

스프링 프레임워크를 활용한 논문 투고 및 심사 시스템의 설계 및 구현

양동선, 한연희¹⁾

한국기술교육대학교 첨단기술연구소 지능네트워크연구소
animato@kut.ac.kr, yhhan@kut.ac.kr

Design and Implementation of a Paper Submission and Review System Using Spring Framework

Dong-sun Yang, Youn-Hee Han
Laboratory of Intelligent Networks,
Advanced Technology Research Center
Korea University of Technology and Education

요 약

최근 웹 기반 엔터프라이즈 애플리케이션들의 규모나 복잡도가 고도화 되어감에 따라 웹 애플리케이션마다 공통적으로 사용되는 기반 요소들의 설계나 구현은 관련 프레임워크를 사용하고 복잡한 비즈니스 로직에 좀 더 집중하는 것이 최근 웹 애플리케이션 개발의 추세이다. 이에 따라 본 논문에서는 대부분의 학회에서 필수적으로 구축하는 논문 편집 시스템을 예로 들어 스프링 프레임워크와 관련 기술 등을 사용한 설계 및 구현 사례를 제시하고자 한다.

1. 서론

최근의 웹 기반 엔터프라이즈 애플리케이션들은 요구사항의 복잡도나 비즈니스 로직의 규모면에서 보다 복잡해지고 고도화되어가고 있다. 이에 따라 대규모 웹 기반 엔터프라이즈 애플리케이션을 개발하는데 있어서 공통적으로 사용되거나 필수적으로 사용되는 기반 요소들을 미리 구현해 놓고 이를 모듈화 하여 만들어진 웹 프레임워크들이 각광받고 있다. 이런 프레임워크들을 사용함으로써 설계를 용이하게 하고 코드의 재사용성을 높이며, 내부 구성 요소들의 모듈화를 추구하고 더 나아가 비즈니스 로직에 역량을 좀 더 집중할 수 있다.

본 논문에서는 많은 학술 학회에서 필수적으로 구축하여 활용하는 논문 투고 및 심사 시스템을 웹 기반 엔터프라이즈 애플리케이션 선정하고 이를 스프링 프레임워크(Spring Framework)에서 설계하고 구현한 사례를 제시하고자 한다.

2. 관련 연구

학회의 학술 정보나 학술 논문들을 웹 기반의 엔터프라이즈 애플리케이션으로 구현하고자 하는 사례는 오래전부터 꾸준히 보고되어 왔다. 최근 들어서는 스프링 프레임워크를 활용한 웹 기반의 애플리케이션의 개발도 다양한 사례가 보고되고 있다. 기존의 스프링 프레임워크를 활용한 구현 사

례는 기본적인 스프링 프레임워크만을 활용한 사례가 많았으나 [3][4], 최근에는 애플리케이션의 각 파트별로 스프링 프레임워크와 연계하여 최종적으로는 시너지 효과를 볼 수 있는 스프링 관련 프레임워크들이 많이 출시되고 있다.

2.1 Spring Framework

스프링 프레임워크(Spring Framework)는 웹 기반 엔터프라이즈 애플리케이션에서 필요로 하는 기능을 제공하는 이른바 “Full Stack Application Framework”이다. 스프링은 JEE(Java Enterprise Edition)가 제공하는 다수의 기능을 지원하고 있어서 JEE를 대체하는 프레임워크로 자리잡았다 [5].

스프링 프레임워크는 자바 객체의 생성, 소멸과 같은 라이프 사이클을 관리하고 객체가 필요할 때 마다 손쉽게 가져와 사용할 수 있도록 설계되었다. 또한 스프링 프레임워크는 Dependency Injection (DI)을 지원하여 객체 간의 의존 관계 설정에 있어서 객체를 직접 생성하거나 검색 할 필요가 없이 설정 파일이나 어노테이션을 통해 객체 관계를 설정 할 수 있다. 또한 POJO (Plain Old Java Object)를 지원하여 스프링 컨테이너에 저장되는 자바 객체들은 특정한 인터페이스에 종속되거나 특정 클래스를 상속하여 만들지 않아도 된다. 이는 기존에 작성한 코드를 수정할 필요 없이 바로 스프링 프레임워크에서 사용할 수 있음을 의미한

1) 한연희: 교신저자

다.

이외에도 스프링 프레임워크는 영속성과 관련된 다양한 API를 제공하여 손쉬운 데이터베이스 처리가 가능하고 JMS (Java Message Service), 메일, 스케줄링 등 엔터프라이즈 어플리케이션을 개발하는데 필요한 다양한 API를 제공하고 있다. 더불어 스프링 프레임워크는 자체적으로 MVC (Model, View, Controller)프레임워크를 제공하고 있어 웹 기반 어플리케이션의 대표적 설계 모델인 MVC 설계를 어렵지 않게 구현해 낼 수 있다.

2.2 Spring Web Flow

웹 기반의 엔터프라이즈 어플리케이션에는 각각의 기능들에 대한 진행이 플로우 형태로 나타나는 경우가 많다. 전통적인 웹 어플리케이션 개발 과정에서는 개발자들이 이러한 웹 플로우를 프로그래밍 적으로 관리했기 때문에 웹 플로우를 관리하거나 재사용하기가 어려웠다. 스프링 프레임워크의 MVC는 특정 프로세스를 플로우 형태로 구현하고 싶을 때 AbstractWizardController와 같이 미리 만들어진 컨트롤러를 사용하거나 보다 복잡한 플로우라면 플로우만을 위한 컨트롤러를 따로 만들어야 했다.

이런 문제점들을 보완하기 위해 나온 개념이 스프링 웹 플로우(Spring Web Flow) [6]이며 이를 이용하면 XML을 사용하여 예를 들어 그림 1과 같은 사용자 계정 관리 웹 플로우를 정의할 수 있다.

```
<?xml version="1.0" encoding="UTF-8"?>
<flow
xmlns="http://www.springframework.org/schema/webflow"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.springframework.org/schem
a/webflow
http://www.springframework.org/schema/webflow/spring-we
bflow-2.0.xsd">

<view-state id="start">
<transition to="CreateAccountName" />
</view-state>
<view-state id="CreateAccountName">
<on-render>
<evaluate expression="accountAction.setupForm" />
</on-render>
<transition on="submitAccountName"
to="CreateAccountAddress">
<evaluate expression="accountAction.bindAndValidate"
/>
</transition>
</view-state>
<view-state id="CreateAccountAddress">
<on-render>
<evaluate expression="accountAction.setupForm" />
</on-render>
<transition on="submitAccountAddress"
to="CreateAccountId">
<evaluate expression="accountAction.bindAndValidate"
/>
</transition>
</view-state>
...
<end-state id="end" />
</flow>
```

그림 1 XML을 사용한 계정 관리 웹 플로우 정의 예

애플리케이션의 표현 계층(Presentation Layer)에서 스프링 웹 플로우를 이용하면 사용자 인터페이스 (User Interface, UI) 플로우를 보다 명확하게 표현 가능하며, UI 플로우별로 모듈화가 가능해져 UI 플로우의 재사용성이 높아진다. 또한 단순히 UI 플로우만을 구현하기 위한 컨트롤러의 작성이 필요 없어지면서 컨트롤러의 레이어 구분이 보다 명확해진다.

2.3 기존 논문 투고 및 제출 시스템

오래 전부터 학회 등의 학술 기관에서는 기관에서 생산되는 학술 정보 및 논문의 관리를 좀 더 효율적으로 하기 위해 웹 기반의 시스템을 구축하여 왔다 [1][2]. 하지만 학술 정보의 관리 시스템의 설계는 학회 특성에 따라 세부 사항이 많이 달라지므로 설계가 쉽지 않고 시스템 구현 시 정책 변경이나 규모 확장 등을 고려하기 어려워 시스템의 유지 보수비용이 많이 들게 된다. 그로인해 최근에는 Scholar One²⁾등의 학술 정보 통합 관리 시스템들을 전문적으로 개발하는 업체들도 많이 생기고 있다. 본 논문에서 구현하는 시스템을 개발하면서 ScholarOne을 많이 참고하였다.

3. 논문 편집 시스템의 설계 및 구현

본 논문에서는 스프링 프레임워크와 스프링 웹 플로우를 활용하여 웹 기반의 논문 편집 시스템을 설계 및 구현하였다. 논문 편집 시스템의 전체적인 요구사항과 처리과정은 한국정보처리학회의 국제 저널인 3)JIPS(Journal of Information Processing Systems)의 과정을 반영하였다.

3.1 시스템의 설계

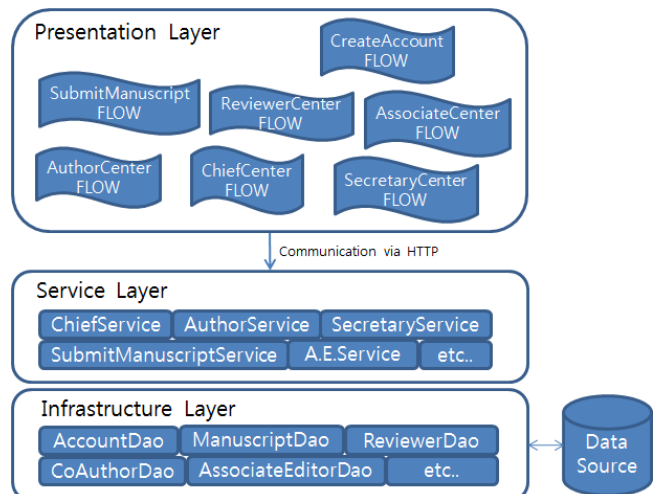


그림 2 시스템 계층의 개략적인 구조

본 시스템은 크게 3단계의 계층으로 구분하여 설계하였다. 첫 번째 계층은 Infrastructure Layer로써 이 계층에

2) <http://scholarone.com/>

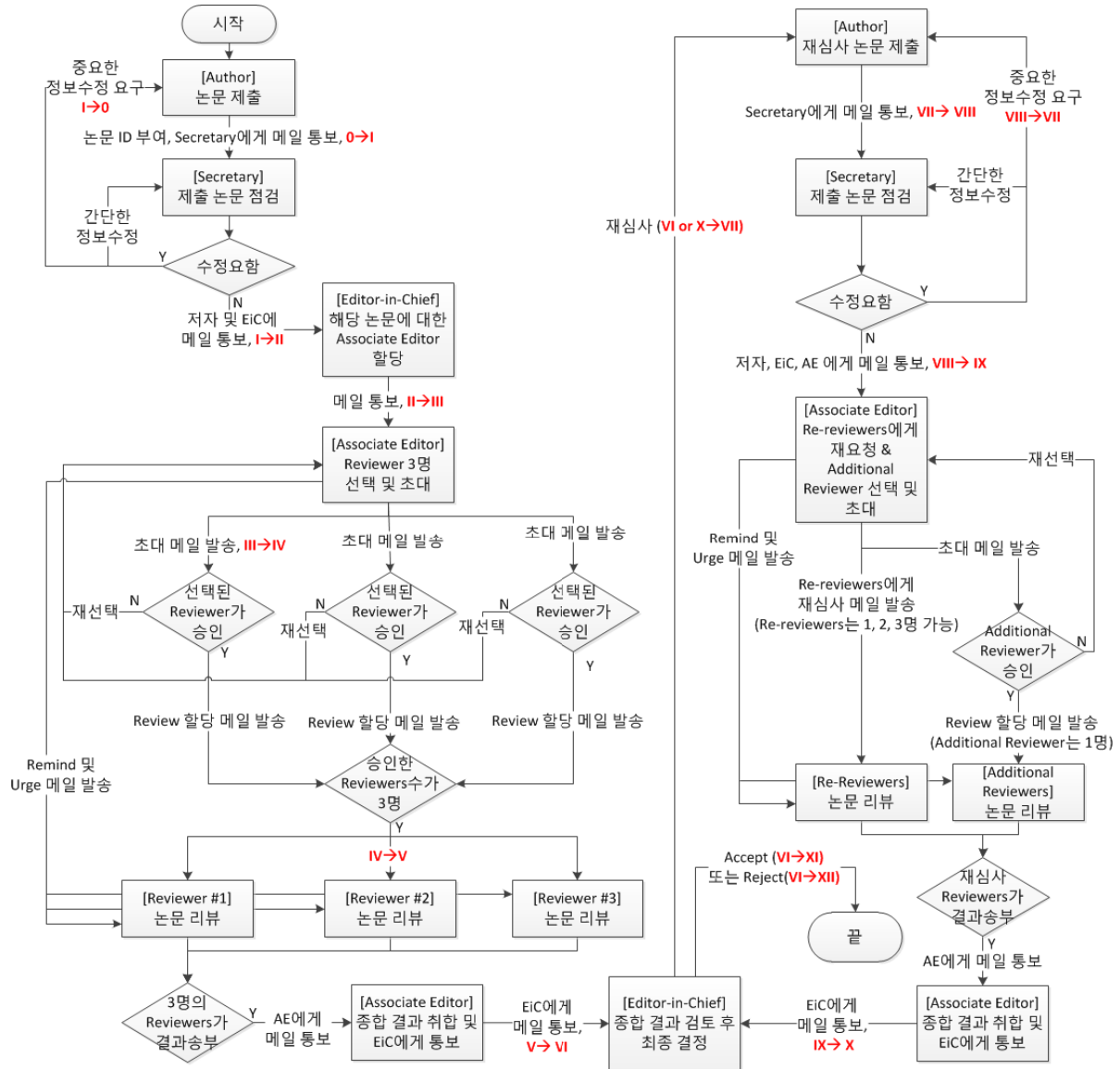


그림 3 논문 편집 시스템의 상세한 흐름도

는 데이터베이스를 직접적으로 다루는 Data Access Object들이나 각종 기반 클래스들이 모여 있어 다른 계층들이 동작하는데 있어 필수적인 기능들을 제공하는 계층이다. 두 번째 계층인 Service Layer는 시스템의 실제적인 요구사항을 처리하는 계층으로써 시스템의 업무 로직을 구현한 컨트롤러를 포함하고 있다. 마지막으로 Presentation Layer는 외부에 노출된 계층으로써 사용자의 요청을 해석하고 그에 맞게 서비스 계층의 컨트롤러들을 호출하는 역할을 한다.

사용자가 외부로 노출된 뷰를 통해 업무 요청을 하게 되면 Presentation Layer에서 이를 해석해 그에 알맞은 서비스를 호출하게 된다. Service Layer에 구성된 각각의 Service들은 요청에 따라 업무를 수행하고 필요에 따라 Infrastructure Layer에 있는 기반 클래스들을 활용하게 된다. 각 계층별로 역할과 구성물들을 명확히 구분함으로

써 시스템 내부 구조의 명확성의 향상은 물론이고 시스템을 구현해 나가는 과정에 일관성이 유지되어 구현이나 유지 보수가 용이해 진다.

3.2 시스템의 구현

본 시스템의 전체적인 흐름은 그림 3에 자세히 도식화하였다. 또한 시스템의 주요 기능들을 표 1에 정리하였다.

Presentation Layer는 각 모듈별로 스프링 웹 플로우를 적용시켜 UI 플로우는 대부분 XML로 작성하였다. 실제로 보여지는 JSP 뷰에서는 SpEL을 통해 자바 객체의 데이터들을 출력하였고 HTML 태그 중에는 간단히 사용할 수 있고 바로 적용 가능한 HTML5 태그도 일부 적용하였다. 화면 꾸밈은 CSS를 적극 도입하여 전체 시스템 화면의 일관성 유지와 향후 꾸밈의 변화에 대비하였으며, 화면과 사용자간의 상호작용을 담당하는 Javascript 부분은 대표적인

3) <http://jips-k.org>

Javascript 프레임워크인 jQuery를 사용하여 코드의 가독성과 안정성 향상을 도모하였다. Service Layer는 POJO 클래스들로 비즈니스 로직이 구현되어 있다. 기반 계층인 Infrastructure Layer에서 데이터베이스를 직접적으로 다루는 DAO(Data Access Object) 들은 SimpleJdbcTemplate을 이용해 구현하였다. 데이터베이스는 MySQL 5.0을 사용하였다.

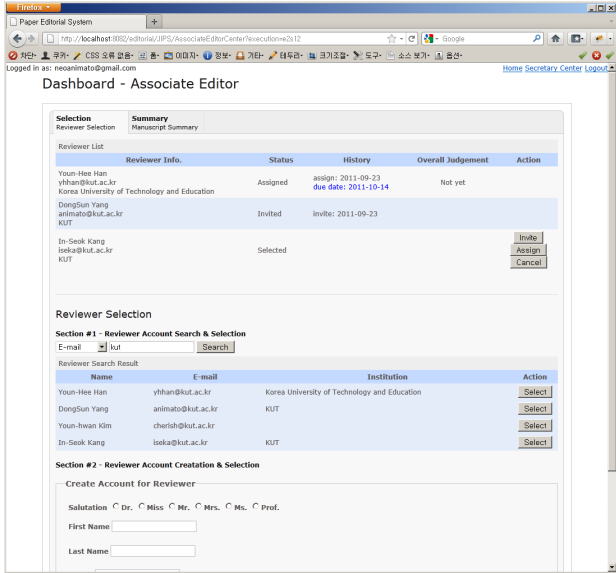


그림 4 부편집자의 심사위원 선정 페이지

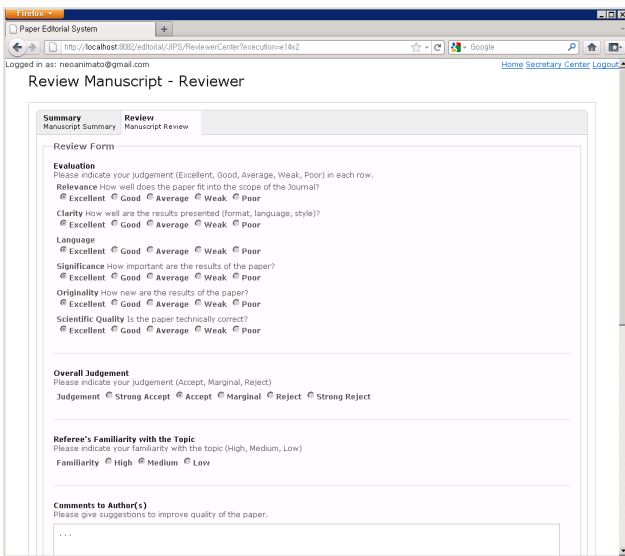


그림 5 Reviewer를 위한 Review 페이지

표 1 논문 편집 시스템의 주요 기능

역할	주요 기능
저자 (Author)	제출할 논문에 대한 정보 기입 및 수정 제출한 논문에 대한 요약 열람 제출 논문에 대한 리뷰 진행 상태, 최종 결정 열람 과거 제출했던 논문 목록 열람

편집비서 (Secretary)	시스템 사용자 계정 관리 시스템 사용자에게 권한 부여 제출된 논문에 대한 정보 열람 및 수정 제출된 논문에 대한 목록, 리뷰 상태 열람 E-mail 템플릿 관리
최고 편집자 (Editor-in-Chief)	제출된 전체 논문에 대한 정보 요약 열람 제출된 각 논문에 대한 부편집자 선정 부편집자로부터 받은 심사 결과 검토 및 최종 결정 제출된 각 논문에 대한 심사 진행 상황 열람
부편집자 (Associate Editor)	할당 받은 각 논문에 대한 정보 열람 할당 받은 논문에 대한 심사위원 검색, 선택 기능 할당 받은 논문에 대한 심사위원 초대 기능 초대 받은 심사위원이 수락 시 논문 심사 할당
심사위원 (Reviewer)	할당 받은 각 논문에 대한 정보 열람 할당 받은 논문에 대한 심사 결과 기입 할당 받은 논문에 대한 심사 결과 전송
기타	각 단계별 진행 상황을 E-mail을 통해 알림

4. 결론 및 향후 계획

본 논문은 스프링 프레임워크와 그 관련 프레임워크들을 사용하여 대규모 웹 기반 엔터프라이즈 애플리케이션인 논문 편집 시스템의 설계 및 구현 사례를 제시하였다. 본 사례를 통해 대규모 웹 기반 애플리케이션 구현에 따르는 설계의 계층화와 효율적인 UI 플로우의 관리가 가능하다는 것을 보였다.

향후에는 본 시스템에 논문 편집 시스템의 사용자인 편집자의 편의를 위한 Reviewer 추천 기능, Review결과 통계 출력 기능들이 추가 구현될 계획이다.

참고문헌

- [1] 박재원, 강무영, 최명석, 정희석, 김병규, 이형진 “대한 아동간호학회 웹기반 논문투고관리 연구 및 발전방안”, 아동간호학회 동계학술대회, page(s): 2-90, 2005.12
- [2] 한종엽, 권성국 “학술논문의 웹기반 투고심사시스템 설계·구현 연구”, 한국정보관리학회, 제13회 한국정보관리학회 학술대회 논문집, page(s): 289-296, 2006.8
- [3] 박윤영, 박해철, 변해원 “스프링 프레임워크를 이용한 사용자 공유 중심의 온라인 서재 시스템 설계 및 구현”, 한국정보과학회, 한국정보과학회 2009 한국컴퓨터종합학술대회 논문집 제36권 제1호(B), page(s): 383-387, 2009.6
- [4] 차지훈, 김태형, 이승하, 김양우 “스프링 (Spring Framework) 기반 HDFS(Hadoop Distributed File System) 연동 인터페이스 설계”, 한국인터넷정보학회, 한국인터넷정보학회 2009 제20차 정기총회 및 추계학술발표대회, page(s): 177-180, 2009.10
- [5] 최범균 저 “웹 개발자를 위한 Spring 3.0 프로그래밍” 가메출판사, 2010
- [6] 고종범 역 “스프링 3 레시피”, 위키북스, 2011