

정보구조 모델링의 객체를 이용한 EJB의 매핑

문혜경⁰ 고병선 박재년
컴퓨터학과 정보과학부 숙명여자대학교
(mhkyung⁰, kobs, jnpark)⁰@sookmyung.ac.kr

EJB Mapping using Object of the Information Structure Modeling

Hye-Kyoung Moon⁰ Byung-Sun Ko Jai-Nyun Park
Sookmyung Women's University

요 약

컴포넌트 시장의 확산으로 소프트웨어 개발 형태가 컴포넌트 기반의 소프트웨어로 새롭게 변형되었다. COM, CORBA, EJB등은 컴포넌트 아키텍처들이다. 그 중에서 선의 EJB를 이용하여 만들어진 애플리케이션은 확장성 있고, 트랜잭션을 보장하며 멀티유저 환경에서도 안전하다. EJB 설계의 기본 모델링 언어인 UML을 이용한 EJB 추출방법은 UML 자체가 객체의 식별을 위한 자세한 가이드 라인을 제시하지 않으므로 EJB 추출에 어려움이 있고, 직관적인 경험에 의한 추출방법은 EJB 기반의 프로젝트를 처음으로 개발하고자 하는 개발자에게는 어려움이 있다. 본 논문에서는 정보구조 모델링의 방법[1]을 이용하여 구체적인 객체 추출방법을 제시하며 추출된 객체에서 쉽게 EJB의 빈들로 매핑시킬 수 있다.

1. 서 론

컴포넌트 기반 개발방법론은 객체지향 개발방법론을 기본으로 하여 등장하였다. 컴포넌트 개발방법론에 따른 구현 기술 중 하나인 EJB(Enterprise Java Bean)는 객체지향 분산 엔터프라이즈 어플리케이션 개발을 위한 컴포넌트 아키텍처이다[2][3]. EJB는 서버의 컴포넌트 개발 개념을 결합하여 비즈니스 프로그램 개발 방법을 좀 더 쉽고 단순화시키면서 간단한 노력만으로 분산 처리가 가능하고, 보안성이 있으며, 트랜잭션이 가능한 자바 클래스를 만들 수 있는 장점이 있다. 하지만 실제로 EJB를 적용하여 EJB의 특징에 맞는 설계와 구현을 하기가 쉽지 않다. UML을 이용한 EJB 추출방법은 식별을 위해서는 어떤 가이드 라인도 제시하지 않고 있다. 그러므로, 잘못된 객체 식별을 가져 올 수 있고, 이것은 또한 EJB의 잘못된 추출을 가져 올 수도 있다. 또한 경험에 의한 EJB 추출은 개발자들의 특성에 따라서 많은 변화가 있을 수 있고, 처음 개발하는 개발자에게는 정확한 지침이 없으므로 어려움이 있다. 따라서 정보구조 모델링 방법[1]을 이용하여 객체 식별의 정확한 지침을 제공하고 그 추출된 객체를 기반으로 EJB 빈으로의 매핑방법을 제시한다.

2. EJB의 개요

컴포넌트 기반 프로그램이 활성화되기 위해서는 컴포넌트가 운용되는 유연하고 확장성 높은 기반 구조가 필요하다. 그 기반 구조로서 선 마이크로시스템즈사에서 제안한 구조가 엔터프라이즈 자바빈즈(Enterprise JavaBean)이고, 자바에서는 개발환경이 J2EE이다. EJB는 J2EE안에서 분산 아키텍처에 기반한 서버 컴포넌트 모델로 제시한 중요한 스펙이다[4].

2.1 EJB의 구조

EJB의 구조에는 EJB Server, EJB Container, EJB 홈 객체, EJB 객체, 엔터프라이즈 빈이 있다. EJB 서버는 EJB 컨테이너를 하나 혹은 그 이상을 포함하는 형태로, 클라이언트의 요청을 처리하는 역할을 담당하고 있다. EJB 컨테이너는 EJB 컴포넌트의 실행환경을 제공하는 것으로, 세션 빈과 엔티 빈의 실행환경을 제공, 트랜잭션 처리를 제공, 엔터프라이즈 빈에 대한 보안 관리를 제공, 엔터프라이즈 빈의 풀링(Pooling) 기능을 제공, 컨테이너간의 호환성을 제공하며 엔터프라이즈 빈에 대한 커스터마이징이 가능하다. EJB 홈 객체(Home Interface)는 클라이언트에서 엔터프라이즈 빈 클래스를 직접 접근하는 것이 아니라, 홈 인터페이스를 통해서 빈을 생성, 삭제, 관리하며 이를 통해서 원하는 엔터프라이즈 빈 클래스를 찾을 수 있다. EJB 객체(Remote Interface)는 클라이언트가 호출할 수 있는 비즈니스 메소드를 선언하기 위한 인터페이스이다. 빈에 포함된 메소드들은 결국 클라이언트에 의해 직접 호출될 수 없다. 엔터프라이즈 빈(Enterprise Bean)은 핵심 클래스로서 실질적인 일을 수행하는 부분으로 비즈니스 로직을 메소드로 구현한다. 반드시 원격 인터페이스에 정의된 메소드들과 똑같은 이름의 비즈니스 메소드들을 가져야 하며, EJB 컨테이너 규약 메소드들도 가져야 한다. 엔터프라이즈 빈은 크게 엔티 빈(Entity Bean)과 세션 빈(Session Bean)으로 나눌 수 있다[2,3,4].

2.2 엔티 빈(Entity Bean)

엔티 빈은 데이터베이스와 같은 영속적인 저장 메커니즘에 저장되어 있는 비즈니스 객체를 나타낸다. 엔티 빈에 대한 변경은 데이터베이스에 대한 변경으로 귀착되며, 엔티 빈은 여러 클라이언트로부터 공유 액세스를 허가하며, 모든 엔티 빈은 빈을 찾고 로드하며 제거하기 위해 사용되는 프라이머리 키 객체(Primary Key Object)를 갖는다. 엔티 빈에는 두 중

류가 있다. CMP(Container-managed persistence) 엔티티 빈은 EJB 컨테이너가 자동으로 영속성을 관리하므로 개발자는 비즈니스 로직에만 집중하여 개발하면 된다. 단지 빈을 배치시 엔티티 빈의 어느 필드가 컨테이너에 의해 관리되고, 그것들이 어떻게 데이터베이스에 대응되는지를 지정해준다. BMP(Bean-managed persistence)은 CMP 엔티티 빈에서 컨테이너가 해주던 작업들을 개발자가 직접 빈 클래스 내에 명시적으로 퍼시스턴스 로직을 작성해야 하기 때문에 컨테이너 관리 퍼시스턴스보다 복잡하며, 빈을 특정 데이터베이스 타입이나 구조에 종속되도록 한다는 점이다.

2.3 세션 빈(Session Bean)

세션 빈은 비즈니스 프로세스를 위한 로직을 포함하고 있는 재사용 가능한 컴포넌트이다. 세션 빈은 자료의 지속성이 일시적인 성격을 가지며, 데이터베이스 내의 공유된 데이터에 대해서 데이터를 읽고, 변경하고, 추가하는 등의 역할을 한다. 세션 빈은 두 가지의 형태가 있다. 무상태(Stateless) 세션 빈은 한번의 메소드 호출이 일어난 후, 다음 번 호출까지 상태를 유지하지 않는다. 개발하기가 쉽고, 매우 효율적이며, 영속성을 갖지도 않고, 서버의 자원도 많이 요구하지 않는다. 상태유지(Stateful) 세션 빈은 클라이언트와 엔터프라이즈 빈 사이에 연속적으로 상호작용이 있는 경우에 사용되며 빈 인스턴스가 살아있는 동안 오직 하나의 클라이언트에 할당되며, 클라이언트의 대리인 역할을 한다[5].

3. 정보구조 모델링 개요

3.1 정보구조 모델링

모델링 방식은 사건/응답중심(Event/Response-Driven) 방식으로 사용자와 시스템의 관계는 사용자가 시스템에 요구를 하고 시스템은 그에 대한 응답을 시스템내의 정보를 이용하여 처리하는 것을 기본 개념으로 한다. 정보구조 모델링에서의 도구로는 다음과 같은 것이 있다[1].

- 배경도(Context Diagram)

배경도는 개발 범위를 보여주는 것으로, 분석가가 사용자와 충분한 대화를 통해 개발할 시스템의 범위를 결정하고 그에 따라 관련자를 결정한다. 여기서 결정된 관련자에 따라 인터페이스 객체의 수가 결정되며 메소드 추출의 관점이 결정되는 것이다. 각 관련자가 시스템에 행할 수 있는 대표적인 이벤트와 시스템의 응답을 추출하여 배경도를 작성한다. 시스템 내에서 사용되는 자료의 종류를 암시적으로 표현하고 이벤트를 개괄적으로 표시해 주는 역할을 한다. 시스템은 타원, 관련자는 사각형, 사건과 응답은 화살표로 표현한다. 관련자는 시스템에 접근이 가능하여 어떤 영향을 줄 수 있는 사람, 조직, 시스템으로 정한다.

- 이벤트도(Event Diagram)

사용자가 시스템에 행할 수 있는 이벤트를 WOD(Wanier-Orr Diagram)의 형태로 표현한다. WOD는 행위와 처리를 서술식으로 일목요연하게 볼 수 있는 장점이 있고,

형식은 시작, 처리, 끝으로 구성된다. 이는 해당 사용자가 시스템에서 사용하는 이벤트들의 생명주기를 기술하는 것이며, 배경도에 나타난 관련자들과 시스템 간의 관계를 상세히 표현하는 것이다. 이벤트도는 시스템 사용자에게 대한 역할을 보여주는 도구이다.

- 정보구조도(Information Structure Diagram)

사용자와 사용자의 이벤트를 해결하기 위한 기본 자료들의 종류를 표현한 다이어그램으로서 관련자가 시스템에 어떤 요구를 하며, 그 요구를 만족시키기 위해서 어떤 정보들이 필요한지를 나타낸다. 이 정보들은 구현시 데이터베이스 스키마로 변환된다. 정보구조도는 사용자, 관리자, 운영자의 관점에 따라 보는 형태가 틀리다[1,6].

- 행위구조도(Behavior Structure Diagram)

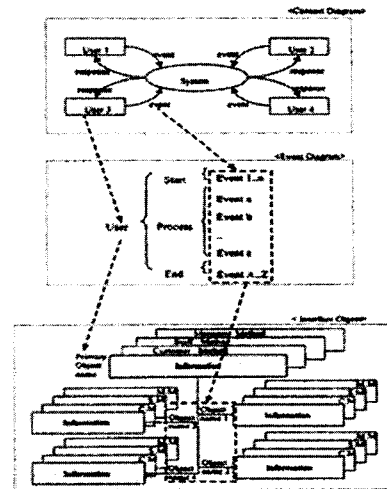
추출된 각 객체들마다 행할 수 있는 행위(Method)들을 나타내며, 행위는 사용자, 관리자, 운영자관점으로 나누어 각 관점에서의 행위를 추출한다. 따라서 각 객체마다 세 개의 행위구조도가 존재하는 것이다[1,6].

4. 정보구조 모델링을 이용한 EJB로의 매핑

시스템의 범위와 각 관련자별로 이벤트를 결정하였으므로, 이를 바탕으로 객체를 추출하며, 객체를 정제, 반복수행하고 새로운 업무가 추가될 때 추가작업을 수행한다. 객체의 추출은 분석단계의 상당부분을 차지하며 핵심작업이다. 정보구조 모델링에서의 객체는 인터페이스 객체(Interface Object), 엔티티 객체(Entity Object), 컨트롤 객체(Control Object)로 분류한다[6,7].

4.1 인터페이스 객체(Interface Object) 추출

인터페이스 객체는 시스템이 고객의 요구사항을 받아들이고 응답해 주는데 필요한 정보를 가진 객체로서, 사용자가 인터페이스에서 발생시킬 수 있는 각각의 이벤트 별로 필수 정보들을 표현한다. 이벤트도를 바탕으로 인터페이스 객체를 추출한다.



[그림 1] 인터페이스 객체의 추출

1. 사용자를 주 객체(Primary Object)로 한다.
2. 이벤트의 명사형을 인터페이스 객체명으로 한다.
3. 각 인터페이스 객체의 속성필드에 필수 정보들을 나타낸다.
4. 사용자 관점 메소드를 추출한다.
5. 관리자 관점 메소드를 추출한다.
6. 운영자 관점 메소드를 추출한다.

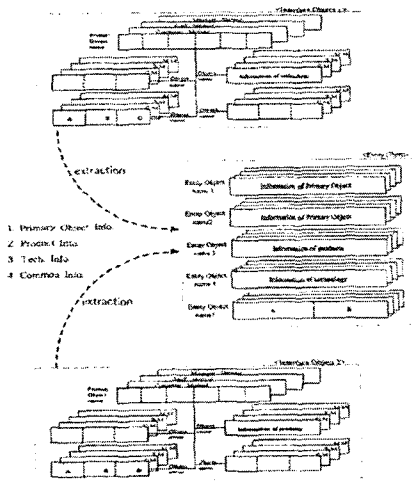
이와 같은 작업을 거쳐 인터페이스 객체가 추출되며, 1, 2, 3 단계에서는 정보 구조도, 4, 5, 6 단계에서는 행위 구조도를 얻을 수 있다. [그림 1]은 배경도와 이벤트를 이용하여 인터페이스 객체를 추출하는 단계를 나타낸 그림이다.

4.2 엔티티 객체(Entity Object) 추출

엔티티 객체는 시스템이 장기적으로 가지고 있어야 할 정보로서 시스템이 고객에게 제공해야 하는 제품이나 기술에 관한 정보가 이에 해당된다. 추출을 위해서는 인터페이스 객체에서 나타나는 정보 중에서 쉽게 변하지 않고 보존되어야 할 것과 각기 다른 관련자에 속하는 인터페이스 객체사이에 공유되어야 할 정보를 찾아내어야 한다. 따라서 엔티티 객체 추출은 반드시 인터페이스 객체를 토대로 이루어진다. 추출단계는 다음과 같다.

1. 인터페이스 객체 내의 주 객체를 엔티티 객체로 한다.
2. 제품, 기술에 관한 정보를 엔티티 객체로 한다.
3. 공용인 물건에 대한 정보를 추출한다.

[그림 2]는 엔티티 객체의 추출단계를 보여준다.



[그림 2] 엔티티 객체의 추출

4.3 컨트롤 객체(Control Object) 추출

인터페이스 객체와 엔티티 객체를 자연스럽게 조절할 수 있는 객체로서, 사용자와 저장된 자료 사이를 연결해 주는 고리 역할하며, 정보에 처리를 가하는 역할을 담당하는 객체이다.

1. 비즈니스 규칙과 방침을 추출한다.
2. 엔티티 객체의 조합, 가공으로 이루어진 자료를 추출한다.
3. 컨트롤 객체로의 확정 유무를 결정한다.

4. 사용자 관점 메소드를 추출한다.
5. 관리자 관점 메소드를 추출한다.
6. 경영자 관점 메소드를 추출한다.

위와 같은 방법으로 정보 구조도 모델링의 배경도, 이벤트도를 이용하여 인터페이스 객체, 엔티티 객체, 컨트롤 객체를 추출하게 된다. 정보구조 모델링을 이용하여 추출된 엔티티 객체는 시스템이 장기적으로 가지고 있어야 할 정보로 EJB의 엔티티 빈과 유사성을 보인다. 또한, 추출된 인터페이스 객체의 각 관점별 행위 구조도에서는 구현해야 할 메소드와 그 메소드에서 어떤 처리를 해야 하는지를 보여준다. 그리고 컨트롤 객체는 규칙이나 정책등을 나타내므로 인터페이스 객체와 엔티티 객체를 자연스럽게 조절할 수 있는 객체이다. 추출된 객체를 기반으로 엔티티 객체는 EJB의 엔티티 빈으로 매핑이 될 수 있고, 인터페이스 객체는 EJB의 세션 빈 메소드와 매핑이 되며, 또한 추출된 컨트롤 객체는 빈에서 메소드를 구현하는데 있어서 필요한 규칙이나 정책이므로 이를 반영한 비즈니스 로직을 구현함으로써 정보구조 모델링을 이용하여 EJB의 빈들로 쉽게 매핑시켜 구현할 수 있다.

5. 결론 및 향후 연구과제

본 논문에서 제시한 정보 구조 모델링을 이용하여 추출된 객체를 기반으로 EJB의 엔티티 빈과 세션 빈으로 매핑하여 구현할 수 있다. 이 방법을 이용하여 개발하면 처음 EJB 개발을 하는 개발자들도 각 단계를 따라가면서 쉽게 필요한 정보와 객체를 추출할 수 있고 엔티티 빈이나 세션 빈의 정보를 쉽게 얻을 수 있을 것이다. 향후 연구로는 정보 구조 모델링 방법을 이용하여 더 많은 프로젝트에 적용시켜 보는 것이다.

6. 참고문헌

- [1] 박재년, "정보 구조 모델링에 의한 시스템 분석", 숙명여자대학교 논문집, 제33집, 1992년 12월
- [2] 유재우, 최재영, 최종명, 박준서 저, "프로그래머를 위한 EJB", 프리랙, 2001
- [3] Sun Microsystem, "Enterprise JavaBeans Programming SL-351 Student Guide", 2000
- [4] 박지훈의 5명 저, "Enterprise JavaBeans", 대청, 2001
- [5] R. Monson, "Enterprise JavaBean", O'Rally, 2000
- [6] 정정인, "비즈니스 업무에서의 정보 구조 모델링을 이용한 객체 추출방법에 관한 연구", 숙명여자대학교 석사학위 논문, 2000
- [7] 김현희, "정보구조모델에서의 객체 추출과 수정 시 영향 범위 분석", 숙명여자대학교 석사학위 논문, 1999