이스에 정의된 메소드에 대응하는 메소드 구현
프라이머리 키 (Primary Key)	- 빈을 데이터베이스에서 얻어내기 위해 데이터베이스로의 포인터를 제공

**3. 빈 유형 추출을 위한 모델링 방법**

**3.1 EJB의 빈 추출 방법**

빈의 추출을 위한 모델링 방법을 4단계로 분류할 수 있는데 1단계는 도메인별 유스케이스 다이어그램을 작성하고 상세하게 유스케이스 리스트를 작성한다. 이 유스케이스 리스트를 기반으로 비즈니스 도메인을 영역별 패키지로 분류한다. 2단계는 유스케이스에 근거한 종속성 관계의 업무부문별 분류에 의한 패키지 다이어그램을 작성한다. 3단계는 계층화 설계 단계로서 패키지 다이어그램을 표현 계층, 비즈니스 로직 계층, 데이터 계층으로 계층별 객체를 모델링하고 계층별 객체들의 종속성 관계를 작성한다. 4단계는 비즈니스 로직계층을 상세하게 분류하여 빈을 추출하고 빈의 유형을 설계한다. [그림 1]은 빈 유형 추출 관점에서 본 모델링 단계를 나타내고 있다.



[그림 1] EJB 빈 추출 단계

**1단계 : 유스케이스 작성**

사용자 요구분석 단계의 유스케이스를 작성하고 작성된 유스케이스별 유스케이스 리스트를 작성한다. 업무의 도메인에 따라서 유스케이스를 패키지로 분류하여 작성한다. 이 단계까지를 [표 4.1]에서 보여주고 있다.

**2단계 : 종속관계의 패키지 다이어그램을 작성**

1단계의 유스케이스 리스트에서 분류한 패키지간의 데이터 흐름의 종속관계를 UML을 이용하여 패키지 다이어그램으로 작성한다.

**3단계 : 계층별 모델링 작업**

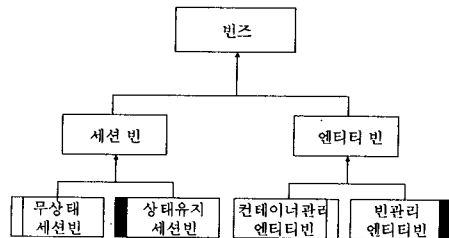
계층화 설계 단계로서 패키지 다이어그램을 표현 계층, 비즈니스 로직 계층, 데이터 계층으로 계층별 객체를 모델링한다. 표현계층은 패키지 다이어그램에 있는 패키지를 배치한다. 비즈니스 로직 계층은 유스케이스 다이어그램의 유스케이스를 배치한다. 데이터 계층은 패키지별 영속성 데이터를 추출하여 데이터 베이스에 저장할 필요가 있는 데이터를 DB로 모델링한다. 이렇게 계층별 배치가 끝나면 표현 계층, 비즈니스 로직 계층, 데이터 계층의 종속관계의 계층별 객체 데이터 흐름을 UML의 종속관계 모델요소로 링크한다.

**4단계 : 빈 유형의 상세설계**

비즈니스 로직 계층의 각각의 객체가 종속성과 영속성에 따라서 빈의 특성에 맞는 빈 유형을 결정한다. 빈의 유형을 결정하는 기준은 다음과 같다. 첫째, 무상태 세션 빈은 표현 계층에만 종속되어있는 비즈니스 로직 계층의 객체. 둘째, 상태 세션 빈은 비즈니스로직 계층에 종속되어있는 비즈니스로직 계층의 객체. 셋째, 컨테이너관리 엔티티 빈은 데이터 계층에 종속되어있는 비즈니스 로직 계층의 객체. 마지막 빈관리 엔티티 빈은 데이터 계층에 종속되어 있는 비즈니스 로직 계층의 객체로서 비즈니스 로직이 필요한 객체이다.

**3.2. EJB의 빈 유형의 표기법**

UML에서는 빈을 표현하기 위한 모델요소가 없으므로 본 논문에서는 빈을 표기하기 위하여 빈 유형을 [그림 2]와 같이 표기하기로 하였다.



[그림 2] 빈 유형별 표기법

**4. 빈 추출 모델링 사례**

**1단계 : 유스케이스 작성**

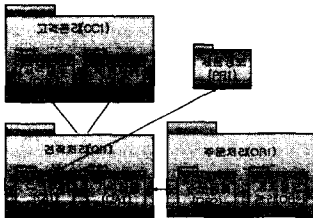
요구분석 단계로 도메인별 유스케이스 다이어그램을 작성하고 유스케이스를 기반으로 도메인을 패키지 영역으로 분류하기 위하여 좀더 상세하게 유스케이스 리스트를 작성하고 유스케이스를 패키지로 분류하는데 [표 3]은 그 예를 나타내고 있다.

[표 3] 유스케이스 리스트

유스케이스명	내용	패키지분류
고객정보조회	고객정보조회한다	CC1
고객정보	고객정보를 등록한다	CC2
비밀번호관리	고객 인증을 점검 등록한다	CP1
제품관리	제품정보를 등록한다	PR1
견적처리	고객이 주문한 제품정보를 계산한다	QI1
견적항목편집	견적정보 항목을 편집한다	QI2
가격산출	견적제품 가격을 계산한다	QA1
주문처리	주문한다	OR1
주문정보등록	주문정보를 등록한다	OR2
계좌잔고조회	계좌 잔고를 조회한다	OB1
계좌잔고등록	계좌 잔고를 갱신한다	OB2

**2단계 : 종속성 관계의 패키지 다이어그램 작성**

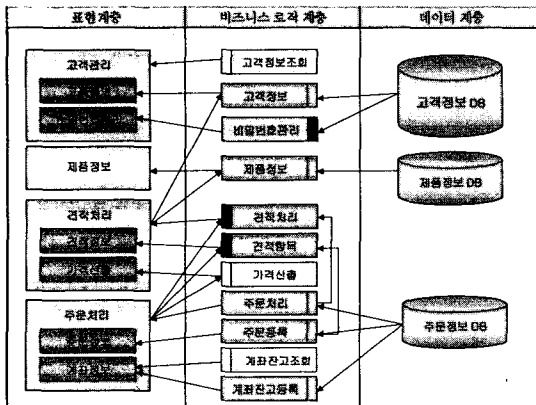
분석 단계로 유스케이스 다이어그램을 이용하여 종속성 관계를 표현하기 위한 모델링 단계로서 패키지 다이어그램을 계층화하고 분류된 패키지간의 유스케이스 인터페이스를 순서적으로 상세화하여 패키지 다이어그램을 작성하는 단계이다. [그림 3]은 패키지의 종속성을 패키지 다이어그램으로 작성한 예를 나타내고 있다.



[그림 3] 패키지 다이어그램

**3단계 : 계층별 객체 모델링**

계층화 단계는 첫째, EJB 컴포넌트를 모델링하기 위한 본격적인 단계로서 분석 과정에서 추출한 객체를 작업 유형에 따라 추상화하여 계층별로 분류하는 과정이다. 표현계층은 패키지 다이어그램을 이용하고, 비즈니스 로직 계층은 유스케이스 다이어그램의 유스케이스 리스트를 이용하여 작성한다.



[그림 4] 계층별 빈의 추출결과

데이터 계층은 패키지별 영속성 데이터를 이용하여 객체를 모델링한다. [그림 4]는 본 사례에서 분석한 프리젠테이션 계층, 비즈니스 로직계층, 데이터 계층의 3계층으로 계층화한 객체 모델링의 예를 나타내고 있다. 표현 계층, 비즈니스 로직 계층, 데이터 계층의 종속관계의 계층별 객체 데이터 흐름을 연결한다.

**4단계 : 빈 유형의 상세 설계**

계층별 객체 모델링을 근거로 하여 빈을 추출하고 빈의 유형을 정확하게 분류하는 단계로서 [표 4]는 3장의 빈 유형의 상세설계 기준에 의하여 유형을 설계하는 예를 나타내고 있다

[표 4] 빈 추출 및 유형 설계

빈	포함된 데이터	빈유형			
		세션빈	엔터티빈	무상태	상태유지
고객정보조회	고객ID, 성명, 주소, 주민번호, 나이, 직업			○	
고객정보	"				○
비밀번호관리	비밀번호암호화				○
제품정보	제품ID, 제품명, 가격, 단위, 용도, 규격				○
견적처리	고객ID, 견적수량, 단가, 세금, 견적금액, 견적일자			○	
견적항목	품목ID, 제품명, 규격, 수량, 할인율			○	
가격산출	가격계산	○			
주문처리	주문ID, 수량, 단가, 세금, 금액, 주문금액, 주문일자				○
주문등록	품목ID, 제품명, 규격, 수량, 할인율				○
계좌잔고조회	계좌번호, 예금주명, 계좌잔고	○			
계좌잔고등록	계좌잔고갱신				○

**5. 결과 및 향후 과제**

본 논문에서 빈 유형의 추출방법은 첫째, EJB 환경에서 유연성 있고 재사용 가능한 컴포넌트의 설계를 위하여 기능별로 효율적인 빈의 유형을 추출하기 위한 절차를 찾아내고자 노력하였으며, 둘째는 요구분석 단계에서 상세설계 단계까지의 모델링을 프로토타입 형태로 재사용함으로써 빈 추출방법을 찾을 수 있도록 하였다. 향후 연구과제로는 빈 추출 방법을 이용한 EJB 시스템의 모델링 기법을 개발하여 빈 유형 설계의 효율성을 검증할 필요가 있다. 또한 EJB 컴포넌트의 효율적인 설계를 위한 빈 유형 설계뿐만 아니라 값에 의한 객체전달, 공동인터페이스 구현등의 설계지침을 모델링에 반영할 수 있도록 하기 위하여 UML을 확장하고 설계시에 반영시킬 수 있는 모델링 방법을 개발할 필요가 있다.

**참고문헌**

[1] Sun Microsystems Inc. "Enterprise JavaBeans Specifications", URL:http://www.javasoft.com  
 [2] Ed Roman, "Mastering Enterprise JavaBeans", John Wiley & Sons, Inc., 1999.  
 [3] Richard Monson-Haefel, "Enterprise Java Beans", O'Reilly, 1999