2000년 한국정보처리학회 추계 학술발표논문집 제7권 제2호

EJB 컴포넌트 생성을 위한 도구 설계 및 구현

국윤규*, 김운용*, 최영근*
*광운대학교 컴퓨터과학과
e-mail:ykkook@cs.kwangwoon.ac.kr

Design and Implementation of Tool for EJB Component Producting

Youn-Gyou Kook*, Woon-Yong Kim*, Young-Geun Choi*
*Dept of Computer Science, Kwang-Woon University

요약

인터넷의 급격한 발달로 인하여 기존의 클라이언트/서버환경으로 구성된 웹 환경에서는 서비스 제공이 원활하지 못하고, 시스템의 안정성 및 신뢰성이 미약하게 되어 이를 보완하기 위하여 새로운 환경인 분산 처리 기술의 필요성이 부각되었다. 분산 객체 애플리케이션 서버가 이러한 문제점을 해결하였지만, 서버 구축에 대한 어려움은 남아있었다. 그러나 EJB 컴포넌트의 사용으로 애플리케이션 서버 구축이 한결 쉬워졌다. EJB는 자바 플랫폼 상에서 운영되는 서버 컴포넌트 기술이다. 컴포넌트 개발 과정에서는 일정한 모듈의 중복으로 인한 개발 시간과 비용의 낭비, 컴포넌트 유지 보수에 대한 어려움, 명세에 따른 개발의 어려움이 있다. 따라서 본 논문에서는 컴포넌트의 재사용성과 모듈성을 극대화하고, 코드 최적화 및 개발 시간과 비용 절감, 또한 개발자간의 의사소통을 원활하게 할 수 있는 개발 표준이 정립될 수 있도록 컴포넌트를 분석하여 모듈을 추출하고, 모듈의 정보저장소를 설계하며, 이를 이용한 EJB 컴포넌트 생성 도구를 구현하였다.

1. 서론

최근 컴퓨터와 통신의 급격한 발달로 인하여 인터넷 사용이 보다 편리해지고, 사용자들의 급격한 증가 추세로 인하여 사용자 요구에 서비스를 제대로 제공하기가 힘들다. 현재 인터넷은 소규모의 B2C 전자 상거래 환경에서 대규모의 B2B 시장으로 확대되어 더욱 많은 발전을 이룩해왔다. 인터넷은 서비스 제공의 원활함, 그리고 보다 안정적이고 신뢰할 수 있는 인터넷 환경의 필요성으로 인하여 새로운 분산 처리 기술로서 3계층 구조의 미들웨어인 애플리케이션 서버가 그 역할을 하였다. 애플리케이션 서버 구축의 난해함은 서버 컴포넌트 기술인 EJB의 사용으로 한 결 쉬워졌다[6]. 그러나 시스템 설계에 따른 일정한 모듈의 중복[3]으로 개발 시간과 비용이 낭비된다. 또한 명세의 정확한 이해 부족으로 인하여 컴파일시 에러 발생율이 높고, 엔터프라이즈 서버 구축에는 유지보수의 어려움이 있으며 코드 일관성과 최적화의 결여로 인한 컴포넌트의 성능 저하를 가져온다[5,7].

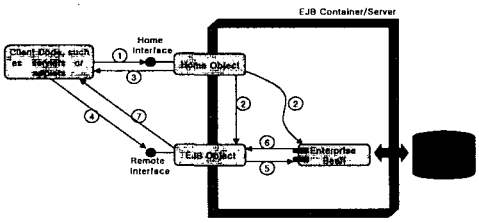
따라서 본 논문에서는 EJB 컴포넌트의 일정한 모듈을 추출하여 모듈에 따른 정보 저장소를 설계하고, 사용자의 선택에 따라 정보저장소의 모듈을 이용한 EJB 컴포넌트 생성 도구를 구현하였다. 사용자의 요구사항과 모듈을 분석하여 이를 기반으로 구성된 본 시스템은 재사용성과 모듈성을 극대화하고, 코드 최적화 및 개발 시간과 비용 절감을 야기할 수 있다. 또한 개발자들간의 의사소통을 원활하게 하기 위한 개발 표준이 정립될 수 있다.

본 논문의 구성은 2장에서 EJB 시스템에 대해서 간략하게 살펴보고, 3장에서는 컴포넌트의 분석에 따른 모듈을 추출하며, 이에 대한 정보저장소를 설계하고 4장에서 시스템을 설계 및 구현하며 응용 사례를 살펴본다. 마지막으로 5장에서 결론 및 향후과제에 대하여 기술한다.

2. 관련 연구

2.1 EJB 시스템 개요

EJB(Enterprise JavaBeans, EJB)는 엔터프라이즈 환경에서 애플리케이션 개발에 적합한 J2EE(Java 2 platform Enterprise Edition)의 서버 측면의 컴포넌트 기술이다[1,2,4,6]. EJB 시스템은 미들웨어인 애플리케이션 서버, 컨테이너, 그리고 컴포넌트로 구성되어 있다. 서버는 보안, 지속성, 트랜잭션 관리 및 컨테이너를 관리하고 컨테이너는 컴포넌트를 관리한다. 컨테이너에서 컴포넌트를 관리하는 EJB 시스템의 구동 절차[2]는 [그림 1]과 같다.



1. 클라이언트는 애플리케이션 서버의 홈 객체에 사용하고자 하는 객체 생성 요구
2. 객체를 생성하여 원격 인터페이스를 통한 빈 사용기능 및 사용할 빈이 활성화
3. 홈 객체는 원격 인터페이스를 통하여 사용할 수 있는 메소드의 참조자 전달
4. 클라이언트는 객체의 원격 인터페이스에 사용할 메소드를 호출
5. 클라이언트의 요구에 따라 빈 내부의 해당 메소드에 명령을 전달
6. 작업 완료한 빈은 객체에 메소드를 리턴
7. 빈의 작업 결과를 클라이언트에게 전달

[그림 1] EJB 시스템의 구동절차

2.2 EJB 시스템 설계 및 구현 과정

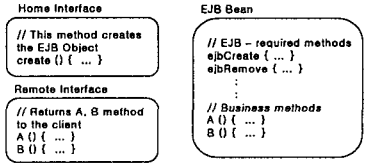
컴포넌트는 빈은 명세에 따라 세션 빈과 엔티티 빈으로 구분된다. 세션 빈은 클라이언트의 세션을 관리하기 때문에 클라이언트 연결과는 무관하고 EJB 저장소에서 제거될 때가지 존재한다.

세션 빈은 상태 보유에 따라 무상태 세션 빈과 상태유지 세션 빈으로 구분되고, 엔티티 빈은 관리 주체에 따라 컨테이너 관리 지속성과 빈 관리 지속성 엔티티 빈으로 구분된다[2,4,5]. 빈의 성격에 따른 차이점은 [표 1]과 같다.

무상태 세션 빈 (Stateless Session Bean)	하나의 빈이 여러 클라이언트에 대응된다. 클라이언트의 컨텍스트(상태)는 관리되지 않는다. 현재 트랜잭션이 동기화되지 않는다.
상태유지 세션 빈 (Stateful Session Bean)	클라이언트와 1:1로 대응된다. 클라이언트의 컨텍스트(상태)가 유지된다.
빈 관리 지속성 엔티티 빈 (Bean Managed Persistence Entity Bean)	지속성 관리를 위한 코딩이 필요하다. 위치 지정 시점에 지속성 관리 변경은 할 수 없다. 객체 관련성에 더 유연하게 제어할 수 있다. RDBMS를 사용한다면 더 유연한 SQL문을 사용할 수 있다.
컨테이너 관리 지속성 엔티티 빈 (Container Managed Persistence Entity Bean)	지속성 관리를 위한 코딩이 필요 없다. 지속성 저장이 유연하다. 객체 관련성에 제한적 이다. 컨테이너의 자동생성으로 SQL문이 제한 적이다. 성능상의 문제가 발생할 수 있다.

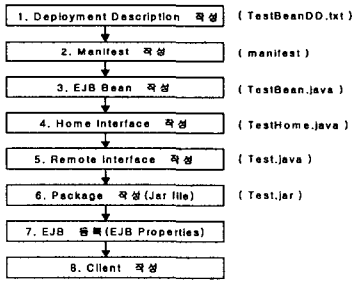
[표 1] 빈의 성격에 따른 차이점

사용될 컴포넌트의 종류 결정 이후에는 설계에 들어간다. 컴포넌트는 빈, 홈 인터페이스(Home Interface)와 원격 인터페이스(Remote Interface)를 작성한다. 또한 컴포넌트를 등록하기 위해서는 위치 지정자(Deployment Descriptor)를 작성하고, 캡슐화를 위하여 패키지 작업을 해야 한다. [그림 2]는 개발할 빈과 인터페이스이다.



[그림 2] 컴포넌트의 구조

서버의 JNDI 기능을 이용하기 위하여 생성되는 각 클래스들의 이름은 규칙이 있다. 리모트 인터페이스는 주어진 컴포넌트의 이름을 사용하고, 홈 인터페이스와 빈의 경우는 'Home'과 'Bean'이라는 이름을 덧붙여 사용한다. 컴포넌트의 개발 절차와 네이밍을 표현하면 [그림 3]과 같다.



[그림 3] 컴포넌트 개발 절차

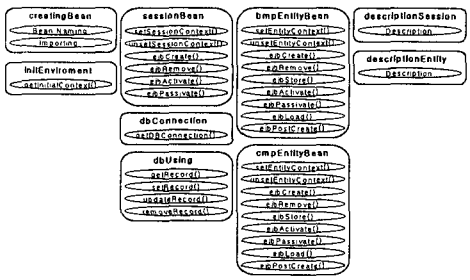
3. 정보저장소 설계

3.1 EJB 컴포넌트 모듈 추출

컴포넌트는 명세에 따라 요구되는 메소드들의 일정한 모듈이 형성되고 있다. 이러한 모듈에 따른 컴포넌트 개발은 재사용이 가능하고, 개발 시간과 비용을 절감할 수 있다. 또한 개발 이후의 유지 보수가 용이하고, 커스터마이징이 쉽다. 요구되는 메소드들은 컴포넌트 사용에 필요한 요소들이다. 컴포넌트 생성과 제거, 활성화와 비활성화, 세션 내용의 보관 유무 등의 역할을 한다.

컴포넌트 개발시에 필요한 명세를 분석하여 모듈

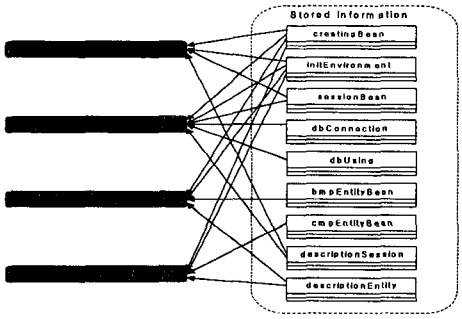
을 추출한다. 모듈은 빈 생성 클래스, 세션 빈과 엔티티 빈 클래스, 초기화 클래스, 데이터베이스 연결 클래스, 데이터베이스 사용 클래스, 위치지정 클래스를 추출한다. [그림 4]는 추출된 모듈을 나열한 것이다.



[그림 4] EJB 컴포넌트의 모듈 추출

3.2 정보저장소 설계

정보저장소는 추출한 모듈의 클래스로 구성된다. 추출된 모듈의 클래스는 생성하고자 하는 빈에 따라 적용되는 정보가 달라진다. 정보저장소는 빈 생성과 초기화에 관여하는 creatingBean, initEnvironment 클래스와 빈의 종류에 따른 sessionBean, bmpEntity-Bean, cmpEntityBean클래스, 위치 지정에 관여하는 desccriptionSession, descriptionEntity 클래스, 상태 세션 빈의 데이터베이스 사용에 관여하는 db-Connection, dbUsing 클래스로서 9개로 구성된다. 정보저장소를 통한 컴포넌트 생성은 [그림 5]와 같다.



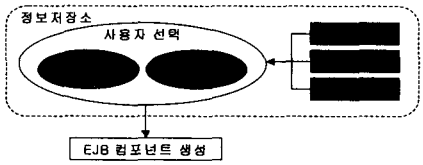
[그림 5] 추출된 모듈에 따른 정보저장소

정보저장소의 모듈 내부에는 사용자 정보를 사용하는 메소드가 존재한다. 메소드는 사용자 입력 정보에 따라 패키지와 네이밍, 데이터베이스 관련 부분이 처리되고, 기본적으로 명세에 따른 모듈 부분이 처리된다.

4. 시스템 구현 및 설계

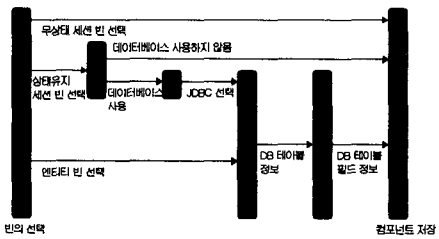
4.1 시스템 설계

컴포넌트 생성 도구 시스템은 사용자의 정보 입력에 따라 정보저장소내의 모듈을 사용하도록 한다. 본 시스템은 컴포넌트 생성에 필요한 기초 부분에 빈 생성, 초기화, 위치지정이 있고, 사용자 설정에 따른 빈의 종류와 데이터베이스 처리부분으로 구분되어 각 정보에 대한 재사용을 가능하게 하고, 코드 최적화를 야기할 수 있도록 [그림 6]과 같이 구성하였다.



[그림 6] 컴포넌트 생성 도구
시스템 구성도

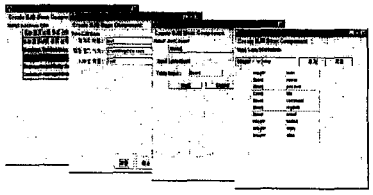
본 시스템은 사용자 입력 처리 부분에 따라 구성이 달라진다. 사용자에게 요구되는 입력부분은 [그림 7]과 같다.



[그림 7] 컴포넌트생성 순차다이어그램

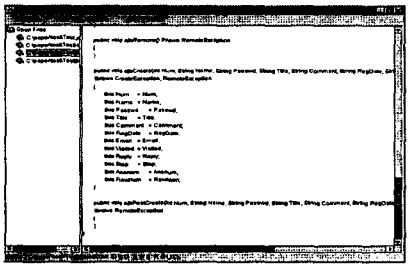
4.2 시스템 구현

본 시스템은 생성하고자 하는 빈의 선택에 따라 [그림 8]과 같이 사용자 정보 입력을 요구한다. 입력된 사용자 정보는 정보저장소에 저장되어 컴포넌트 생성시에 이를 이용한다.



[그림 8] 사용자 선택 다이얼로그

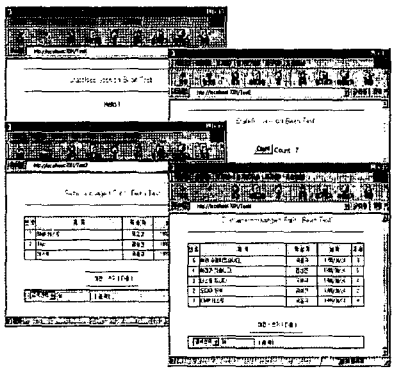
컴포넌트 생성 이후, 사용자 모듈 추가와 클라이언트를 개발해야 하는 과정의 편의를 도모하기 위하여 에디터 기능을 추가시켰다. [그림 9]는 본 시스템의 메인 화면으로서 컴포넌트 생성 이후의 비즈니스 로직 부분에 사용자 모듈을 추가하기 위한 화면이다.



[그림 9] EJB 컴포넌트 생성 도구
메인화면

본 시스템에 의해 생성된 컴포넌트는 컴파일 후에 애플리케이션 서버에 등록을 하고, 클라이언트 부분을 작성한다. 클라이언트는 애플릿, 애플리케이션, JSP, 서블릿 등으로 작성을 하고, 작성된 클라이언트에 대한 정보를 서버에 등록한다.

클라이언트 부분은 서블릿으로 작성하고, 서버는 웹로직 4.5, 데이터베이스는 MS-sql을 사용한 간단한 응용 사례는 [그림 10]과 같다. 무상태 세션 빈과 상태유지 세션 빈, 빈 관리 지속성 엔티티 빈, 컨테이너 관리 지속성 엔티티 빈의 순으로 나열된다.



[그림 10] 컴포넌트 응용 사례

5. 결론 및 향후과제

본 논문에서는 EJB 컴포넌트를 분석하여 모듈을 추출하고, 모듈에 대한 정보저장소를 설계하여 이를 이용한 컴포넌트 생성 도구를 기술하였다. 객체지향 언어에서의 컴포넌트는 일정한 모듈을 통하여 재사용성과 확장성, 개발 시간과 비용의 절감을 야기한다. 이에 본 논문에서 EJB 컴포넌트의 재사용성과 확장성을 고려하여 사용 목적에 따른 세션 빈과 엔티티 빈의 모듈, 개발 환경에 따른 데이터베이스 부분의 모듈을 추출하고 모듈에 따른 정보저장소를 설계하였으며, 사용자 정보 입력에 따라 정보저장소에 사용자 정보를 저장하고 이를 이용한 컴포넌트 생성 도구를 구현하였다.

또한 이 도구를 이용하여 생성된 컴포넌트와 클라이언트 부분의 서블릿을 이용한 예를 보였다. 그러나 비즈니스 로직에 추가될 사용자 모듈과 생성된 컴포넌트를 이용한 클라이언트 부분의 처리를 통한 완전한 자동생성 도구의 구현이 필요하다.

참고문헌

[1] http://www.javasoft.com/ejb, "EJB 1.1 specification"

[2] Ed Roman, "Mastering Enterprise JavaBeans and the Java 2 Platform, Enterprise Edition", *Wiley*, 1999

[3] Craig Larman, "Enterprise JavaBeans 201: The Aggregate Entity Pattern", April, 2000

[4] Anne Thomas, Patricia Seybold Group , "Enterprise JavaBeans Server Model Component Model for Java", December, 1997

[5] Ed Roman, "EJB Design Strategies and Performance Optimization", TS-678: *Sun's 2000 Worldwide Java Developer Conference*, 2000

[6] Mark Hapner, Dale Green, "An Overview of Java Components for Middle-Tier Servers", July, 2000

[7] Philip Mougin, "EJBs from a Critical Perspective", Decempber, December, 1999

[8] 김수동, "EJB 기반의 컴포넌트 프로그래밍", 한국정보처리학회지, 제7권 제4호, 2000년 7월, p40-45