

EJB 3.0과 EJB 3.1의 소프트웨어 개발 생산성 비교 연구

이명호
세명대학교, 전자상거래학과

A Study on Comparison of Software Development Productivity with EJB 3.0 and EJB 3.1

Myeong-Ho Lee

Dept. of eCommerce, Semyung University

요약 본 논문은 EJB 3.0과 EJB 3.1에 연관된 객체 지향 소프트웨어 개발 생산성에 대한 정량적인 평가 지표를 제공하는데 목적이 있다. N-Tiers 엔터프라이즈 아키텍처가 가지고 있는 문제점을 해결하고 장점들을 지원하기 위하여 새롭게 등장한 아키텍처가 경량 컨테이너 아키텍처이다. 이 구조는 EJB(Enterprise JavaBeans) 아키텍처와 같이 무겁지 않으면서도 EJB 컨테이너의 모든 기능을 제공하는 구조이다. 그러나 동일한 플랫폼 상에서 EJB 3.0과 EJB 3.1에 대한 성능 평가 연구는 부족하였다. 또한 정량적인 분석도 일부분의 LoC(Line of Code) 분석만 시도함에 따라 새로운 사양이 발표됨에도 구체적인 평가 지표와 지침이 부족하여 소프트웨어 개발 생산성의 평가와 프로젝트의 새로운 시도에 제한이 있었다. 따라서 본 연구에서는 동일한 개발 플랫폼 환경을 기반으로 영화예매시스템을 EJB 3.0과 EJB 3.1에서 설계하고 구현한 후, 개발 플랫폼 환경별 객관적인 소프트웨어 개발 생산성 비교 및 표준화에 따른 평가 지침을 제공하고자 한다.

주제어 : 평가 지표, EJB 3.0, EJB 3.1, N-Tiers, 소프트웨어 개발 생산성

Abstract The purpose of this study is to provide a quantitative evaluation index of object-oriented software development productivity for EJB 3.0 and EJB 3.1. A lightweight container architecture is a newly-developed architecture in order to solve demerits that N-Tiers enterprise architecture has and to support merits therein. A lightweight container architecture is not as heavy as EJB(Enterprise JavaBeans) architecture is, whereas it provides all the functionality of the EJB container. However, there have been insufficient studies on a performance evaluation of EJB 3.0 and EJB 3.1 on an identical platform. Since a quantitative analysis has been performed only with respect to a partial LoC(Line of Code) analysis, there have been insufficient objective evaluation indexes and guidelines for a new specification released. Accordingly, there has been limited evaluation of software development productivity and on new innovations in a project. Therefore, this study designed and implemented a movie reservation system in EJB 3.0 and EJB 3.1, which was based on the same development platform environment. This study then intends to provide evaluation guidelines for an objective software development productivity cl LoC(Line and standardization thereof by development platform environment.

Key Words : Evaluation Index, EJB 3.0, EJB 3.1, N-Tiers, Software Development Productivity

*본 연구는 2013년도 세명대학교 자체연구비 지원에 의해 연구되었음.

Received 5 February 2014, Revised 6 March 2014

Accepted 20 March 2014

Corresponding Author: Myeong-Ho Lee(Semyung University)

Email: mhlee@semyung.ac.kr

ISSN: 1738-1916

© 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.

1. 서론

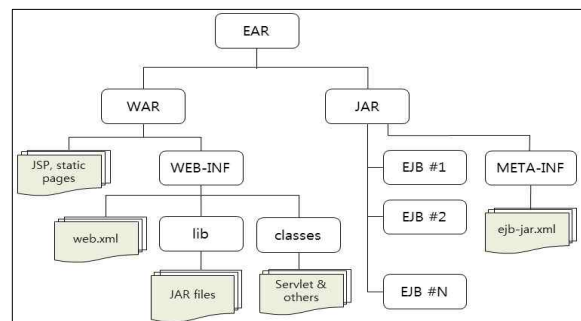
시멘틱 웹 기술은 인터넷에서 리소스에 대한 정보와 자원 사이의 관계를 온톨로지 형태로 표현한 후 기계가 처리하도록 하는 프레임워크인 동시에 이를 지원하기 위한 기술이다. 시멘틱 웹 서비스는 온톨로지를 활용해서 서비스를 기술하고, 온톨로지의 의미적 상호 운용성을 이용하여 서비스를 검색한 후 그 결과를 토대로 조합 및 중재 기능을 자동화하는 웹 3.0을 위한 핵심기술로 진화되고 있다[1][2]. 또한 급속도로 증가하고 있는 SNS, 웹로그, 소셜미디어, 이메일, 이미지, 동영상 등의 비정형 데이터 중심의 빅데이터 관심이 이제는 클라우드 컴퓨팅 환경의 시대를 예고하고 있다. 향후 웹 3.0은 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[3]. 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[4-6], EJB 3.1 사양을 통하여 인터페이스 뷰가 없어졌으며, 싱글톤 빈, EJB Lite 및 임베디드 EJB가 추가되어 경량 컨테이너 구조 환경으로 확실한 변신을 하였다[7][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[3,10-12]. 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[4]. 따라서 본 연구에서는 대용량 분산 객체의 가장 성공 모델인 EJB에 대하여 같은 플랫폼

상에서 EJB 3.0과 EJB 3.1 사양에 대하여 정량적인 평가 지표를 제시하여, 새로운 EJB 사양에 대한 정량적인 분석을 통하여 객관적인 소프트웨어 개발 생산성 연구에 대한 지침을 제공하고자 한다.

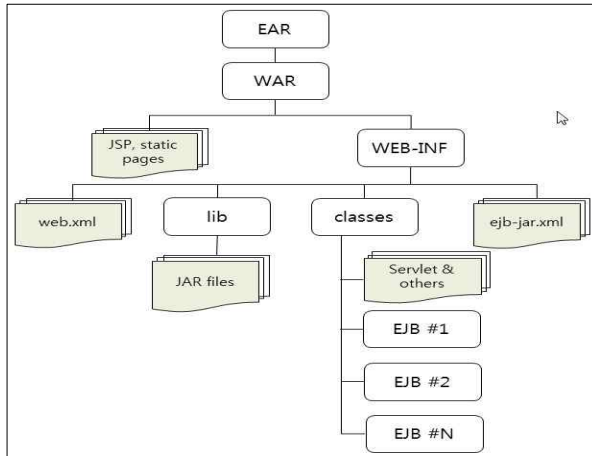
2. EJB 버전별 비교

2.1 EJB의 정의 및 특징

EJB는 N-Tiers의 분산 객체지향 자바 애플리케이션을 개발하고 보급하기 위한 컴포넌트 구조로서 1998년에 SUN사에서 EJB 1.0 사양을 만들면서 J2EE 플랫폼의 핵심 컴포넌트로 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 기반의 코딩, 메타 데이터, 엔터티 빈의 변화를 통하여 시대의 요구사항에 맞춰 좀 더 유연하고, 테스트 및 개발하기 쉬운 새로운 EJB로 탄생하게 되었다[4-6]. 2009년에 발표된 EJB 3.1 사양에서는 인터페이스 뷰가 사라졌으며, WAR에 EJB 컴포넌트가 포함되었다. 또한 싱글톤 빈, 비동기 세션빈, 임베디드 가능한 EJB, 그리고 EJB 라이트가 새롭게 추가되었다[7][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[13]. EJB 3.0과 EJB 3.1의 패키징 구조는 [Fig. 1]과 [Fig. 2]와 같다.



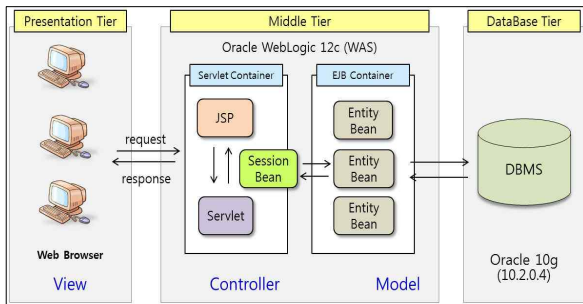
[Fig. 1] Packaging Structure of EJB 3.0



[Fig. 2] Packaging Structure of EJB 3.1

2.2 개발 환경

본 연구에서 EJB 3.0과 EJB 3.1 사양에서 개발 생산성을 비교 분석하기 위한 J2EE(Java EE)로 구현한 N-Tiers 개발 환경을 살펴보면 [Fig. 3]과 같다.



[Fig. 3] Architecture of EJB 3.0/3.1 Systems

3. 개발 생산성 비교 방안

3.1 테스트 환경

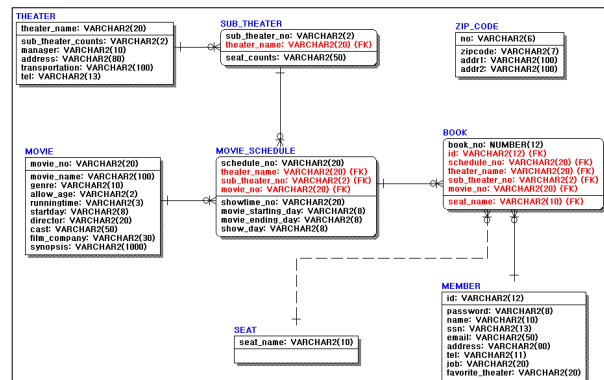
본 연구에서는 EJB 3.0과 EJB 3.1 사양을 기반으로 하는 소프트웨어 개발 생산성을 비교 분석하기 위한 방안으로 <Table 1>과 같은 동일한 개발환경과 데이터베이스 스키마를 이용하여 EJB 3.0과 EJB 3.1 환경에서의 파일럿 프로그램을 개발한 후, 이 프로그램 통하여 각 항목별 평가 지표를 이용하여 사양별 정량적으로 개발 생산성을 비교하도록 한다.

<Table 1> Development Environment of EJB Versions

Items	EJB 3.0	EJB 3.1
OS	Windows 7 Ultimate K	Windows 7 Ultimate K
Platform	Java SE 7/Java EE 6	Java SE 7/Java EE 6
WAS	Oracle WebLogic 12c (12.1.1)	Oracle WebLogic 12c (12.1.1)
DB	Oracle 10g(10.2.0.4)	Oracle 10g(10.2.0.4)
DB Modeling	ERWin7.3	ERWin7.3
IDE	Eclipse Indigo(3.7)	Eclipse Indigo(3.7)
CASE	IBM Rational Rose 7.0	IBM Rational Rose 7.0

3.2 데이터베이스 스키마 설계

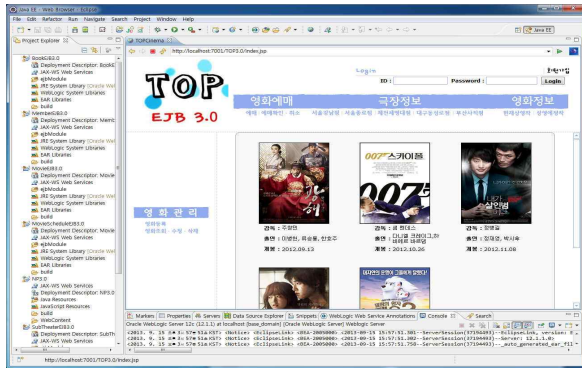
소프트웨어 생산성 비교를 위하여 개발될 파일럿 시스템의 데이터베이스 스키마는 EJB 3.0과 EJB 3.1 사양에서 [Fig. 4]와 같이 영화관리, 극장관리, 영화예매, 회원관리를 하고, 영화 관리자는 영화등록, 영화관리, 영화관등록, 영화관관리, 상영관관리, 좌석관리 등을 기반으로 동일한 데이터베이스 스키마를 이용하여 비교 분석한다.



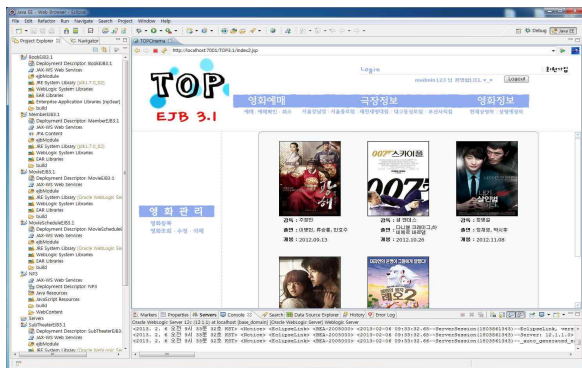
[Fig. 4] Architecture of Database Schema

3.3 비교 시스템의 구현

이상과 같은 개발 환경과 데이터베이스 스키마를 기반으로 분석 및 설계를 통하여 동일한 플랫폼 개발 환경에서 EJB 3.0과 EJB 3.1로 구현된 파일럿 시스템은 [Fig. 5]와 [Fig. 6]이다.



[Fig. 5] Pilot System of EJB 3.0



[Fig. 6] Pilot System of EJB 3.1

<Table 2> Comparison of interface and class files

Items	EJB 3.0		EJB 3.1	
	Session	Entity	Session	Entity
THEATER	TeaterRemote TeaterRemoteBean	TheaterEntity	TheaterBean	TheaterEntity
MOVIE	MovieRemote MovieRemoteBean	MovieEntity	MovieBean	MovieEntity
SUB-THEATER	SubTheaterRemote SubTheaterRemoteBean	SubTheaterEntity	SubTheaterBean	SubTheaterEntity
MOVIE-SCHEDULE	MovieScheduleRemote MovieScheduleRemoteBean	MovieScheduleEntity	MovieScheduleBean	MovieScheduleEntity
BOOK	BookRemote BookRemoteBean	BookEntity	BookBean	BookEntity
MEMBER	MemberRemote MemberRemoteBean	MemberEntity	MemberBean	MemberEntity
Total	12	6	6	6

4.2 LoC의 비교

(1) 세션빈의 LoC 비교

일반적으로 소프트웨어 개발 생산성을 비교할 때 LoC 비교 방법을 자주 사용한다. 따라서 본 연구에서도 동일한 플랫폼에서 EJB 3.0과 EJB 3.1 사양에서의 세션빈에 대한 LoC를 비교해 보면, <Table 3>과 같이 EJB 3.0에서는 빈 클래스를 생성하고, 찾고, 삭제할 수 있는 메서드들을 선언하는 홈 인터페이스가 필요 없기 때문에 명시적인 복잡도가 줄어드는 효과가 있다. 그러나 EJB 3.1에서는 패키징 구조가 단순화 됨에 따라 리모트 인터페이스가 필요 없기 때문에 EJB 3.0 보다 25% LoC가 줄어드는 것으로 나타났다.

(2) 엔티티빈의 LoC 비교

EJB 3.0과 EJB 3.1 사양의 엔티티빈 부분은 <Table 4>와 같이 호출할 수 있는 메서드를 선언하는 리모트 인터페이스와 홈 인터페이스 부분이 필요 없기 때문에 LoC만 비교해 보면, EJB 3.0과 EJB 3.1의 LoC는 같게 나타났다. 또한 리모트 인터페이스와 홈 인터페이스의 구현이 필요하지 않아 소프트웨어를 개발하기 위한 복잡도 감소와 디버깅의 효율성이 가능하게 된 긍정적인 효과도 있다.

4. 평가 지표에 따른 비교 분석

4.1 파일 개수의 비교

정량적인 소프트웨어 평가지표 중에서 가장 기본적인 사양별 파일 개수를 비교하기 위하여 [Fig. 4]와 같은 데이터베이스 스키마 구조를 기반으로 EJB 3.0과 EJB 3.1의 파일럿 시스템을 구현하여 인터페이스 파일과 클래스 파일의 개수를 비교해 보면, EJB 3.0에서 세션빈의 개수는 12개가 필요하고, EJB 3.1에서는 6개만 있으면 구현이 가능하다. 또한 엔티티빈의 경우에는 EJB 3.0과 EJB 3.1은 같이 6개만으로 구현이 가능하다. 따라서 EJB 3.1에서는 EJB 3.0보다 세션빈의 소프트웨어 생산성의 극대화를 볼 수 있다. 이상과 같은 결과를 요약하면 <Table 2>와 같다.

<Table 3> LoC Comparison of Session Bean

Items	File Name		LoC		Note
	EJB 3.0	EJB 3.1	EJB 3.0	EJB 3.1	
THEATER	TheaterRemoteBean	TheaterBean	83	85	EJB 3.0:109
	TeaterRemote	-	26	-	EJB 3.1:85
MOVIE	MovieRemoteBean	MovieBean	87	89	EJB 3.0:110
	MovieRemote	-	23	-	EJB 3.1:89
SUB-THEATER	SubTheaterRemoteBean	SubTheaterBean	85	87	EJB 3.0:108
	SubTeaterRemote	-	23	-	EJB 3.1:87
MOVIE SCHEDULE	MovieScheduleRemoteBean	MovieScheduleBean	92	94	EJB 3.0:121
	MovieScheduleRemote	-	29	-	EJB 3.1:94
BOOK	BookRemoteBean	BookBean	116	118	EJB 3.0:141
	BookRemote	-	25	-	EJB 3.1:118
MEMBER	MemberRemoteBean	MemberBean	74	76	EJB 3.0:97
	MemberRemote	-	23	-	EJB 3.1:76
Total(Unit : Line)			686	549	

<Table 4> LoC Comparison of Entity Bean

Items	File Name		LoC		Note
	EJB 3.0	EJB 3.1	EJB 3.0	EJB 3.1	
THEATER	TheaterEntity	TheaterEntity	92	92	EJB 3.0:92 EJB 3.1:92
MOVIE	MovieEntity	MovieEntity	147	147	EJB 3.0:147 EJB 3.1:147
SUB-THEATER	SubTheaterEntity	SubTheaterEntity	60	60	EJB 3.0:60 EJB 3.1:60
MOVIE SCHEDULE	MovieScheduleEntity	MovieScheduleEntity	110	110	EJB 3.0:110 EJB 3.1:110
BOOK	BookEntity	BookEntity	101	101	EJB 3.0:101 EJB 3.1:101
MEMBER	MemberEntity	MemberEntity	116	116	EJB 3.0:116 EJB 3.1:116
Total(Unit : Line)			626	626	

4.3 XML의 비교

EAR 파일은 내부적으로 엔터프라이즈 빈을 위한 JAR 파일, 웹 응용프로그램을 위한 WAR 파일, 응용프로그램 클라이언트를 위한 JAR 파일 등을 포함하고 있다. 각 파일들은 내부적으로 자바 클래스 파일과 XML 문서 파일을 가지고 있다. 이 XML 파일을 배포서술자(Deployment Descriptor)라고 하며 엔터프라이즈 빈들에 대한 선언적인 정보를 가지고 있다. 이 배포서술자는 프로그램 코드가 아닌 트랜잭션 처리나 보안 처리, 네이밍

서비스와 같은 선언적인 정보를 이용하여 엔터프라이즈 빈의 행동을 정의할 수 있다. 본 연구에서 EJB 3.0과 EJB 3.1 환경의 파일럿 시스템 배포서술자의 비교표는 <Table 5>와 같다.

<Table 5> XML Comparison of EJB Versions

Items	Deployment Descriptor	EJB 3.0		EJB 3.1	
		THEATER	MOVIE	THEATER	MOVIE
EJB	persistence.xml	12	10	12	10
		10	10	10	10
		10	10	10	10
		10	10	10	10
		16	16	16	16
		8	8	8	8
	ejb-jar.xml	X		X	
	weblogic-ejb-jar.xml	X		X	
	weblogic-cmp-rd-bms-jar.xml	X		X	
	WEB	web.xml	10		10
weblogic.xml	4		4		
Total(Unit : Line)		80		80	

위의 <Table 5>와 같이 엔터프라이즈 빈의 행동을 정의하는 배포서술자는 EJB 3.0과 EJB 3.1은 모든 빈들이 매우 단순해 졌으며, WAS 벤더들과 무관한 표준 사양으로 작성되어 소프트웨어의 독립성과 생산성을 확보하게 되었다.

5. 결론

현재까지 대용량 분산 객체 컴퓨팅 환경의 가장 성공 모델로 알려진 EJB에 대하여 표준 사양들이 지속적으로 발표되고 있지만, 실무 프로젝트에서는 EJB의 기술 스킬 습득의 시간이 길고 표준 사양의 복잡도가 높음에 따라 쉽게 새로운 사양들을 현업에 적용하지 못하는 실정이다. 또한 EJB의 소프트웨어 개발 생산성 비교에 대한 연구도 사양의 일부분 LoC 분석만 시도하고 있으며, EJB의 새로운 사양이 발표됨에도 현재까지 구체적인 평가 지표 개발과 평가 지침이 부족하여 소프트웨어 생산성의 평가와 프로젝트의 새로운 시도에 제한이 있었다.

따라서 본 연구에서는 같은 플랫폼 상에서 EJB 3.0과

EJB 3.1 사양을 기반으로 파일럿 프로젝트를 개발하여 평가 지표를 제시하였으며, 새로운 EJB 사양에 대한 개발된 평가 지표에 따라 정량적인 분석을 통하여 객관적인 소프트웨어 개발 생산성 연구에 대한 특징과 장단점을 제시하였다. 향후 Spring 3.0/3.1 프레임워크 상에서 EJB 3.1/3.2와의 소프트웨어 개발 생산성 비교와 JavaEE(J2EE) N-Tiers 구조별 오픈 소스 플랫폼과의 평가 지표별 개발 생산성 비교에 대한 연구가 지속되어야 할 것이다.

ACKNOWLEDGMENTS

This work was supported by a research grant from Semyung University(2013)

REFERENCES

- [1] M.H. Lee and J.S. Han, Design and Implementation of Movie Reservation System with EJB 3.0 Enterprise for Standardization, The Society of Digital Policy & Management, Vol. 10(8), pp. 219-224, 2012.
- [2] M.H. Lee, Design and Implementation of Movie Reservation System with EJB 3.1 Lightweight Container Architecture, The Society of Digital Policy & Management, Vol. 11(5), pp. 307-312, 2013.
- [3] 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(3), pp. 137-142, 2012.
- [4] Debu Panda, Reza Rahman, and Derek Lane, EJB 3 in Action, Manning Publications Co., pp. 3-176, 2007.
- [5] R. Monson-Haefel, B. Burke, Enterprise JavaBeans 3.0, O'Reilly, pp. 1-150, 2006.
- [6] Sun Microsystems, JSR220: Enterprise JavaBeans, Version 3.0, Final Release, 2006.
- [7] A.R. Rubinger and B. Burke, Enterprise JavaBeans 3.1, O'Reilly, pp. 1-726, 2010.
- [8] R. M. Reese, EJB 3.1 Cookbook, PACKT Publishing, pp. 1-403, 2011.
- [9] http://java.sun.com/developer/technicalArticles/J2EE/intro_ee5/index.html
- [10] M.H. Lee, A Study on Comparison of Development Productivity of Spring 2.5 and Seam 2.0 Based JPetStore Order System, Journal of The Korea Academia-Industrial cooperation Society, Vol. 10(6), pp. 1265-1274, 2009.
- [11] M.H. Lee, A Study on Comparison of Development Productivity of Spring 2.0 and 2.5 with Lightweight Container Architecture, Journal of The Korea Academia-Industrial cooperation Society, Vol. 11(7), pp. 2610-2615, 2010.
- [12] M.H. Lee, A Study on Comparison of Development Productivity of Hibernate 3.2 and iBatis 2.3 Based Lightweight Container Architecture, Journal of The Korea Academia-Industrial cooperation Society, Vol. 12(4), pp. 1919-1926, 2011.
- [13] H.S. Chae, Object Oriented CBD Development Bible, Hanbit Media, pp. 35-76, 2006.

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