

# EJB 3.1 경량 컨테이너 구조 환경에서 영화 예매 시스템의 설계 및 구현

이명호\*  
세명대학교, 전자상거래학과

## Design and Implementation of Movie Reservation System with EJB 3.1 Lightweight Container Architecture

Myeong-Ho Lee\*  
Dept. of eCommerce, Semyung University

**요약** 본 논문은 경량 컨테이너 구조 환경에서 EJB 3.1 기반의 영화 예매 시스템의 설계 및 구현을 하는데 목적이 있다. N-tier 엔터프라이즈 아키텍처가 가지고 있는 문제점을 해결하고 장점들을 지원하기 위하여 새롭게 등장한 아키텍처가 경량 컨테이너 아키텍처이다. 이 구조는 EJB 아키텍처와 같이 무겁지 않으면서도 EJB 컨테이너의 모든 기능을 제공하는 구조이다. 현재까지 경량 컨테이너 아키텍처로 표준 엔터프라이즈 기반의 EJB 3.1이 있다. 따라서 본 연구에서는 N-tier 엔터프라이즈 아키텍처가 가지고 있는 단점들을 해결하고 장점들을 지원하기 위하여 개발된 최신 경량 컨테이너 아키텍처인 새로운 표준 환경인 EJB 3.1을 기반으로 영화 예매 시스템의 분석 및 설계한 후 구현하고자 한다.

**주제어** : EJB 3.1, 경량 컨테이너 아키텍처, Non EJB, EJB

**Abstract** This paper aims to design and realize movie reservation system with EJB 3.1 lightweight container architecture. N-Tier enterprise architecture to resolve the problem with benefits to support the new architecture is a lightweight container architecture. This architecture, such as the EJB, but not heavy, to provide all of the architecture is possible. The lightweight container architecture is most often used in business EJB 3.1 is well-known architecture. Therefore, this research has the N-tier enterprise architecture to solve the advantages and disadvantages developed to support the latest EJB 3.1 architecture based on the design and implementation of a movie reservation system.

**Key Words** : EJB 3.1, Lightweight Container Architecture, N-Tier Enterprise Architecture

## 1. 서론

차세대 인터넷과 웹이 발전하면서 현재의 인터넷에서

리소스에 대한 정보와 자원 사이의 관계를 온톨로지(Ontology) 형태로 표현해 기계가 처리하도록 하는 프레임워크인 동시에 이를 지원하기 위한 시맨틱 웹 기술과

Received 15 April 2013, Revised 10 May 2013

Accepted 20 May 2013

Corresponding Author: Myeong-Ho Lee(Semyung University)

Email: mhlee@semyung.ac.kr

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ISSN: 1738-1916

온톨로지를 활용해 서비스를 기술하고 온톨로지의 의미적 상호 운용성을 이용하여 서비스를 검색하고 그 결과를 토대로 조합 및 중재 기능을 자동화하는 시맨틱 웹 서비스로의 웹 3.0을 위한 핵심기술로 진화되고 있다[2]. 또한 급속도로 증가하고 있는 SNS, 웹로그, 소셜미디어, 이메일, 이미지, 동영상 등의 비정형 데이터 중심의 빅데이터 관심이 이제는 클라우드 컴퓨팅 환경의 시대를 예고하고 있다. 향후 웹 3.0 인터넷 혁명의 파동에 대한 가설을 기반으로 N-스크린, 클라우드 컴퓨팅, 빅데이터, 디지털 콘텐츠, 스마트워크, 소셜 서비스 플랫폼, 스마트 디바이스, 모바일 웹앱, HTML5 등의 키워드를 중심으로 발전될 전망이다[5]. 이러한 엔터프라이즈 애플리케이션 환경에서는 이 기종 컴퓨터들 간에 부하를 줄여 시스템의 성능 저하와 네트워크 병목 현상을 줄일 수 있는 분산 객체 구조가 필요하며 대용량 데이터 처리에 필요한 분산 객체의 성공모델로 먼저 알려진 것이 EJB(Enterprise JavaBeans)이다. EJB 1.1/ 2.0까지는 실행 속도의 저하, 개발 생산성의 저하, 품질 및 이식성의 저하 등의 단점이 발생하였다. 그러나 EJB 3.0 사양이 발표되면서 유연하고, 테스트 및 개발하기 쉬운 새로운 EJB로 탄생하게 되었으며[3,4], EJB 3.1 사양을 통하여 인터페이스 뷰가 없어졌으며, 싱글톤 빈, EJB Lite 및 임베디드 EJB가 추가되어 경량 컨테이너 구조 환경으로 확실한 변신을 하였다[8]. 현재까지 플랫폼의 변화에 따른 개발 생산성에 대한 비교 연구는 2가지의 애플리케이션에 대하여 다른 J2EE 플랫폼별에서의 개발 생산성을 비교한 연구였으며[9], 그 이후 EJB 2.0과 EJB 3.0의 소프트웨어 개발 생산성 비교 연구와 경량 컨테이너 구조 환경에서 스프링 2.5와 EJB 3.0의 개발 생산성 연구가 있었다[4,5]. 그러나 EJB 3.0/3.1 사양이 발표되었지만 플랫폼 상에서 소프트웨어 개발 생산성 비교에 대한 연구가 미비와 구체적인 평가 지표와 지침이 부족하여 현업에서는 대용량 분산 객체 처리를 위한 새로운 사양의 설계 및 구현에 제한이 있었다[7]. 따라서 본 연구에서는 대용량 분산 객체의 새로운 표준 환경인 EJB 3.1 사양을 기반으로 파일럿 시스템을 분석 및 설계한 후 구현하고자 한다.

## 2. EJB의 기본 개념

### 2.1 EJB 3.1의 고찰

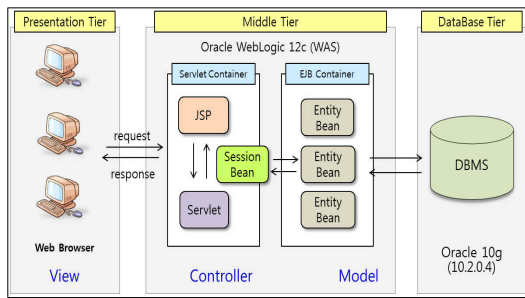
EJB는 N-tier의 분산형 객체지향 자바 애플리케이션

을 개발하고 보급하기 위한 컴포넌트 구조로서 1998년에 SUN사에서 EJB 1.0 사양을 만들면서 J2EE 플랫폼의 핵심 컴포넌트로 자리 잡았다. 1999년에 EJB 1.1 사양의 발표되었고, 2001년에 EJB 2.0 사양이 출시되면서 인터넷 환경의 확산 및 자바 언어의 대중화에 따라 컴포넌트 개발을 위한 표준 플랫폼으로 자리 잡았다. 2003년 EJB 2.1을 거쳐 2006년에는 EJB 3.0 사양이 발표되면서 어떤 프레임워크에도 종속되지 않으며, 어떤 컨테이너에도 종속되지 않는 일반적인 자바 객체인 POJO(Plain Old Java Object) 기반의 코딩, 메타 데이터, 엔티티 빈의 변화를 통하여 시대의 요구사항에 맞춰 좀 더 유연하고, 테스트 및 개발하기 쉬운 새로운 EJB로 탄생하게 되었다[1,7]. 2009년에 발표된 EJB 3.1 사양에서는 인터페이스 뷰가 사라졌으며, WAR에 EJB 컴포넌트가 포함되었다. 또한 싱글톤 빈, 비동기 세션빈, 임베디드가 가능한 EJB. 그리고 EJB 라이트가 새롭게 추가되었다[8].

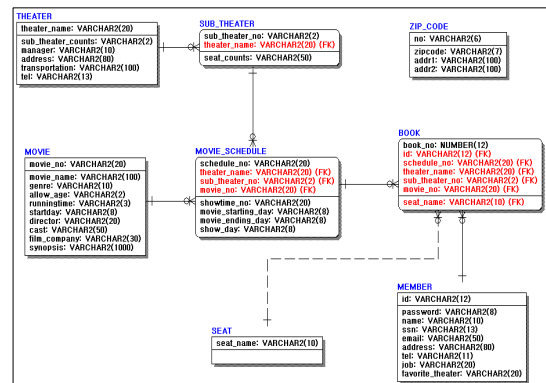
따라서 EJB는 확장성 있는 애플리케이션 서버 컴포넌트들을 지원하는 여러 서비스들을 제공함으로써 비즈니스 애플리케이션들을 컴포넌트 단위로 쉽게 작성할 수 있도록 되었다. EJB 컴포넌트 모델은 서버 컴포넌트들을 지원하기 위해 자바 빈 컴포넌트 모델로 확장하였으며, 이 서버 컴포넌트는 애플리케이션 서버에서 실행되는 애플리케이션 컴포넌트를 의미한다. 서버 컴포넌트들은 애플리케이션을 개발하기 위해 다른 컴포넌트들과 결합될 수 있다[3,4,6].

### 2.2 EJB 3.1의 구성

단독 애플리케이션 개발환경인 J2SE(Java SE)에서는 대용량 비즈니스 애플리케이션을 구현하기에는 부족한 점이 많아 트랜잭션, 보안, 메시징, 네이밍, 웹 등의 분산 애플리케이션을 작성할 수 있도록 N-tier 구조인 J2EE(Java EE) 플랫폼을 개발하게 되었다. 이 구조는 클라이언트-서버 구조에서 유연하게 대처하기 위하여 비즈니스 로직과 데이터베이스 처리 등의 개별적인 고유 애플리케이션을 서로 다른 시스템에 분산시킨 3-tier 구조에서 컴포넌트 구조를 좀 더 세분화시켜서 구현한 개념으로 이 구조부터 웹 기반 분산 객체 기반 애플리케이션이 운영되게 되었다[1,7]. 본 연구에서 경량 컨테이너 구조 환경에서 EJB 3.1 사양으로 구현한 시스템 구성도를 살펴보면 [Fig. 1]과 같다.



[Fig. 1] Architecture of EJB 3.1 System



[Fig. 2] Architecture of Database Schema

### 3. 개발 시스템의 분석 및 설계

#### 3.1 개발 환경

본 연구에서는 경량 컨테이너 구조 환경에서 EJB 3.1 기반의 영화 예매 시스템을 개발하기 위하여 <Table 1> 과 같은 개발환경을 이용하여 시스템을 분석하고 설계한 후 영화 예매 시스템을 구현하도록 한다.

<Table 1> Environment of Movie Reservation System

Items	EJB 3.1
O/S	Windows 7 Ultimate K
JAVA	Java SE 7/Java EE 6
WAS	Oracle WebLogic 12c(12.1.1)
DB	Oracle 10g(10.2.0.4)
DB Modeling	ERWin7.3
IDE TOOL	Eclipse Indigo(3.7)
CASE TOOL	IBM Rational Rose 7.0

#### 3.2 데이터베이스 스키마의 설계

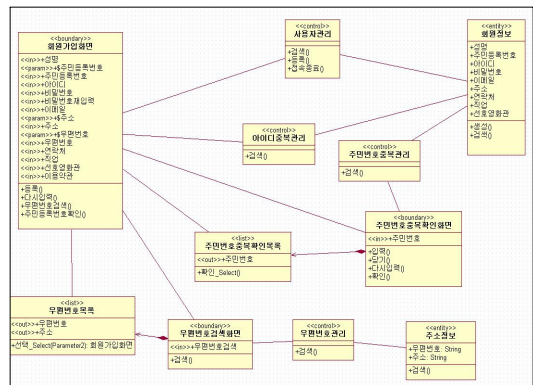
표준 엔터프라이즈 대용량 분산 객체 처리를 위한 EJB 3.1 사양의 영화 예매 시스템은 테이블로 영속적인 정보를 저장하기 위하여 데이터 모델이 먼저 정의되어야 한다. 따라서 엔티티 클래스를 기반으로 데이터 모델을 작성한다. 본 연구에서 제시한 데이터 모델인 데이터베이스 스키마 구조는 [Fig. 2]와 같이 영화관 정보, 상영관 정보, 영화 정보, 상영 스케줄 정보, 예매 정보, 회원정보, 예약정보, 좌석 정보, 우편번호 정보 등으로 설계하여 구현하도록 한다.

#### 3.3 유스케이스 명세서

영화 예매 시스템의 기본적인 요구사항을 기술한 문제 기술서를 기반으로 액터와 유스케이스를 추출한다. 추출된 유스케이스는 영화 관리, 영화관 관리, 상영관 관리, 상영스케줄 관리, 예매, 로그인, 로그아웃, 회원 가입이 된다. 그 중에서 유스케이스 별로 작성되며 각 유스케이스에 대한 자세한 정보를 기술하는 문서 산출물로 유스케이스 명세서가 있다.

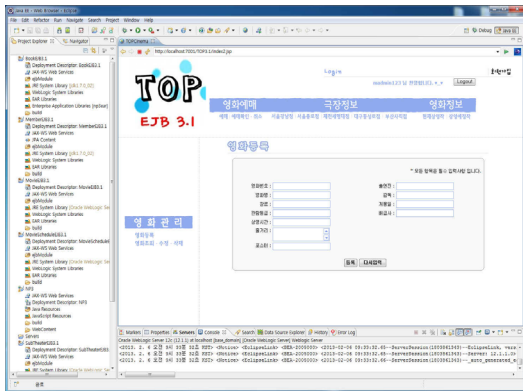
#### 3.4 클래스 다이어그램

비기능적인 요구사항과 플랫폼을 고려한 후, 설계 활동을 통한 분석 클래스를 구체화하여 설계 클래스를 도출한다. 따라서 본 연구의 영화 예매 시스템에서 중요한 회원 가입의 분석 객체 모델인 클래스 다이어그램으로 표현하면 [Fig. 3]과 같다.

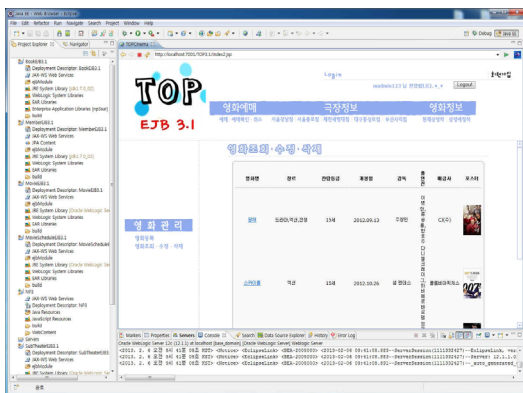


[Fig. 3] Analysis Object Model of Member





[Fig. 7] WebForm View of Movie Insert



[Fig. 8] WebForm View of Retrieve/Update/Delete

마지막으로 예매 확인 웹폼 화면은 보면 [Fig. 9]와 같다.



[Fig. 9] WebForm View of Reservation Check

## 5. 결론

현재까지 표준 엔터프라이즈 경량 컨테이너 구조 환경의 성공 모델로 알려진 EJB 3.1에서 정량적인 성과지표 개발 및 사례의 부족으로 이전 사양으로 운영 중인 실무 프로젝트의 업그레이드나 새로운 기술 사양의 적용이 미비하였다. 그 이유는 EJB 버전별 많은 기술 변화와 기본적인 EJB의 기술 스킬 습득의 시간이 길고 표준 사양의 복잡도가 높음에 따라 쉽게 새로운 사양들을 현업에 적용하지 못하였다. 또한 EJB의 소프트웨어 개발 생산성 비교에 대한 연구도 사양의 일부분 LoC 분석만 시도하고 있으며, EJB 2.0/EJB 3.0/EJB 3.1의 새로운 사양이 발표됨에도 현재까지 버전별 구체적인 분석 및 설계 기반에 따른 구현 지침이 부족하여 소프트웨어 생산성의 평가와 프로젝트의 새로운 시도에 제한이 있었다.

본 연구에서는 대용량 엔터프라이즈 구조 환경의 시스템 처리를 위하여 새로운 경량 컨테이너 구조 환경인 EJB 3.1 사양을 기반으로 파일럿 프로젝트의 분석 및 설계를 통하여 구현하였다. 따라서 새로운 표준 환경에서 사양이 발표되더라도 현업에 적용할 수 있는 사례를 제시하였다.

향후 경량 컨테이너 구조 환경에서의 EJB 3.0과 3.1을 이용한 성능 비교 평가와 스프링 3.1과의 소프트웨어 개발 생산성 비교 연구가 지속되어야 할 것이다.

## REFERENCES

- [1] B.G. Kim, Enterprise Java Beans 3.0, Game, 2006.
- [2] H.H. Kim, FOAF Technology Analysis of Semantic Web Generation, Microsoft, Vol. 9, pp. 124-129, 2012
- [3] M.H. Lee, Design and Implementation of Large Size Distributed Object Process Based EJB3.0 Standardization, Journal of Korean Institute of Plant Engineering, Vol. 13, No. 2, pp. 51-58, 2008
- [4] M.H. Lee, A Study on Comparison of Software Development Productivity with EJB 2.0 and EJB 3.0, Journal of the Society of Korea Industrial and Systems Engineering. Vol. 31, No. 3, pp. 1-7, 2008.
- [5] M.H. Lee and J.S. Han, Comparison of Development Productivity of Spring 2.5 and EJB 3.0 with

Lightweight Container Architecture, The Society of Digital Policy & Management, Vol. 10, No. 3, pp. 137-142, 2012.

- [6] H.S. Chae, Object Oriented CBD Development Bible, HanbitMedia, 2006.
- [7] D. Panda, R. Rahman, and D. Lane(2007), EJB 3 in Action, Manning Publications.
- [8] Richard M. Reese(2011), EJB 3.1 Cookbook, PACKT Publishing.
- [9] [http://java.sun.com/developer/technicalArticles/J2EE/intro\\_ee5/index.html](http://java.sun.com/developer/technicalArticles/J2EE/intro_ee5/index.html)

#### 이 명 호(Lee, Myeong Ho)



- 1984년 2월 : 아주대학교 산업공학과(공학사)
- 1986년 2월 : 아주대학교 대학원 산업공학과(공학석사)
- 2001년 2월 : 아주대학교 대학원 산업공학과(공학박사)
- 2002년 3월 ~ 현재 : 세명대학교 전자상거래학과 부교수

- 관심분야 : 물류정보시스템, WAS 프로그래밍, 모니터링 시스템
- E-Mail : mhlee@semyung.ac.kr