

JPetStore 주문시스템 기반으로 Spring 2.5와 Seam2.0의 개발 생산성 비교 연구

이명호^{1*}

¹세명대학교 전자상거래학과

A Study on Comparison of Development Productivity of Spring 2.5 and Seam 2.0 Based JPetStore Order System

Myeong-Ho Lee^{1*}

¹Department of eCommerce, Semyung University

요약 본 논문은 Spring 2.5와 Seam 2.0 프레임워크의 연관된 객체지향 소프트웨어 개발에 대한 지침과 평가 지표를 제공하는데 목적이 있다. 경량 컨테이너 아키텍처는 EJB 아키텍처와 같이 무겁지 않으면서도 EJB 컨테이너의 모든 기능을 제공하는 구조이다. 현재까지 경량 컨테이너 아키텍처로 현업에서 가장 많이 사용되고 잘 알려진 아키텍처로 Spring과 Seam 프레임워크가 있다. 그러나 동일한 플랫폼 상에서 Spring과 Seam 프레임워크의 개발 생산성 비교 분석은 없었다. 따라서 본 연구에서는 Java EE 플랫폼 상에서 개발된 자바 블루프린트 Pet Store 프로그램을 동일한 개발 플랫폼 환경을 기반으로 JPetStore 주문시스템을 Spring 2.5와 Seam 2.0에서 설계하고 구현한 후, 개발 플랫폼 환경별 객관적인 소프트웨어 개발 생산성 비교 및 표준화에 따른 평가 지침을 제공하고자 한다.

Abstract This paper proposes an object-oriented software development guidance and an evaluation index for the productivity related to Spring 2.5 and Seam 2.0 framework. Lightweight container architecture, such as the EJB, but not heavy, to provide all of the architecture is possible. Currently in production until the lightweight container architecture, known most commonly used architecture framework is Spring and Seam. However, there is no comparison research about the performance of Spring 2.5 and Seam 2.0 framework with same identical platform. In this study, the Java Pet Store blueprint application development platform environment based on the same ordering system JPetStore Spring 2.5 and Seam 2.0 is in the design and implementation. In addition, comparison and standardization of software development productivity assessment is to provide guidance.

Key Words : Spring 2.5 and Seam 2.0 Framework, Lightweight Container Architecture, Software Development Productivity

1. 서론

디지털 컨버전스 환경에서는 대용량 컴퓨터 자원 관리자나 제공자가 지리적 위치와 상관없이 다양한 도메인의 컴퓨팅 자원을 이용하여 고성능 컴퓨팅 자원 집합체를 구축하고 이를 매우 유연하고 경제적으로 제공할 수 있는 차세대 클라우드 컴퓨팅 및 가상화 환경의 도입이 필

요하게 되었다. 앞으로 사용자는 인터넷 접속과 연산 기능만 갖춘 단말기만 있으면 장소에 구애받지 않고 온라인상에서 응용프로그램의 이용과 소프트웨어 개발과 같은 다양한 컴퓨팅 서비스를 제공받을 수 있다. 기업 사용자는 사용한 만큼의 비용만을 클라우드 컴퓨팅 제공회사에 지급하면 되므로 컴퓨팅 비용을 절감하고 이동성을 제공받을 수 있으며, 제공회사는 대량의 가입자를 유치하

*교신저자 : 이명호(mhlee@semyung.ac.kr)

접수일 10년 06월 15일

수정일 10년 07월 05일

게재확정일 10년 07월 06일

여 수입을 창출할 수 있는 무한한 가능성의 시장이라 할 수 있는 것이다[3]. 또한 엔터프라이즈 환경에서는 이기종 컴퓨터들 간에 프로그램을 분산시켜 부하를 줄여 시스템의 성능 저하와 네트워크 병목 현상을 줄일 수 있는 분산객체 구조가 필요하게 되었으며, 이를 해결하기 위한 컴포넌트 기반이나 애자일 개발 방법론까지 이르게 되었다. 현재 인정되고 있는 컴포넌트 모델의 표준 중에서 대용량 분산 객체의 가장 성공모델로 알려진 것이 EJB이다 [1-3]. EJB는 N-Tier의 분산 객체지향 자바 애플리케이션을 개발하고 보급하기 위한 컴포넌트 구조로서 1998년에 SUN사에서 EJB 1.0 사양을 만들면서 J2EE 플랫폼의 핵심 컴포넌트로 자리 잡았다. 2000년에 EJB 1.1 발표, 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, 엔티티빈의 변화를 통하여 시대의 요구사항에 맞춰 좀 더 유연하고, 테스트 및 개발하기 쉬운 새로운 EJB로 탄생하게 되었다. 그러나 EJB의 단점은 분산 환경을 지원하기 위하여 객체를 직렬화하는 과정 때문에 실행 속도의 저하가 발생하며, 개발 주기가 소스 수정, 빌드, 배포, 테스트와 같은 복잡한 과정을 거치기 때문에 개발 생산성의 저하가 일어나며, 테스트의 어려움으로 제품의 품질저하, 변형된 패턴들로 인한 객체 지향적으로 개발하는데 제약사항도 발생하며, 대형 벤더사들의 EJB 컨테이너 사이의 이식성 저하 등이 발생한다[9]. Non EJB와 EJB 아키텍처가 가지고 있는 문제점을 해결하고 장점들을 지원하기 위하여 새롭게 등장한 아키텍처가 경량 컨테이너 아키텍처(Lightweight Container Architecture)이다. 이와 같이 경량 컨테이너 아키텍처의 가장 중요한 6가지 기본 핵심가치로 아키텍처 리팩토링에 의해서 확장할 수 있는 단순한 아키텍처 구성, 소프트웨어 개발 생산성 확보, 객체지향 중심적, 비즈니스 요구사항의 중요성, 기술과 아키텍처의 검증과정의 중요성, 그리고 테스트 가능성 등의 지향점을 추구하기 위한 결과물로 등장한 것이 Spring 프레임워크이다[2]. 또한 위의 단점을 해결하기 위하여 자바언어를 위한 경량 ORM 서비스인 하이버네이트를 개발한 Gavin King이 2005년에 Seam을 개발하였다. Seam은 차세대 Web 2.0을 구현하기 위한 새로운 애플리케이션 프레임워크이다. 이를 위하여 Ajax, JSF, EJB3, Java Portlets, jBPM등을 결합시키는 기술이다. 아키텍처와 API 수준에서의 복잡성을 줄이도록 디자인되었으며, 복잡한 웹 애플리케이션의 구성을 어노테이션을 사용한 POJO로 쉽게하였고 컴포넌트화시킨 UI 위젯을 제공하고, XML 사용을 줄였다[5,8]. 특히 SUN사에서는 자바 블루프린트 프로그램이라는 J2EE 개발자들을

위해 내놓은 애플리케이션 프로그래밍 모델이 있다. J2EE 블루프린트 프로그램은 EJB, Servlet, JSP, JDBC, XML, JNDI와 같은 J2EE의 최신 기술들을 사용하여 엔터프라이즈 애플리케이션을 구축할 때 필요한 프로그래밍 모델과 이에 따른 분석 및 설계 구현을 위한 추천사항을 제공하며, 이러한 추천사항들을 실제구현을 통해 보여주고 있는 자바 Pet Store 인터넷 전자상거래 애플리케이션의 가장 잘 알려진 예제 시스템이다[4]. 따라서 본 연구에서는 Java EE 플랫폼 상에서 개발된 자바 블루프린트 자바 Pet Store 프로그램을 동일한 개발 플랫폼 환경을 기반으로 JPetStore 주문시스템을 Spring 2.5와 Seam 2.0에서 설계하고 구현함으로써 개발 플랫폼 환경별 객관적인 소프트웨어 개발 생산성을 비교하고, 표준화에 따른 평가 지표를 제공하고자 한다.

2. 프레임워크의 기본 개념

2.1 비교 프레임워크의 고찰

Spring 프레임워크의 첫번째 버전은 2002년 10월 Rod Johnson이 처음 소개하였으며, 2003년 6월에 아파치 2.0 라이선스로 릴리즈 되었다. 2004년 3월에 첫번째 Spring 프레임워크 1.0, 2006년에 2.0, 그리고 2007년 11월에 2.5가 릴리즈 되었다. 그러나 Spring 프레임워크 2.5에서 기존 2.0 버전과 비교하여 새로운 특징의 변화가 있었다. 가장 큰 특징으로는 어노테이션을 이용한 의존성 삽입의 도입이다[9-10]. Seam은 2005년 9월에 EJB 3.0, JSF, Ajax 기술을 기반으로 차세대 새로운 애플리케이션 프레임워크로 발표되었다. 2006년 12월에 Seam 1.1GA, 2007년 2월에 Seam 1.2GA, 2007년 12월에 Seam 2.0GA, 그리고 2008년 10월에 Seam 2.1GA가 발표되면서 JSF 태그와 Facelets 템플릿을 사용한 Excel과 CSV에서의 리포트를 생성하고, Seam 시작시 리소스 배포능력을 갖추었다. JBoss jBPM을 통해 비즈니스 프로세스 관리를 통합하고, 모델, 구현, workflow와 pageflow를 보다 쉽게 생성할 수 있도록 화면 흐름의 관리가 가능하다. Seam 컴포넌트나 POJO는 테스트가 가능하나 복잡하고 단독으로 유닛 테스트하기 부적합하지만 쉬운 애플리케이션의 시험 가능성을 제공하고 JUnit이나 TestNG에 기록할 수 있는 통합 테스트의 용이성이 있다[5,8].

2.2 비교 프레임워크의 구성

Spring이 시작한 의존성 삽입 기술은 피코 컨테이너, EJB3.0, SEAM, Google Guice 등의 다양한 프레임워크와

4. 프레임워크별 비교 평가

4.1 Ajax의 평가

다음 표 2는 Spring 2.5와 Seam 2.0 프레임워크의 JPetStore 주문시스템에서 사용한 회원가입 폼 ID 확인에 서 Ajax를 비교한 평가표이다.

[표 2] 회원가입 폼 ID의 Ajax 평가표

항 목	Spring 2.5		Seam 2.0		
	Loc	Files	LoC	Files	
jpg / xhtml	Script	18	1	-	1
	Code	3	위와 동일	8	위와 동일
Java	30	1	26	1	
xml (petstore-servlet.xml)	4	1	-	-	
js (login.js /prototype.js)	-	2	-	-	
합 계	55	5	34	2	

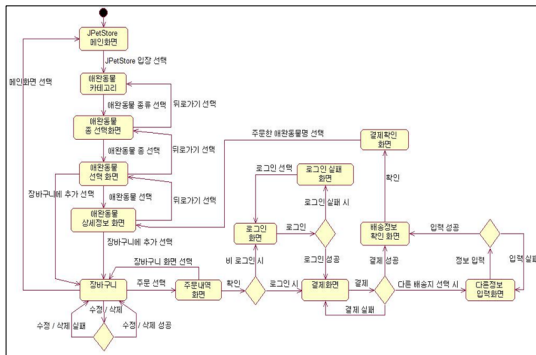
회원가입 폼 ID에서의 Ajax를 비교 평가해 보면, Seam에서는 스크립트 및 xml 통신을 구현하지 않고 적은 코드로 쉽게 구현이 가능하고 Spring에 비하여 디버깅에 사용할 수 있다. 반면에 Spring은 클라이언트에서 자바스크립트를 사용할 때 디버깅이 힘들다. 따라서 Seam이 Spring보다 Ajax에 필요한 LoC는 38%, 파일은 60%의 감소가 나타났다.

4.2 JSF와 JSTL의 태그 평가

다음 표 3은 Spring 2.5와 Seam 2.0 프레임워크의 JPetStore 주문시스템에서 사용한 JSF와 JSTL의 태그를 비교한 평가표이다.

[표 3] JSF와 JSTL의 태그 평가표

항목	Spring 2.5		Seam 2.0	
	LoC	Files	LoC	Files
jpg / xhtml	9	2	8	1
Java	Controller.java	6	1	-
	메시지출력	-	1	-
properties (메시지출력)	-	-	-	1
합계	15	4	8	2



[그림 4] 주문시스템의 화면 흐름도

이상과 같은 데이터베이스 스키마를 기반으로 분석 및 설계를 통하여 Spring 2.5와 Seam 2.0의 파일럿 시스템을 구현한 메인 화면은 그림 5, 그림 6과 같다.



[그림 5] Spring 2.5의 메인 화면



[그림 6] Seam 2.0의 메인 화면

JPetStore 주문시스템에서 사용한 JSF와 JSTL의 태그를 비교 평가해 보면, Seam에서는 JSF의 편리한 입력값의 유효성을 검증할 수 있으며, Seam이 Spring보다 유효성 검사에 필요한 LoC는 47%, 파일은 50%의 감소가 나타났다.

4.3 한글처리의 평가

다음 표 4는 Spring 2.5와 Seam 2.0 프레임워크의 JPetStore 주문시스템에서 사용한 한글처리를 비교한 평가표이다.

[표 4] 한글처리 평가표

항 목	Spring 2.5		Seam 2.0	
	LoC	Files	LoC	Files
page.jsp/xhtml	1	1	1	1
web.xml	12	1	8	1
합 계	13	2	9	2

JPetStore 주문시스템에서 사용한 한글처리를 비교 평가해 보면, Seam이 Spring보다 한글처리에 필요한 LoC는 30%의 감소가 나타났다.

4.4 XML의 평가

다음 표 5는 Spring 2.5와 Seam 2.0 프레임워크의 JPetStore 주문시스템에서 사용하는 xml 파일 항목들에 대한 LoC 및 파일들을 요약한 평가표이다.

[표 5] XML 평가표

항 목	Spring 2.5		Seam 2.0	
	LoC	Files	LoC	Files
xml	406	8	293	8
합 계	406	8	293	8

다음 표 6은 Spring 2.5와 Seam 2.0 프레임워크의 JPetStore 주문시스템에서 사용하고 있는 xml 파일들의 비교표이다.

[표 6] XML 비교표

Spring 2.5	Seam 2.0
applicationContext.xml dataAccessContext-jta.xml dataAccessContext-local.xml petstore-servlet.xml remoting-servlet.xml server-config.wsdd sql-map-config.xml web.xml	components.xml face-config.xml jboss-web.xml pages.xml web.xml Seam-ds.xml ejb-jar.xml persistence.xml
LoC : 406	LoC : 293

Seam에서는 XML이 아닌 어노테이션을 사용하며, 되도록 다른 프레임워크들과는 다르게 XML 사용을 최소화하게 설계된다. 따라서 Seam이 Spring보다 한글처리에 필요한 LoC는 28%의 감소가 나타났다.

5. 결론

프로젝트를 실패를 만들지 않기 위하여 가장 먼저 생각할 것은 단순성, 생산성, 객체지향성, 고객이 원하는 핵심 요구사항에 중요성, 실증적인 방법으로 기술을 도입 가능성, 그리고 테스트 가능성 등이다. 이러한 핵심가치들을 추구하기 위하여 등장한 것이 Spring 프레임워크이다. Seam은 차세대 Web 2.0을 구현하기 위한 새로운 애플리케이션 프레임워크로 제안되고 있으며, 이를 위하여 JSF, EJB3, Bijection, Ajax, Java Portlets, jBPM등을 결합시키는 경량 기반의 프레임워크 기술이다. 또한 Seam은 아키텍처와 API 수준에서의 복잡성을 줄이도록 디자인 되었으며, 복잡한 엔터프라이즈 웹 애플리케이션의 구성을 어노테이션을 사용한 POJO로 쉽게 하였고, 컴포넌트화된 UI 위젯을 제공한다.

그러나 현재까지 경량 컨테이너 아키텍처의 성공 모델로 알려진 Spring과 Seam 프레임워크 사양의 정량적인 성과지표 개발 및 사례의 부족으로 이전 사양으로 운영 중인 실무 프로젝트의 업그레이드나 새로운 기술 사양의 적용이 미비하였다. 그 이유는 기본적인 Spring과 Seam 프레임워크의 기술 변화의 속도가 빠르고 표준 사양의 복잡도가 높음에 따라 쉽게 새로운 사양들을 현업에 적용하지 못한 것이다. 또한 Spring과 Seam 프레임워크의 소프트웨어 개발 생산성 비교에 대한 연구도 부족한 상태이며, Spring과 Seam 프레임워크의 새로운 사양이 발표됨에도 현재까지 구체적인 분석 및 설계 기반에 따른 구현 지침이 부족하여 소프트웨어 생산성의 평가와 프로

젝트의 새로운 시도에 제한이 있었다.

따라서 본 연구에서는 대용량 분산객체 시스템 처리를 위하여 Spring 2.5와 Seam 2.0 프레임워크를 기반으로 JPetStore 주문시스템의 비교분석 및 설계를 통하여 구현 지침을 제시하였다. 또한 Spring과 Seam 프레임워크에 대한 성능 평가 기반으로 정량적인 분석을 통하여 객관적인 소프트웨어 개발 생산성 연구에 대한 지침을 제시하였다. 향후에는 동일한 데이터베이스 스키마를 이용하여 iPhone과 Android 운영체제에서의 모바일 소프트웨어 개발 생산성 분석 연구가 지속되어야 할 것이다.

참고문헌

- [1] 이명호, “EJB2.0과 EJB3.0의 소프트웨어 개발 생산성 비교 연구”, 한국산업경영시스템학회지, 제31권 제3호, pp. 1-7, 2008.
- [2] 이명호, “동일한 경량 컨테이너 구조 환경에서 스프링 프레임워크 2.0과 2.5의 개발 생산성 비교 연구”, 한국산학기술학회논문지, 제10권 제6호, pp. 1265-1274, 2009.
- [3] 이명호, “JBoss Seam 프레임워크를 기반으로 JPetStore 주문시스템의 설계 및 구현”, 한국산학기술학회논문지, 제11권 제5호, pp. 1708-1715, 2010.
- [4] 이용원, 박지훈외, “J2EE 응용과 디자인 패턴”, 대청, pp. 603- 785, 2002.
- [5] Dan Allen, “SEAM In Action, Manning”, pp. 3-136, 2009.
- [6] Joseph F. Nusairat, “Beginning JBoss Seam:From Novice to Professional”, Apress, pp. 1-186, 2007.
- [7] Kito D. Mann, “JavaServer Faces In Action”, Manning, pp. 4-184, 2005.
- [8] Michael J. Yuan, Jacob Orshalick, and Thomas Heute, “Seam Framework”, Prentice Hall, pp. 3-73, 2009.
- [9] Rod Johnson, “Expert One-on-One J2EE Design and Development”, Wrox, pp. 441-673, 2002.
- [10] Rod Johnson, and Juergen Hoeller, “Expert One-on-One J2EE Development without EJB”, Wrox, pp. 1-141, 2004.

이 명 호(Mycong-Ho Lee)

[종신회원]



- 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