

EJB 서버 시스템의 영속성 관리 방법

정승욱⁰, 노명찬, 변영철, 함호상
한국전자통신연구원 전자거래연구부
(swjung⁰,mcroh,ycb,hsham)⁰@etri.re.kr

The Persistence Management Architecture for EJB Server System

Seung-Woog Jung⁰, Myung-Chan Roh, Yung-Cheol Byun, Ho-Sang Ham
Electronic Commerce Department
Electronics and Telecommunications Research Institute

요 약

EJB(Enterprise Java Beans)는 웹 응용 서버 스펙인 J2EE(Java2 Enterprise Edition)의 핵심으로서, 비즈니스 로직을 컴포넌트 형태로 작성하여 재 사용성을 높이기 위한 서버 측 컴포넌트 프로그래밍 모델이다. EJB는 컴포넌트를 특성에 따라 일반적인 비즈니스 로직을 나타내는 세션 빈(Session Bean), 데이터베이스에 저장된 정보와 같은 여러 클라이언트에 의해 공유되며 영속 장치에 저장되는 엔터티 빈(Entity Bean) 그리고 JMS 메시지를 처리하는 메시지 드리븐 빈(Message-driven Bean)으로 구분한다. 본 논문에서는 ETRI에서 개발한 E504 EJB 서버에서 엔터티 빈의 데이터베이스 접근을 자동화 해주는 영속성 관리자(Persistence Manager)에 대해 논의한다.

1. 서 론

J2EE는 SUN사가 자바를 기업형 환경에 적용할 수 있도록 만든 기업형 자바 미들웨어 프레임워크이며, 멀티 티어(multi-tier) 상에 위치 가능한 엔터프라이즈 어플리케이션을 개발하기 위한 웹 응용 서버 표준을 일컫는다. J2EE는 컴포넌트 기반 기업형 비즈니스 응용 개발을 위해 필요한 기본 서비스를 제공 함으로서, 개발자에게 비즈니스 로직 만을 개발하도록 하고 기타 복잡한 과정은 자동적으로 처리해주는 환경을 제공한다.

엔터프라이즈 어플리케이션(Enterprise Application)은 웹 응용 서버로 배포되며, JSP/Servlet으로 구성된 웹 컴포넌트와 EJB 빈으로 구성된 비즈니스 컴포넌트로 이루어진다. JSP/Servlet은 웹 환경에서 사용자에게 동적 콘텐츠(Dynamic Contents)를 제공하며, EJB 빈은 비즈니스 로직을 담당하는 분산 객체 컴포넌트이다.

EJB(Enterprise Java Beans)는 웹 응용 서버 스펙인 J2EE(Java2 Enterprise Edition)의 핵심으로서, 비즈니스 로직을 컴포넌트 형태로 작성하여 재 사용성을 높이기 위한 서버 측 컴포넌트 프로그래밍 모델이다. EJB는 컴포넌트를 특성에 따라 일반적인 비즈니스 로직을 나타내는 세션 빈(Session Bean), 데이터베이스에 저장된 정보와 같은 여러 클라이언트에 의해 공유되며 영속 장치에 저장되는 엔터티 빈(Entity Bean) 그리고 JMS 메시지를 처리하는 메시지 드리븐 빈(Message-driven Bean)으로 구분한다.

엔터티 빈은 데이터베이스 접속 부분을 빈 사용자가 구현해야 하는 빈 관리 엔터티 빈(Bean Managed Entity Bean)과 컨테이너가 자동으로 처리해 주는 컨테이너 관리 엔터티 빈(Container Managed Entity Bean)으로 구분된다. 영속성 관리자는 컨테이너 관리 엔터티 빈을 개발할 때 빈 개발자가 데이터베이스 접근 코드를 작성하지 않고 단지 배치 기술자

(Deployment Descriptor)에 선언만 함으로써 컨테이너가 데이터베이스와 관련된 부분을 자동화하도록 해주는 모듈이다.

본 논문에서는 엔터티 빈의 데이터베이스 접근을 자동화해주는 ETRI에서 개발한 E504 EJB 서버 내의 영속성 관리자(Persistence Manager)에 대해 논의한다.

2. 영속성 관리자 요구 사항

CMP(Container Managed Persistence) 엔터티 빈을 지원하기 위해 영속성 관리자가 제공해야 할 기능은 다음과 같다.

1) 구현 클래스 제공

빈 작성자에 의해 생성된 CMP 엔터티 빈은 추상 클래스이며, 이를 상속 받은 구현(concrete) 클래스를 생성해야 한다. 예를 들어, E504 EJB 서버에서는 ProductBean CMP엔터티 빈의 구현 클래스로 ProductBean_PMBean이라는 클래스를 생성한다. 생성된 ProductBean_PMBean 클래스 내에는 영속성 관리자를 호출하는 부분이 포함되어 있다.

2) CMP 엔터티 빈 내의 데이터베이스 접속 함수 구현

CMP 엔터티 빈 내의 데이터베이스를 접근하는 함수에 대한 구현을 제공해야 한다. 데이터베이스 접근 함수로는 데이터베이스에 빈 객체를 생성하는 함수로서 RDBMS를 사용하는 경우 테이블에 행 하나를 추가하는 기능과 대응되는 ejbCreate() 함수, 데이터베이스에 저장된 내용을 빈 객체내의 필드에 저장하는 함수로서 RDBMS를 사용하는 경우 테이블 내의 한 행을 읽어와서 객체 내의 필드에 할당하는 기능에 대응되는 ejbLoad() 함수, 빈 객체내의 상태를 데이터베이스에 저장하는 함수로서 RDBMS를 사용하는 경우 객체 내의 상태를 테이블 내의 한 행에 저장하는 기능에 대응되는 ejbStore()

함수, 저장되어 있는 빈 중 특정 조건에 맞는 빈을 검색하는 finder/selector 함수, 엔터티 빈 객체 내의 CMP 필드의 값을 읽거나 수정하는 CMP 필드 접근 함수, 엔터티 빈 객체 내의 CMR(Container Managed Relationship) 필드의 값을 읽거나 수정하는 CMR 필드 접근 함수 등이 있다.

3) EJBQL을 SQL로 변환하는 기능
finder/selector 메소드에서 사용되는 EJB QL을 SQL로 변환하는 기능

- 4) CMP/CMR 필드 관리 기능
- CMP 필드의 정보 및 상태 관리
 - 엔터티 빈 간의 릴레이션(relation) 관리

E504 EJB 서버의 영속성 관리자는 EJB 서버에게 BMP 엔터티 빈과 CMP 엔터티 빈에 대해 동일한 뷰를 제공하도록 한다. 즉, EJB 컨테이너는 BMP 엔터티 빈이든 CMP 엔터티 빈이든 상관 없이 클라이언트 요청을 해당 빈에게 전달하게 되고 해당 빈이 CMP 엔터티 빈인 경우 해당 빈 내부에서 E504 영속성 관리자의 적당한 함수를 호출하게 되어, 영속성 관리자와 컨테이너 간의 관계를 최소화 하였다.

그러나, 영속성관리자와 컨테이너 간의 관계는 완전히 독립적일 수 없으며 서로 간의 직접적인 호출이 필요하다. 이를 위해 다음과 같은 2개의 클래스를 두어 서로 간의 통신은 다음 두 클래스를 통해 수행하도록 하였다.

- 1) E504PersistenceServiceSPI
- SPI : Service Provider Interface
 - 영속성 서비스를 수행하는 쪽에서 제공해야 할 API로서 컨테이너가 영속성 서비스를 직접적으로 호출할 필요가 있을 때 사용하는 클래스
- 2) PersistenceManagerCPI
- CPI : Container Provider Interface
 - 컨테이너가 영속성 서비스에 제공해야 할 API로서 영속성 서비스가 컨테이너의 기능을 직접적으로 사용할 필요가 있을 때 쓰이는 클래스

3. 영속성 관리자 기본 구조

엔터티 빈은 저장 장치에 저장되는 객체로서 다수의 클라이언트에 의해 공유된다. 엔터티 빈은 파일, 데이터베이스 등 어떤 영속성 장치에도 저장 가능하지만, 대부분의 경우 관계형 데이터베이스 내에 하나의 행으로 저장된다.

영속성 관리자는 엔터티 빈 개발자가 데이터베이스 접근 코드를 작성하지 않고 단지 배치 기술자(Deployment Descriptor)에 선언만 함으로써 컨테이너가 데이터베이스와 관련된 부분을 자동화하도록 해주는 모듈이다.

그림 1은 E504 EJB 서버의 영속성 관리자에 대한 개략적인 구조를 나타낸 그림이다. EJB 빈 배포를 담당하는 ContainerFactory 객체는 각 빈 타입에 따라 적당한 컨테이너를 생성하여 주어진 응용 모듈에 포함된 빈을 배포한 후, PMDeploy에게 CMP 엔터티 빈을 배포하도록 한다. PMDeploy는 배치 기술자의 정보를 기초로 CMP 엔터티 빈에 대한 메타 데이터(metadata)를 생성한다. 메타 데이터에는 CMP 엔터티 빈에 대한 정보, CMP 엔터티 빈간의 관계에 대한 정보, dependent value class에 대한 정보, CMP/CMR 필드에 대한 정보 등이 포함된다. 또한, PMDeploy는 실제 빈의

로부터의 요청을 처리할 PersistenceManager를 생성하게 된다.

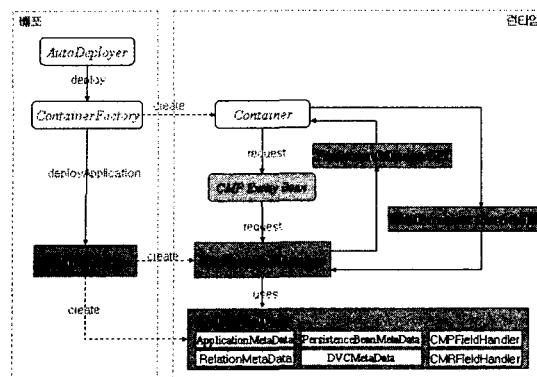


그림 1. 영속성 관리자 구조

PersistenceManager는 PMDeploy에 의해 생성되는 객체로서 데이터베이스와 관련된 모든 요청을 받아 처리하며, 엔터티 빈이 EJB 스펙 1.x를 따르면 PersistenceManager1x 객체가 생성되고 EJB 2.0 스펙을 따르면 PersistenceManager20 객체가 생성된다.

PersistenceManager는 각종 데이터베이스 관련 연산을 수행할 때 해당 빈의 CMP 메타 데이터를 참조하게 된다. 만약, PersistenceManager가 직접 Container에 대한 호출이 필요하게 되면 PersistenceManagerCPI 객체를 사용하고, 반대로 Container 객체가 PersistenceManager의 기능에 대한 직접적인 호출이 필요하게 되면 E504PersistenceManagerSPI 객체를 사용한다.

클라이언트가 특정 빈에 요청을 보내면 Container 객체가 이를 받아 보안, 트랜잭션 등의 기본 서비스를 처리하고 빈에게 메시지를 전달한다. 해당 빈이 CMP 엔터티 빈인 경우 내부적으로 영속성 관리자를 호출하여 데이터베이스 관련 연산을 수행하게 된다. Container는 해당 빈이 BMP인지 CMP인지 구별할 필요 없이 클라이언트가 요청한 빈의 메소드를 호출하면 된다.

PersistenceManager는 CMP 엔터티 빈으로 받는 요청을 처리하게 되는데, 이때 데이터베이스와 관련된 주요 메소드는 다음과 같다.

- ejbCreate()/ejbPostCreate()
: 엔터티 빈 생성과 관련된 함수
- ejbLoad()
: 엔터티 빈의 상태 정보를 데이터베이스에서 로드(load)하는 함수
- ejbStore()
: 엔터티 빈의 상태 정보를 데이터베이스에 저장(store)하는 함수
- ejbActivate()
: 엔터티 빈이 데이터베이스에서 메모리로 활성화될 때 호출되는 함수
- ejbPassivate()
: 메모리 내의 엔터티 빈이 데이터베이스로 비활성화될 때 호출되는 함수

- finder/selector 메소드
 - : 저장되어 있는 빈 중 특정 조건에 맞는 빈을 검색하는 함수로서, finder 메소드는 리모트 클라이언트에서 사용되며 selector 메소드는 빈 내부에서 사용된다.
- CMP 필드 접근 메소드 (getCMP(), setCMP())
 - : CMP 필드 접근 메소드
- CMR 필드 접근 메소드 (getCMR(), setCMR())
 - : 엔터티 빈 간의 관계를 나타내는 CMR 필드를 조회하거나 수정하는 함수

그림 2는 영속성 서비스 내의 주요 클래스들 간의 관계를 나타낸 그림이다.

StateManager 객체는 엔터티 빈의 상태 정보 저장, 조회 및 CMP 필드와 관련된 메소드를 처리한다. PersistenceManager에게 요청된 메소드 중에서 StateManager에게 전달되는 메소드는 ejbCreate(), ejbPostCreate(), ejbLoad(), ejbStore(), ejbActivate(), ejbPassivate(), CMP 필드 접근 메소드 등이다.

CMP 필드 접근 메소드 요청은 해당 호출에 대응되는 CMP 필드의 CMPFieldHandler에 전달되어 처리된다. CMP 필드는 데이터베이스 내의 한 필드와 대응되는 SVC(Single Valued Class) 필드와 데이터베이스 내의 여러 필드와 대응되는 DVC(Dependent Value Class) 필드로 구분되며, 각각의 처리를 위해 SVCCMPFieldHandler, DVCFieldHandler 객체가 사용된다.

RelationshipManager는 엔터티 빈간의 관계를 나타내는 CMR 필드와 관련된 메소드를 처리한다. RelationshipManager에게 전달되는 메소드는 CMR 필드 접근 메소드이며, 이는 해당 호출에 대응되는 CMR 필드의 CMRFieldHandler에 전달되어 처리된다.

엔터티 빈간의 관계는 one-to-one 관계, one-to-many 관계, many-to-one 관계, many-to-many 관계로 구분되며, 각각의 처리를 위해 OneToOneCMRFieldHandler, OneToManyCMRFieldHandler, ManyToOneCMRFieldHandler, ManyToManyCMRFieldHandler 객체가 사용된다.

finder나 selector 메소드는 PersistenceStore에 직접 전달되어 처리된다. finder나 selector 메소드 처리 전에 영속성 관리자는 메모리 내에 캐시되어 있는 해당 빈을 데이터베이스에 저장해야 한다.

PersistenceManager는 CMP 엔터티 빈의 CMP/CMR 영속성 필드의 상태를 담고 있는 PMState 클래스를 생성하고 관리한다. StateManager는 CMP 엔터티 빈의 PMState 상태를 변경한다. 즉, ejbCreate() 메소드가 호출되면 PMState의 상태를 PMState.CREATED로 ejbRemove() 메소드가 호출되면 PMState.REMOVED로 setCMP() 메소드가 호출되면 PMState.DIRTY로 변경한다. ejbStore()가 호출되었을 때 PMState의 상태를 보고 CREATED이면 PersistenceStore의 insertObject() 메소드를 통해 데이터베이스에 한 행을 삽입하게 되고, REMOVED 상태이면 deleteObject() 메소드를 통해 데이터베이스에서 해당 행을 지우게 되며, DIRTY 상태가 되면 빈의 내용을 데이터베이스에 기록하게 된다.

엔터티 빈이 저장되는 형태는 파일, 데이터베이스 등 다양

하게 존재할 수 있으며, 각 형태에 따라 PersistenceStore 객체를 구현해야 한다. 만약 JDBC를 이용하여 엔터티 빈을 저장할 경우 JDBCPersistenceStore를 구현해야 하고, 파일에 저장할 경우 FilePersistenceStore를 구현해야 한다.

현재 E504 EJB 서버에서 제공하고 있는 PersistenceStore 객체는 JDBC를 이용하여 엔터티 빈을 저장하는 JDBCPersistenceStore 객체이다. E504 EJB 서버 영속성 관리자에서 지원하는 모듈은 이외에도 EJBQL을 파싱하여 SQL로 변환하는 모듈, 엔터티 빈과 관계형 데이터베이스 간의 매핑을 제공하는 모듈이 포함되어 있다.

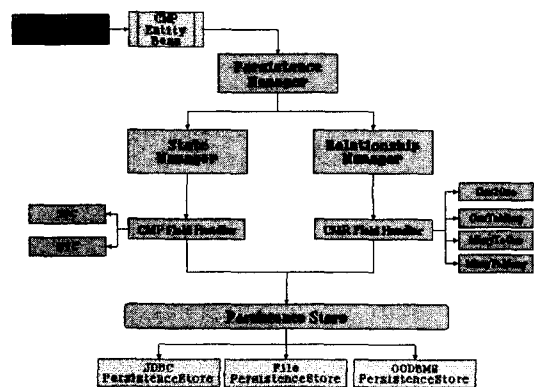


그림 2. 영속성 관리자 클래스 구조

3. 결론

지금까지 E504 EJB 서버내의 영속성 관리자에 대한 기능 및 간략한 구조에 대해 논의하였다.

엔터티 빈에 대한 접근은 저장 장치와의 접속을 필요로 하므로 다른 빈에 비해 상당한 비용이 소요된다. 속도 향상을 위해 E504 EJB 영속성 관리자는 트랜잭션 기반 엔터티 빈 캐시 기능을 기본적으로 제공하고 있다. 향후, 영속성 관리자는 다양한 캐시 기능 및 속도 향상 기능을 제공해야 한다.

[참고 문헌]

1. Java™ 2 Platform Enterprise Edition Specification, v1.4 Public Draft, SUN microsystems, July 15, 2002.
2. Enterprise Java Beans™ Specification Version 2.0 Final Release, SUN microsystems, August 22, 2001
3. Java Data Objects JSR 000012 Version 1.0 ProposedFinalDraft, SUN Microsystems, May 7, 2001