

JBoss Seam 프레임워크를 기반으로 JPetStore 주문시스템의 설계 및 구현

이명호^{1*}

¹세명대학교 전자상거래학과

Design and Implementation of JPetStore Order System Based JBoss Seam Framework

Myeong-Ho Lee^{1*}

¹Department of eCommerce, Semyung University

요약 본 논문은 Seam 프레임워크와 연관된 객체지향 소프트웨어 개발에 대한 지침과 평가 지표를 제공하는데 목적이 있다. 기존의 아키텍처가 가지고 있는 문제점을 해결하고 장점들을 지원하기 위하여 새롭게 등장한 표준화 아키텍처가 Seam 아키텍처이다. 현재까지 중량 및 경량 컨테이너 아키텍처의 표준화를 장점으로 제안된 잘 알려진 아키텍처로 Seam 프레임워크가 있다. 따라서 본 연구에서는 기존의 중량과 경량 아키텍처가 가지고 있는 단점들을 해결하고 장점들을 지원하기 위하여 개발된 최신 경량 컨테이너 아키텍처인 Seam 프레임워크를 기반으로 JPetStore 주문시스템의 설계 및 구현을 통하여 이전의 사양과의 객관적인 소프트웨어 개발 생산성 지침을 제공하고자 한다.

Abstract This paper proposes an object-oriented software development guidance and an evaluation index for the productivity related to Seam Framework. Heavyweight and lightweight architecture to resolve the problem with benefits to support the new architecture is a large size distributed object standardization architecture. This architecture, such as the Seam Framework, to provide all of the architecture is possible. The distributed object standardization architecture is most often used in business Seam Framework is well-known architecture. Therefore, this study is based on the Seam Framework large distributed object architecture, design and implementation of JPetStore order system based standardization software development productivity and the objective is to provide guidance.

Key Words : Evaluation Index, Seam Framework, Heavyweight and Lightweight Architecture, Distributed Object

1. 서론

디지털 컨버전스 시대에서는 새로운 정보미디어나 IT 응용서비스와 같은 대용량의 요구에 대하여 컴퓨팅 자원 관리자 및 제공자가 지리적 위치와 상관없이 다양한 도메인의 자원을 이용하여 고성능 자원 집합체를 구축하고 이를 유연하고 매우 경제적으로 제공하는 차세대 클라우드 컴퓨팅 및 가상화 환경의 도입이 대두되고 있다. 사용

자는 인터넷 접속과 연산 기능만 갖춘 단말기만 있으면 장소에 구애받지 않고 온라인상에서 응용프로그램의 이용과 컴퓨팅 서비스를 제공받을 수 있다. 기업 사용자는 사용한 만큼의 비용만을 클라우드 컴퓨팅 제공회사에 지급하면 되므로 비용을 절감하고 이동성을 제공받을 수 있으며, 제공회사는 대량의 가입자를 유치하여 수입을 창출할 수 있는 무한한 가능성의 시장이라 할 수 있는 것이다[4-6]. 또한 엔터프라이즈 환경에서는 이기종 컴퓨터들

*교신저자 : 이명호(mhlee@semyung.ac.kr)

접수일 10년 04월 14일

수정일 10년 05월 07일

게재확정일 10년 05월 13일

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 표준은 MS사의 COM+, OMG의 CCM(CORBA Component Model), SUN사의 EJB(Enterprise JavaBeans) 등이 있지만, 이 중에서 대용량 분산 객체의 가장 성공모델로 알려진 것이 EJB이다 [2-6]. 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[2,3]. 또한 자바 블루프린트 프로그램은 SUN사에서 J2EE 개발자들을 위해 내놓은 애플리케이션 프로그래밍 모델이다. 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[7].

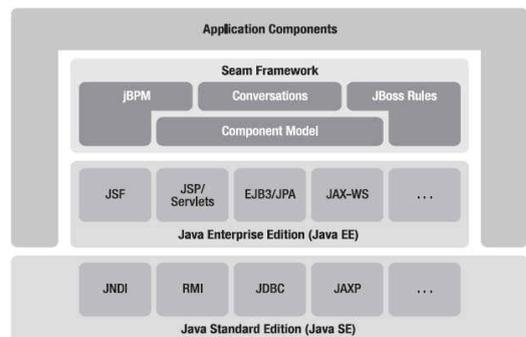
그러나 EJB의 단점은 분산 환경을 지원하기 위하여 객체를 직렬화하는 과정 때문에 실행 속도의 저하가 발생하며, 개발 주기가 소스수정, 빌드, 배포, 테스트와 같은 복잡한 과정을 거치기 때문에 개발 생산성의 저하가 일어나며, 테스트의 어려움으로 제품의 품질저하, 변형된 패턴들로 인한 객체 지향적으로 개발하는데 제약사항도 발생하며, 대형 벤더사들의 EJB 컨테이너 사이의 이식성 저하 등이 발생한다[6]. 이러한 단점을 해결하기 위하여 자바언어를 위한 경량 ORM 서비스인 하이버네이트를 개발한 가빈 킹(Gavin King)이 2005년에 Seam을 개발하였다. Seam은 차세대 Web 2.0을 구현하기 위한 새로운 애플리케이션 프레임워크이다. 이를 위하여 Ajax, JSF, EJB3, Java Portlets, jBPM등을 결합시키는 기술이다. 또한 아키텍처와 API 수준에서의 복잡성을 줄이도록 디자인되었으며, 복잡한 웹 애플리케이션 구성에 대하여 annotation을 사용한 POJO로 쉽게하였으며 컴포넌트화시킨 UI 위젯을 제공하고, XML 사용을 줄였다[1][8-10] [12].

따라서 본 연구에서는 J2EE 플랫폼 상에서 개발된 자바 블루프린트 자바 Pet Store 예제 프로그램을 JBoss Seam 프레임워크를 기반으로 JPetStore 주문시스템을 설계하고 구현함으로써 이전의 플랫폼과의 객관적인 소프트웨어 개발 생산성 및 표준화에 따른 평가 지침을 제공하고자 한다.

2. Seam 프레임워크의 기본 개념

2.1. Seam 프레임워크의 고찰

Seam은 2005년 9월 EJB 3.0, JSF, Ajax 기술 기반의 프레임워크를 최초로 발표하였다. 2006년 12월 Seam 1.1GA는 Seam Gen을 제공하고, J2EE 1.4 애플리케이션 서버에서 Seam 애플리케이션이 작성 가능하고, CRUD 애플리케이션 프레임워크를 제공한다. 2007년 2월 Seam 1.2GA는 Entity 레벨의 보안, EL의 지원, 향상된 Seam Gen과 XML을 통한 통합을 개선하고, 향상된 Seam-Gen, Ajax4JSF의 통합을 이루었다. 2007년 12월 Seam 2.0GA는 개발에서 컴포넌트 배포의 강점을 지원하였으며 개발 툴인 Eclipse과 Maven을 지원하였다. 2008년 10월 Seam 2.1GA는 Wicket의 지원을 위한 최초의 클래스이며, JSF 태그와 Facelets 템플릿을 사용한 Excel과 CSV에서의 리포트 생성하고, Seam 시작 시 리소스 배포능력을 갖추었다. 다음 그림 1은 Seam 프레임워크가 애플리케이션 코드와 Java EE 프레임워크 사이에 존재하는 것을 도식화한 것이다[10].



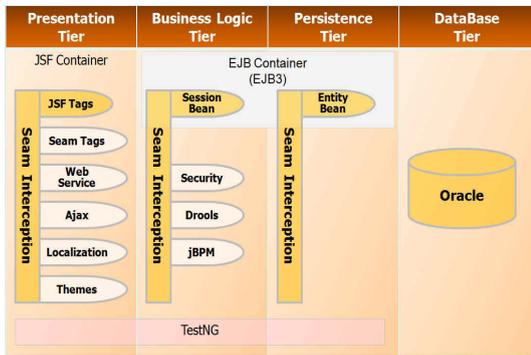
[그림 1] Java EE 프레임워크의 Seam 구성도

Seam을 사용하는 이유는 Java EE 5.0의 핵심 프레임워크인 EJB 3.0과 JSF를 연결하고 통합에 필요한 복잡한 코드를 대신 수행하여 EJB 3.0을 적용한 최초의 애플리케이션 프레임워크이다. EJB 3.0을 지원하는 환경에 제

한되지 않고 다른 Java EE 환경이나 Tomcat에서도 사용 가능한 Java EE 5.0을 지원한다. 오픈 소스 Ajax 솔루션을 기반으로 한 JSF에 통합되고, JavaScript를 배울 필요 없이 애플리케이션에 쉽게 적용 가능하며, 잠재하는 버그로부터의 보호할 수 있도록 Ajax로 구현이 용이하다. 또한 심각한 메모리 누수 및 성능 병목 현상, 보안문제, Back 버튼의 오작동, 멀티탭 브라우저의 동작 등의 HTTP 세션의 과충 문제를 극복하기 위해 stateful을 사용하며, 동일한 브라우저 내에서 여러 작업에 대한 세션을 구분하여 여러 탭의 작업들이 독립적인 세션으로 처리한다. 또한 HTTP Session의 이용이 보다 쉽고 효율적이고 정밀한 상태관리를 할 수 있는 혁신적인 상태관리 기능을 제공한다. 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[1,9].

2.2 Seam의 구성

Seam 프레임워크는 Web 2.0을 구현하기 위한 JBoss의 새로운 애플리케이션 프레임워크이다. 또한 Ajax, JSF, EJB3, Java Portlets, jBPM, Rules 등의 다양한 프레임워크 및 표준을 유기적으로 잘 결합하여 아키텍처와 API 수준에서의 복잡도를 줄이도록 설계되었다. 기존의 웹 애플리케이션에서 제공하지 못했던 다양한 상태관리 모델을 제공하며 JPA를 기본 퍼시스턴스 모델로 사용한다[8]. 따라서 본 연구에서 Seam 프레임워크 기반의 JPetrStore 주문시스템을 구현한 구성도를 살펴보면 그림 2와 같다[10].



[그림 2] Seam의 구성도

3. 비주얼 모델링의 분석 및 설계

3.1 개발 환경

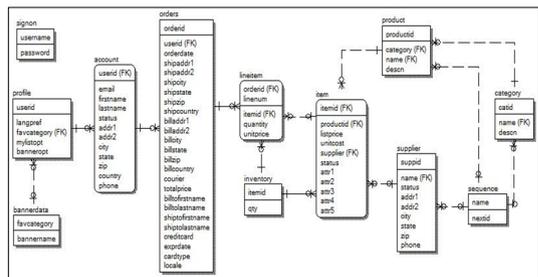
본 연구에서는 대용량 분산 객체 시스템을 개발하기 위한 Seam을 기반으로 표 1과 같은 개발환경을 이용하여 Seam 프레임워크 환경에서의 JPetStore 주문시스템 프로그램을 분석하고 설계한 후 파일럿 시스템을 구현하도록 한다.

[표 1] Seam 프레임워크의 개발 환경

구분	Seam 2.0
O/S	Windows Vista
JAVA	Java SE 1.6 / Java EE 1.5
WAS	JBoss 5.0
DB	Oracle 10g XE
DB Modeling	ERwin7.1
IDE TOOL	Eclipse 3.4
CASE TOOL	Rational Rose 2006

3.2 DB 스키마의 설계

대용량 표준 분산 객체 처리를 위한 Seam 프레임워크의 JPetStore 주문시스템의 파일럿 시스템은 테이블로 영속적인 정보를 저장하기 위하여 데이터 모델이 먼저 정의되어야 한다. 따라서 자바 Pet Store 인터넷 전자상거래 애플리케이션의 DB 스키마를 기반으로 엔티티 클래스의 데이터 모델을 작성한다. 본 연구에서 파일럿 시스템으로 제시한 데이터 모델인 데이터베이스 스키마 구조는 그림 3과 같이 로그인정보, 회원가입/수정정보, 내주문정보, 주문서확인정보, 상세정보, 장바구니정보, 결제정보, 결제입력정보, 주문확인정보, 다른배송지정보, 카테고리정보, 종선택정보, 애완동물선택정보, 그리고 검색정보 등으로 설계하여 구현한다.



[그림 3] 데이터베이스 스키마 구조

사용자 항목별 기능 내역을 요약하면 다음 표 2와 같다.

[표 2] 사용자 인터페이스별 기능 내역서

UI 항목	UI 기능 내역
첫화면	index.html
로그인 (SignonForm.jsp)	IncludeTop.jsp IncludeBottom.jsp
회원가입 및 회원수정 (EditAccountForm.jsp)	IncludeTop.jsp IncludeAccountFields.jsp IncludeBottom.jsp
내주문 (ListOrder.jsp)	IncludeTop.jsp IncludeBottom.jsp
주문서확인 (ViewOrder.jsp)	IncludeTop.jsp IncludeBottom.jsp
상세정보 (Item.jsp)	IncludeTop.jsp IncludeBottom.jsp
장바구니 (Product.jsp)	IncludeTop.jsp
	IncludeMyList.jsp
	IncludeBanner.jsp
	IncludeBottom.jsp
결제 (Checkout.jsp)	IncludeTop.jsp IncludeBottom.jsp
결제정보입력 (NewOrderForm.jsp)	IncludeTop.jsp IncludeBottom.jsp
주문정보확인 (ConfirmOrder.jsp)	IncludeTop.jsp IncludeBottom.jsp
다른배송지정보 (ShippingForm.jsp)	IncludeTop.jsp IncludeBottom.jsp
카테고리 (Index.jsp)	IncludeTop.jsp (IncludeQuickHeader.jsp)
	IncludeBanner.jsp
	IncludeBottom.jsp
	IncludeBottom.jsp
종선택 (Category.jsp)	IncludeTop.jsp IncludeBottom.jsp
애완동물선택 (Product.jsp)	IncludeTop.jsp IncludeBottom.jsp
검색 (SearchProducts.jsp)	IncludeTop.jsp IncludeBottom.jsp

엔티티 형별로 엔티티 형의 정의를 요약하면 표 3과 같다.

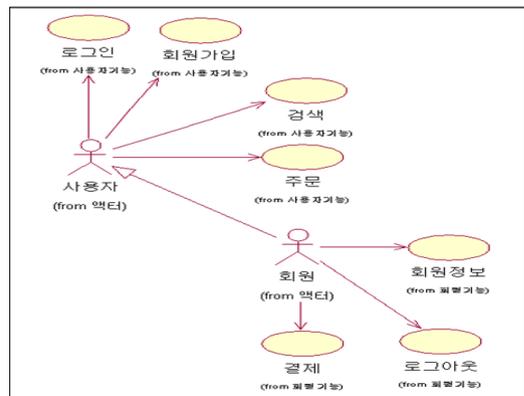
[표 3] 엔티티 형의 정의서

엔티티 형	Relation 및 Membership	필수/선택	대상 엔티티형
로그인	별개의 테이블		
프로필	하나의 프로필은 여러개의 회원정보를 가진다.	선택	회원 정보
	하나의 회원정보는 하나의 프로필만을 가진다.	필수	
배너	하나의 배너는 여러개의 프로필을 가진다.	선택	프로필
	하나의 프로필은 하나의 배너를 가진다.	필수	

회원 정보	한명의 회원은 여러개의 주문정보를 가진다.	선택	주문
	하나의 주문정보는 한 명의 회원정보만을 가진다.	필수	
주문	하나의 주문은 여러개의 주문정보를 가진다.	필수	주문정보
	하나의 주문정보는 하나의 주문을 가진다.	필수	
주문 정보	하나의 주문정보는 하나의 재고와 애완동물 상세정보를 가진다.	필수	재고, 애완동물 상세정보
	하나의 재고와 애완동물 상세정보는 여러개의 주문정보를 가진다.	필수	
재고	하나의 재고는 여러개의 애완동물 상세정보를 가진다.	필수	애완동물 상세정보
	하나의 애완동물 상세정보는 하나의 재고만을 가진다.	필수	
애완 동물 상세 정보	하나의 애완동물 상세정보는 하나의 애완동물종과 공급자를 가진다.	필수	애완 동물종, 공급자
	하나의 애완동물종과 공급자는 여러개의 애완동물 상세정보를 가진다.	필수	
애완 동물종	하나의 애완동물종은 하나의 동물 카테고리 및 시퀀스를 가진다.	필수	동물 카테고리, 시퀀스
	하나의 동물카테고리와 시퀀스는 여러개의 애완동물종을 가진다.	필수	
동물 카테고리	하나의 동물카테고리는 하나의 시퀀스를 가진다.	필수	시퀀스
	하나의 시퀀스는 하나의 여러개의 동물카테고리를 가진다.	필수	
시퀀스	하나의 시퀀스는 여러명의 공급자를 가진다.	필수	공급자
	하나의 공급자는 하나의 시퀀스를 가진다.	필수	

3.3 유스케이스 다이어그램

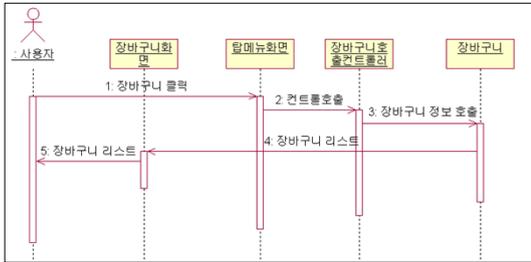
JPetStore 주문시스템의 기본적인 요구사항을 기술한 문제 기술서와 유스케이스 명세서를 기반으로 주문시스템에 대한 요구사항 정의활동으로 모델링한 결과인 유스케이스 다이어그램으로 표현해 보면 그림 4와 같은 유스케이스 모델이 된다.



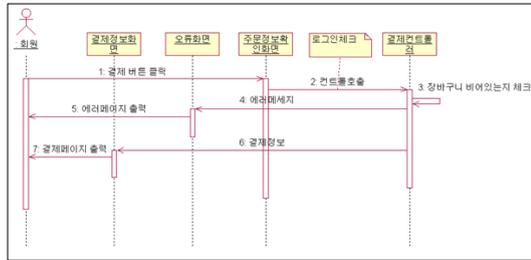
[그림 4] 주문시스템 유스케이스 다이어그램

3.4 시퀀스 다이어그램

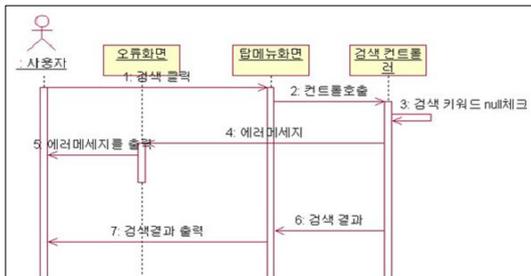
객체들 간의 메시지 전송을 시간적인 순서로 표현하는데 효과적인 다이어그램이 시퀀스 다이어그램이다. 유스케이스의 기능을 실현하기 위하여 필요한 분석 객체들의 상호작용을 시퀀스 다이어그램으로 표현한 산출물이 분석 유스케이스 실현 모델이다. 따라서 본 연구의 파일럿 시스템에서 중요한 장비구니 불러오기, 결재시 회원확인, 그리고 검색의 분석 유스케이스 실현 모델을 시퀀스 다이어그램으로 표현하면 그림 5, 그림 6, 그림 7과 같다.



[그림 5] 장비구니 불러오기 시퀀스 다이어그램



[그림 6] 결재시 회원확인 시퀀스 다이어그램



[그림 7] 검색 시퀀스 다이어그램

4. Seam 프레임워크의 평가

4.1 회원가입 폼 ID확인 Ajax의 평가

따라서 본 연구에서도 Seam 프레임워크에서의 회원 가입 폼 ID 확인 Ajax에 대한 LoC를 평가해 보면 표 4와 같다. Seam은 사용자가 검색했던 키워드 정보를 이용하여 특정 제품 추천 등과 같은 상태유지(Stateful)를 사용한 풍요로운 웹을 구현할 수 있다. 또한 스크립트 및 xml 통신을 구현하지 않고 적은 코드로 쉽게 구현이 가능하다.

[표 4] 회원가입 폼 ID 확인 Ajax 평가표

항 목		Seam 2.0	
		LoC	Files
jpg / xhtml	Script	0	1
	Code	8	위와동일
Java		26	1
xml (petstore-servlet.xml)		0	0
js (login.js /prototype.js)		0	0
합 계		34	2

사용자 인터페이스 중 회원가입 및 회원수정에서 ID 입력과 메시지 출력에 대한 구현 내용은 다음과 같다.

```

<h:inputText id="ID" value="#{user.id}" required="true">
  <a:support event="onkeyup"
    actionListener="#{accountIm.idSearch(user.id)"
    reRender="searchId" />
</h:inputText>

<a:outputPanel id="searchId">
  #{accountIm.ajax}
</a:outputPanel>
    
```

또한 컨텍스트(Context)로부터 인젝션(Injection)을 받으며, 만약 객체 변경이 일어나면 자동으로 컨텍스트를 다음과 같이 업데이트 시킨다.

```

@In(required=false)
@Out(required=false)
String ajax;

public void idSearch(String id) {
  User userSearch= null;
  try {
    setAjax(null);
    userSearch = entityManager.find(User.class, id);
    if(userSearch!=null && id!=""){
      setAjax("중복 아이디");
    }else if(userSearch==null && id!=""){
      setAjax("사용가능 아이디");
    }
  }
}
    
```

```

    }else{
        setAjax("ID를 입력해주세요");
    }
}
}catch (Exception e) {
    setAjax("에러");
}
}
}
public String getAjax() {
    return ajax;
}
}
public void setAjax(String ajax) {
    this.ajax = ajax;
}
}

```

따라서 Seam 프레임워크에서는 간단하고 쉽게 Ajax를 구현할 수 있으면, 다른 프레임워크들 보다 디버깅이 쉽다.

4.2 JSF와 JSTL의 태그 평가

JSF(JavaServer Faces)는 컴포넌트 구조, 표준 위젯 집합, 애플리케이션 기반 구조를 정의하고 있다. 컴포넌트 구조는 UI 위젯을 만드는 공통적인 방법을 정의한다. 또한 사용자의 입력값을 수집하고 이벤트에 응답하는 자바 객체인 기반 빈(Backing Bean)과 UI 컴포넌트를 자동으로 동기화시켜준다. 또한 개발자가 자바 코드가 아닌 HTML과 같은 식의 태그를 이용하여 UI를 정의할 수 있는 표준 태그 라이브러리인 JSTL(JavaServer Pages Standard Tag Library)도 있다[11].

Seam 프레임워크에서 JPetStore 주문시스템의 JSF와 JSTL 태그 평가표를 요약하면 표 5와 같다.

[표 5] JSF와 JSTL의 태그 평가표

항목		Seam 2.0	
		LoC	Files
jpg / xhtml		8	1
Java	Controller.java	0	0
	메시지출력	0	0
properties (메시지출력)		-	1
합계		8	2

다음과 같이 Seam에서의 dataTable은 단지 테이블의 컬럼을 정의하는 것만으로 컴포넌트가 행을 순회하며, dataTable 컴포넌트가 HTML을 만들어내므로 테이블을 위한 HTML이 전혀 필요가 없다.

```
<h:dataTable value="#{proFavo}" var="testt" bgcolor="#8b4513"
```

```

border="0" cellspacing="2" cellpadding="3">
<h:column>
<f:facet name="header">
<h:outputText value="추천 애완동물." />
</f:facet>
<s:link action="#{categoryView.categoryDetail(testt)}"
value="#{testt.name}" />
<h:outputText value="( #{testt.productId} )" />
</h:column>
</h:dataTable>
<f:subview id="favorPage"
rendered="#{authenticator.page.equals('ture')}">
<s:link action="#{authenticator.nextpage(4, 0)}">
<h:outputText value="다음" />
</s:link>
</f:subview>
<f:subview id="favorPage2"
rendered="#{authenticator.page2.equals('back')}">
<s:link action="#{authenticator.backpage()}">
<h:outputText value="이전" />
</s:link>
</f:subview>

```

또한 아래와 같이 Seam으로 JSF의 사용과 입력값의 유효성 검증은 매우 편리하며, 강력한 애플리케이션을 개발할 수 있는 프레임워크를 갖추고 있다. 그러나 Seam에서는 테이블을 디자인하는데 어려움이 발생한다.

```

<div style="float: left; width: 85px" align="right">도시:</div>
<div align="left"><h:inputText id="city" value="#{orders.billCity}" size="17"
required="true"/><h:message for="city" styleClass="errors"/></div>

```

4.3 한글처리 평가

Seam에서 한글처리는 아래와 같이 필터에 param 설정 값이 없고 각 XHTML에 UTF-8 형식만 정의하면 된다. 그러나 페이지에서는 UTF-8의 설정 위치가 다르다. 한글 처리 평가표를 요약하면 표 6과 같다.

[표 6] 한글처리 평가표

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

```

[*].xhtml]
<head>
<meta http-equiv="content-type"
content="application/xhtml+xml; charset=utf-8" />
</head>

```

```
[web.xml]
<filter>
  <filter-name>Seam Filter</filter-name>
  <filter-class>org.jboss.seam.servlet.SeamFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>Seam Filter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

4.4 XML 평가

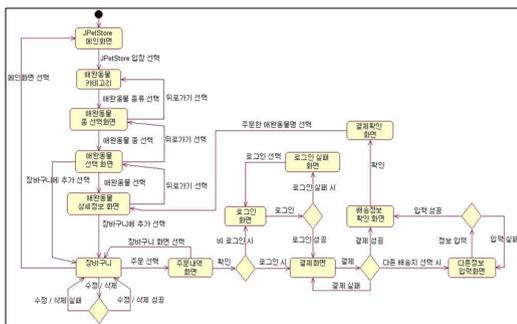
다음 표 7은 Seam 프레임워크에서 JPetStore 주문시스템에서 사용하는 xml 파일 항목들에 대한 LoC 및 파일들을 요약한 평가표이다. Seam에서는 XML이 아닌 annotation을 사용하며, 되도록 다른 프레임워크들과는 다르게 XML 사용을 최소화하게 설계된다.

[표 7] XML 평가표

항 목	Seam 2.0	
	LoC	Files
(components.xml) (face-config.xml) (jboss-web.xml) (pages.xml) (web.xml) (Seam-ds.xml) (ejb-jar.xml) (persistence.xml)	293	8
합 계	293	8

5. 파일럿 시스템의 구현

Seam 프레임워크 기반에서 파일럿 시스템은 각 유스 케이스에 사용되는 화면 간의 전환을 화면 흐름 모델로 설계함으로써 명시적으로 분석하였다. 다음 그림 8은 JPetStore 주문시스템의 화면 흐름 모델을 나타낸다.



[그림 8] JPetStore 주문시스템의 화면 흐름도

이상과 같은 개발 환경과 데이터베이스 스키마를 기반으로 Seam 프레임워크 기반에서 JPetStore 주문시스템을 구현한 메인 화면은 그림 9와 같다.



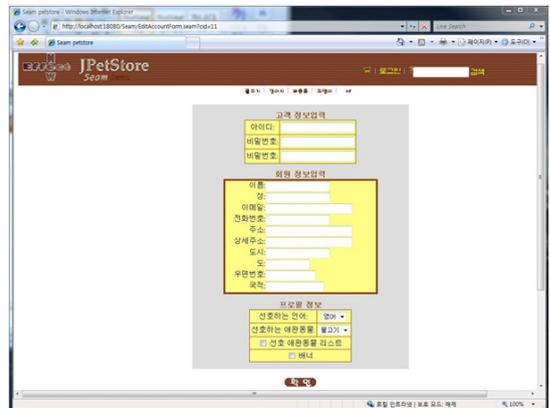
[그림 9] JPetStore 주문시스템의 메인 화면

JPetStore 주문시스템 중에서 구현된 회원가입을 위한 메인 화면을 살펴보면 다음 그림 10과 같다.



[그림 10] 회원가입의 메인 화면

JPetStore 주문시스템 중에서 구현된 회원가입을 위한 실행 화면을 살펴보면 다음 그림 11과 같다.



[그림 11] 회원가입의 실행 화면

6. 결론

Seam 프레임워크는 엔터프라이즈 웹 애플리케이션에서 모든 컴포넌트들에 대하여 일관되고 쉽게 이해되는 프로그래밍 모델을 제공하기 위하여 Java EE 5.0의 최상위에 놓여있으며, 애플리케이션으로 부터 대부분의 상용구 코드나 XML 구성을 제거하였다. 통합 문제를 해결함에 따라 Seam은 웹 개발자들에게 이전에 웹 애플리케이션으로 통합하기가 너무 어려웠던 툴들을 유용하게 개방하였다[10]. 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 디자인되었으며, 복잡한 엔터프라이즈 웹 애플리케이션의 구성을 annotation을 사용한 POJO로 쉽게 하였고, 컴포넌트화된 UI 위젯을 제공한다. 그러나 현재까지 경량 컨테이너 아키텍처의 성공 모델로 알려진 Seam 프레임워크 사양의 정량적인 성과지표 개발 및 사례의 부족으로 이전 사양으로 운영 중인 실무 프로젝트의 업그레이드나 새로운 기술 사양의 적용이 미비하였다. 그 이유는 기본적인 Seam 프레임워크의 기술 변화의 속도가 빠르고 표준 사양의 복잡도가 높음에 따라 쉽게 새로운 사양들을 현업에 적용하지 못한 것이다. 또한 Seam 프레임워크의 소프트웨어 개발 생산성 비교에 대한 연구도 부족한 상태이며, Seam 프레임워크의 새로운 사양이 발표됨에도 현재까지 구체적인 분석 및 설계 기반에 따른 구현 지침이 부족하여 소프트웨어 생산성의 평가와 프로젝트의 새로운 시도에 제한이 있었다.

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

참고문헌

- [1] 김병근, 오재용, 원종석, "JBoss Application Server 5," 가메출판사, pp. 21-196, 2009.
 [2] 이명호, "EJB 3.0 표준을 기반으로 대용량 분산객체 처

리의 설계 및 구현", 대한설비관리학회지, 제13권 제2호, pp. 45-51, 2008.

- [3] 이명호, "EJB2.0과 EJB3.0의 소프트웨어 개발 생산성 비교 연구", 한국산업경영시스템학회지, 제31권 제3호, pp. 1-7, 2008.
 [4] 이명호, "경량 컨테이너 구조 환경의 스프링 프레임워크 2.5를 기반으로 호텔예약시스템의 설계 및 구현", 한국산학기술학회논문지, 제10권 제3호, pp. 589-595, 2009.
 [5] 이명호, "스프링 2.0과 하이버네이트 3.2를 이용한 경량 컨테이너 구조의 설계 및 구현", 대한설비관리학회지, 제14권 제2호, pp. 111-117, 2009.
 [6] 이명호, "동일한 경량 컨테이너 구조 환경에서 스프링 프레임워크 2.0과 2.5의 개발 생산성 비교 연구", 한국산학기술학회논문지, 제10권 제6호, pp. 1265-1274, 2009.
 [7] 이용원, 박지훈외, "J2EE 응용과 디자인 패턴," 대청, pp. 603- 785, 2002.
 [8] Dan Allen, "SEAM In Action", Manning, pp. 3-136, 2009.
 [9] Jim Farley, "Practical JBoss Seam Projects", Apress, pp. 1-98, 2007.
 [10] Joseph F. Nusairat, "Beginning JBoss Seam:From Novice to Professional", Apress, pp. 1-186, 2007.
 [11] Kito D. Mann, "JavaServer Faces In Action", Manning, pp. 4-184, 2005.
 [12] Michael J. Yuan, Jacob Orshalick, and Thomas Heute, "Seam Framework", Prentice Hall, pp. 3-73, 2009.

이 명 호(Mycong-Ho Lee)

[중심회원]



- 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