

ASP 소프트웨어 개발을 위한 오픈 프레임워크 확장 모듈 설계

권영오*, 이세훈**

Design of Open Framework Extended Module for ASP Software Development

Young-Oh Kwon *, Se-Hoon Lee **

요 약

이 논문에서는 오픈 프레임워크를 사용하여 ASP를 개발 하는데 적합하도록 향상된 ASP 개발 전용 프레임워크를 제안한다. 시스템은 MVC 개발 방법론과 쿼리 및 ASP정보를 XML로서 관리 하도록 만들어진 프레임워크로서 고객의 요구 사항에 빠르게 대응하는 유연성과 개발자로서는 이미 만들어진 템플릿 소스를 상속받아 구현되어지는 기본 모델과 개발방법만 신경 쓰도록 하여 개발속도의 향상과 시스템의 무결성을 강화 시키는 프레임워크를 제안하였다. 실험결과를 통해 기존에 개발하는 방식들에 비해 향상된 개발 속도와 모델 소스의 양이 기존에 비해 29% 정도만의 라인수를 가졌음을 보여준다.

▶ Keyword : 개방형 프레임워크, ASP, 스프링(Spring), 웹사이트 구축

• 제1저자 : 권영오
* 인하공업대학원 ** 인하공업전문대학

I. 서론

ASP(Application Service Provider)는 다수의 기업을 대상으로 기업의 비즈니스 어플리케이션의 일부 혹은 전부를 네트워크를 통해 임대해 주는 서비스 사업자이다[1]. 즉, 업체들은 값비싼 기업용 소프트웨어를 일일이 구입할 필요가 없이, ASP 사업자의 서버에 접속, 필요한 S/W 및 프로그램 환경 등을 임대 개념으로 사용할 수 있다[1,2].

현재 다양한 기업들은 자신들의 기업에 맞는 ASP를 요구하고 있으며, 이에 대응하여 ASP 사업자들이 등장하고 있다. 하지만, 급조되어 개발되어진 ASP는 고객의 지속적인 요구 사항에 응답하지를 못하거나 서비스의 규모가 확대되어 가며 발생하는 많은 장애에 의해 기존시스템을 유지 못하고 새로 완전히 개발하는 문제가 발생하고 있다[2,3].

ASP를 개발하는 데 있어서 재사용 가능한 컴포넌트를 개발하고 유지하는 시간 및 인적 비용을 절감시키는 프레임워크가 있다면 급변하는 비즈니스 환경에 유연하게 대응하고 안정적인 시스템을 개발, 유지보수 할 수 있는 경쟁력을 제공할 수 있다.

본 논문에선 공통적인 분석과 설계를 기본으로 오픈되어 있는 MVC 아키텍처를 기반으로 하는 웹 어플리케이션 프레임워크를 비교 분석한 기존 연구를 바탕으로 하여 ASP를 구현 하는데 가장 적합한 프레임 워크를 선택하고 수정하여 ASP를 구현하기 유용한 프레임 워크로 설계·구현한다.

ASP 프레임워크는 표1[4]과 같은 정보공학방법론의 8가지의 접근방법에 만족 시킬수 있는 시스템을 개발할 수 있는 프레임워크를 목표로 하고 설계 및 구현 한다.

<표 1> 정보공학방법론의 8가지 접근방법

접근방법	핵심내용
정보전략계획	컴퓨터의 사용을 최고 경영자의 요구와 관점에 연계 시킴
데이터중심 설계	공식적인 데이터관리 및 데이터모델링 강화
공학적 방법의 추구	전통적인 수작업 방법론을 보다 엄격하고 공학적인 기법적용이 가능한 전산화된 방법론으로 교체
최종 사용자 컴퓨팅	최종 사용자 컴퓨팅의 데이터 설계의 임의성과 중복적인 절차의 생성을 방지할 수 있는 기법의 필요성 대두
설계 자동화	설계자의 업무를 보조하고 설계내용을 검증하는 CASE도구
정보처리의 생산성 향상 추구	정보시스템 개발 기간, 예산 및 유지보수의 생산성을 높이기 위한 4세대 언어, 응용 생성기, 프루타이핑 도구, 최종 사용자 지원도구
재사용 할 수	재사용할 수 있는 설계와 코드를 식별하기 위한 기법

있는 설계와 코드	
전문가 시스템	데이터와 규칙으로 구성된 지식베이스에 추론 과정을 적용함으로써 컴퓨터가 인간 전문가를 흉내 내는 시스템

II. 관련연구

이 장에서는 ASP와 소프트웨어 개발 프레임워크의 개념과 특성에 대해 분석하였으며, 오픈 프레임워크를 고찰 하였다.

1) ASP

ASP(Application Service Provider)는 IT장비 및 소프트웨어 등을 원거리 데이터센터에 상주 시키고, 인터넷 및 전용선을 통하여 소프트웨어, IT 인프라, 고객지원서비스를 고객에게 제공하는 개념의 소프트웨어 임대서비스 제공자를 말한다.

기업은 ASP를 도입함으로써 기업은 다양한 효과를 가질 수 있다. 표 2[3]는 ASP를 도입할 때의 효과를 나타낸 것이다.

<표 2> ASP 도입 효과

비용절감	초기 시스템 구축비용과 향후 유지보수 비용이 저렴, 운영인력 및 관리비용, 신기술 도입비용 부담이 적음
별도의 전산인력 불필요	프로그램 운영을 위해 별도의 전산 전문가를 고용할 필요가 없으며, 대부분 서비스가 범용이어서 사용법이 간단하다.
비즈니스 환경변화에 유연	서비스 회사에서 지속적으로 신기술로 프로그램을 업그레이드 해주므로 이용자가 비즈니스 환경변화에 빠르게 대응
주력업무에 핵심 역량 집중	전산실이나 기타 시스템 구축 및 운영에 필요 없어 회사 고유의 업무에만 집중할 수 있음
움직이는 사무실	인터넷을 사용할 수 있는 곳이면 어디든지 회사의 업무 처리
위험부담 감소	사용할 프로그램을 미리 테스트해 본 뒤 결정할 수 있고 회사에 맞지 않는다고 판단되면 초기에 다른 서비스로 바꿀 수 있어 실패 확률이 적음

2) 소프트웨어 개발 프레임워크

프레임워크(Framework)는 어플리케이션의 공통적인 부분을 따로 분리하여 재사용이 가능하게 만든 semi-complete application이다. 결국, 프레임워크는 소프트웨어의 공통적인 부분을 재사용하여 개발하고자 하는 소프트웨어의 안전성과 생산성을 높여 개발 할 수 있도록 해준다. 현재 다수의 프레임워크가 등장하면서 프레임워크에 대한 개념도 다양한 관점에서 정의되고 있다. 이러한 상황에서 만들어진 프레임워크에 대한 다양한 정의를 다음과 같이 정의 할 수 있다.

- 소프트웨어의 특별한 class로 이루어진 재사용 가능한 디자인 형식으로 협력적인 class들이다[5].
- 프레임워크는 추상클래스와 인터페이스로 기술하여 시스템의 전부 혹은 일부분에서 재사용 가능한 디자인이다[6].
- 어플리케이션 개발자들에 의해 개발되어진 프레임워크는 어플리케이션의 skeleton이다.
- 도메인 안의 어플리케이션이 제공하는 연장 템플릿의 구조 패턴[7].

- 코드의 재사용
- 신뢰성이 있고 검증된 코드를 사용하여 소프트웨어의 질을 향상
- 개발 기간의 단축으로 인한 비용감축
- 표준화된 infrastructure와 architecture guidance를 제공
- 잘 정리된 프레임워크 문서는 개발자와 유지보수자의 개발을 쉽게 함
- 핵심기술이 프레임워크로 이동한 경우 높은 수준의 개발자가 필요하지 않음

프레임워크를 이용하여 어플리케이션을 개발할 때의 효과는 정량적으로 계산하기 힘들다. 그러기 때문에 효과를 알기 위해서는 실질적인 자료를 통해 검증해야 한다. 표 3[8]은 Maryland University에서 조사한 자료로서, 이탈리아 통신 업체인 Telecom Italia S.P.A의 솔루션 개발 프로젝트에 프레임워크 기반에 의하여 개발 하였을 때, 소프트웨어 개발에 대한 효과를 투입시간에 따른 소스코드의 양을 통해 정량적으로 측정한 자료이다. 이 프로젝트는 과거에 생산되었던 솔루션을 리뉴얼 하면서 추가적으로 모듈을 추가하여 개발하였다. 그중에서도 MCS와 VOD는 기존에 있었지만 새로이 개발한 모듈이다. 모듈MCS는 Muliconference의 약자로 화상회의 모듈이다. VOD는 Video on demand 의 약자로 동영상 플레이 모듈이다. MCS과 VOD에 Framework를 적용하기 전과 후를 단위 시간당 코드의 라인을 비교하여 보여주고 있다.

FW는 프레임워크의 투입여부를 나타내고, Effort는 개발자의 투입시간을 나타낸다. LOC는 Line Of Code의 약자로 라인, LOC/H는 시간당 코드 라인생산을 의미한다.

표3에서 보면 프레임워크는 개발할 때 상당한 개발자의 투입시간이 필요하지만, 프레임워크가 적용되어 개발되어진 MCS과 VOD모듈은 프레임워크가 적용되기 전에 개발되어진 모듈과 비교 하였을 때 효율성은 7.3, 7.6 배 만큼 향상된 것을 알 수 있다.

<표 3> 프레임워크기반 소프트웨어 개발 효과

Service	FW	Lang	Effort	LOC	LOC/H
MCS	N	C++	400	15430	38.58
VOD	N	C++	240	11350	47.29
Framework	Y	JAVA	2720	9818	3.61
MCS	Y	JAVA	63	12439	276.42
VOD	Y	JAVA	34	12231	359.74

프레임워크는 다음과 같은 이점을 가지고 소프트웨어의 결과 생산성을 향상 시키며, 개발시간과 비용을 감소시킬 수 있다.

현재 프레임워크는 각자의 목적을 가지고 다양하게 등장하였다. 그 중에는 오픈 되어 사용자들에게 무료로 제공 되어지는 프레임워크도 등장하게 되었다. 이 논문에서는 그 중에서도 현재 가장 널리 사용되며 ASP 구축에 적합하다고 판단한 MVC웹 프레임워크를 중심으로 설명한다.

(1) 스트러츠2

Struts[9]는 MVC 프레임워크 중에서 가장 널리 쓰이는 프레임워크중 하나이다.

이과치 프로젝트에서 만들어졌고 지속되어 개발되어진 오픈 소스 프로젝트이다. 2000년에 등장하여 MVC 프레임워크의 표준이 되다시피 되었다. Struts1 개발자들은 OpenSymphony의 WebWork 개발자들과 진보된 MVC 프레임워크를 위해 아이디어를 모았다. 마침내 두 프로젝트가 합병되어 Struts2가 발표되었다. Struts2는 WebWork2와 합병되면서 내부아키텍처가 완전히 달라졌다. 서블릿 방식에서 필터와 인터셉터 기반의 WebWork2의 아키텍처를 그대로 적용함으로써 유연한 프레임워크로 발전하게 되었다.

(2) 스프링

Spring[10]은 엔터프라이즈 어플리케이션에서 필요로 하는 기능을 제공하는 프레임워크이다. 스프링은 J2EE가 제공하는 다수의 기능을 지원하고 있기 때문에, J2EE를 대체하는 프레임워크로 자리 잡고 있다. MVC 프레임워크를 제공하며, Struts, Webwork2, JSF, 하이베네이트와 같은 프레임워크와의 연동을 지원한다[11].

기업용 비즈니스 시스템을 구축 하는 것은 보통일이 아니다. 개발자들은 복잡도를 낮추기 위해 MVC 패턴이 녹아있는 n-계층 클라이언트/서버 환경을 구성하기 시작했으며, 점차 시간이 흘러가면서 대규모 웹 어플리케이션은 표 4의 다섯 가지 계층으로 일반화되어 적용되고 있다[12].

<표 4> 웹 어플리케이션의 계층

지원계층	역할 및 기능
프리젠테이션 계층	사용자 인터페이스를 말한다. 사용자가 선택할수 있는 기능들이 표시되어야 하고, 요청에 필요한 부가적인 정보 전달을 위한 입력 양식이 있어야 한다
제어 계층	제어계층은 프리젠테이션 계층과 비즈니스 로직 계층을 분리하기 위한 컨트롤러를 제공한다. 전체 시스템의 설정 상태를 유지해야 하며, 그를 통해 어떤 요청이 들어왔을 때 어떤 로직이 처리해야 하는지를 결정한다.
비즈니스로직 계층	핵심 업무를 어떻게 처리하는지에 대한 방법을 기술하는 곳이다. 핵심업무 로직의 구현과 그에 관련된 데이터의 적합성 검증 외에도 다양한 부가적인 구현이 추가된다.
퍼시스턴스 계층	데이터를 처리하는 계층이다. 관계형 정보를 저장하고 수정·삭제하는것과, 그러한 일을 수행 하는데 필요한 질의문을 관리하는 것, 그리고 가져온 관계형 정보를 객체화 시키는 일이다.
도메인 모델 계층	각 계층 사이에 전달되는 실질적인 비즈니스 객체라고 할 수 있다.

표 5는 오픈 프레임워크를 이용하여 개발할 시에 지원되는 부분을 표시 한 것이다. O는 지원되는 부분을 나타 낸 것이고, X는 현재 지원되는 않는 계층이다.

<표 5> 오픈 프레임워크 지원 계층

지원계층	Struts2	Spring
프리젠테이션 계층	O	O
제어 계층	O	O
비즈니스 로직 계층	O	O
퍼시스턴스 계층	X	O
도메인 모델 계층	X	O

3) MVC 구조

MVC(Model-View-Control) 구조는 객체 관계에 있어서 세 가지 종류의 컴포넌트로 나누어진다. 프로그램 상태에 대한 논리적인 표현을 하고, 데이터가 변경 되었을 때 View에 통보하는 모델(Model), 변하는 데이터에 대한 시각적인 표현을 제공하는 뷰(View), 사용자의 입력을 받아 작동하고 사용자와의 상호작용을 하는 컨트롤러(Controller)의 세 가지 객체로 정의된다.

웹 어플리케이션에서의 MVC 구조로 개발을 할 때 사용하는 구조는 Model 1과 Model 2가 있다. 그중에서도 Model 1 구

조를 가장 많이 사용하고 있으며, 또한 가장 쉽기도 하다[13].

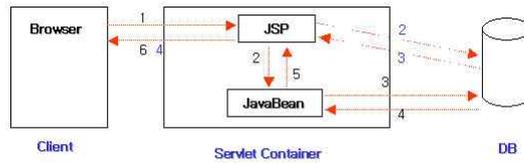


그림 1 Model 1 구조

Model 1의 접근방식은 프리젠테이션이 비즈니스 로직을 처리하는 혼합된 결과물을 생산해냈고, 곧 유지보수와 어플리케이션 확장에 한계점을 드러내었다. 이러한 문제점을 해결하기 위해 Model 2의 접근 방식이 개발되었다. Model 2는 비즈니스 로직과 프리젠테이션의 분리로 인해 어플리케이션이 명료해지며 유지보수와 확장이 용이하며, 디자이너와의 작업이 분리되어져 책임의 소재가 명확해졌다[13].

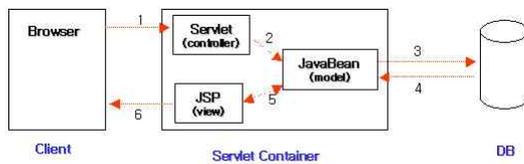


그림 2 Model 2 구조

4) 스프링

스프링 프로젝트는 로드존슨(Rod Johnson)의 책에서 소개된 소스 코드를 기반으로 2003년 2월에 오픈 소스로 시작 되었으며, 크게 다음 두 가지를 추구하고 있다.

- 복잡하고 무거운 J2EE 기술의 사용을 쉽고 가볍게 만들어 주고, 자연스럽게 검증된 최상의 실천 사례들을 구현하도록 함으로써 좋은 프로그램이 작성될 수 있도록 유도한다.
- 기존의 잘 알려진 기술들을 프레임워크 내에서 일관된 방법으로 쉽게 사용할 수 있도록 돕는다[12].

이와 같은 조건들을 만족하기 위하여 개발 되어진 스프링은 다음과 같은 특징들을 가지게 되었다.

- EJB의 사용 유무에 상관없이 비즈니스 객체들을 효과적으로 구성하고, 관리하는 방법을 제공한다.
- 계층화된 아키텍처를 가지고 있으며, 독립적으로 사용이 가능하도록 모듈화 되어있다.

- 디를 이용한 IoC 컨테이너로서의 기능을 가지고 있다. 이는 프로젝트의 설정을 일관된 방법으로 관리 할 수 있도록 한다.
- 스프링 기반으로 작성된 어플리케이션은 스프링의 API에 의존하지않는다.
- AOP 지원을 통하여 트랜잭션, 로깅, 보안과 같은 모듈들을 완벽히 분리 해내도록 도와준다.
- 작성된 코드에 대한 유닛 테스트를 쉽게 할 수 있다.

추가되는 모듈로서 스프링 프레임워크에 추가시 ASP 개발에 빠르게 적용 될수 있도록 해준다.

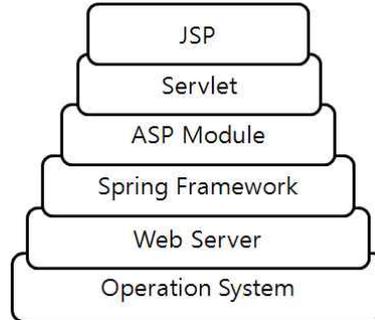


그림 4 모듈 구성 Layer

III. 시스템 설계 및 구현

이 장에서는 이미 검증되어 많이 사용하는 스프링 프레임워크를 개선하여 ASP 프레임워크로 확장이 가능 하도록 하는 모듈을 설계 및 구현 할 것이다.

1) ASP 모듈 구성

스프링 프레임워크는 MVC뿐만이 아니라 퍼시스턴트 계층도 지원해주는 프레임워크이다. 하지만 ASP 개발에 최적화 되어 있는 프레임워크라고는 할 수는 없다. 그래서 안정성과 확장성을 검증받은 스프링 프레임워크를 이용하여 ASP 전용 프레임워크로 확장 하여 개발 하였다.

추가된 프로세스의 주 목적은 ASP 개발시 코드의 양을 최소화 하고 빠르게 유지보수 하고 시스템의 확장성이 높도록 하는 것에 중점을 두었다.

그림3의 구성도와 같이 도메인정보, 모델과 뷰 매핑정보, SQL등을 XML로 분리하여 시스템의 재 컴파일 없이 고객의 요구사항을 빠르게 대응할 수 있도록 하였다.

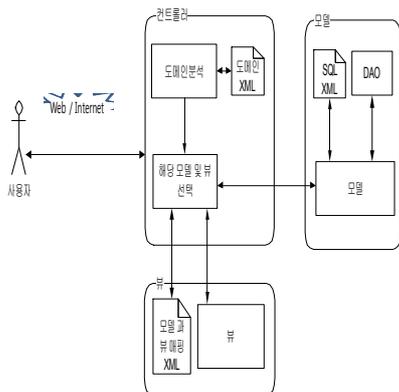


그림 3 ASP 모듈 구성도

구성도대로 개발되어진 ASP 모듈은 스프링 프레임워크에

2) 프로세스 구조

ASP는 다양한 사용자의 요청 사항을 소화해야 한다. 하지만 사용자들은 저마다 요구사항이 틀리다 할 수 있다. 그래서 다양한 사용자들에게 서비스를 할 수 있도록 사용자를 구분할 수 있어야 한다. 이를 해결하기 위해 ASP 시스템에서는 도메인을 이용하여 사용자를 구분한다. ASP 프레임워크는 도메인 분석 프로세스를 추가 하여 사용자를 구분 할 수 있도록 할 것이다.

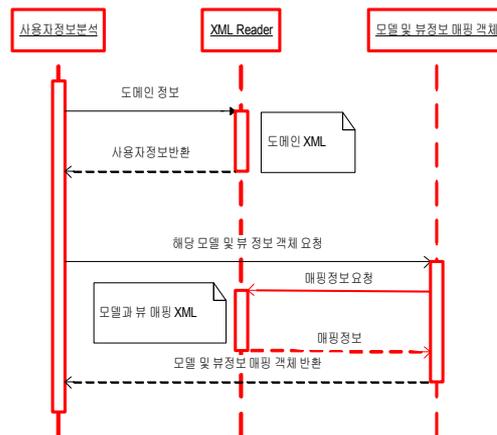


그림 5 도메인 분석 프로세스

사용자가 해당 URL을 가지고 시스템에 접속을 하면, 시스템은 도메인 정보가 저장되어 있는 도메인 XML에서 해당

사용자 정보를 반환한다. 반환된 정보를 가지고 모델 및 뷰 정보 매핑 객체를 생성 시킨다. 생성된 모델 및 뷰정보 매핑 객체는 사용자에게 맞는 서비스를 제공 할것이다.

ASP는 사용자의 요구 사항에 따라 DB의 구조변경 조건 변경이 잦을 수밖에 없다. 하지만 이를 수정하기 위해 코드상에서 SQL등을 수정하고 재 컴파일을 한다면 고객에게 보다 늦은 대응을 할 수밖에 없을 것이다. 그래서 ASP 프레임워크에서는 SQL을 XML로 분리하여 사용자에게 대한 대응을 보다 빠르게 할 것이다.

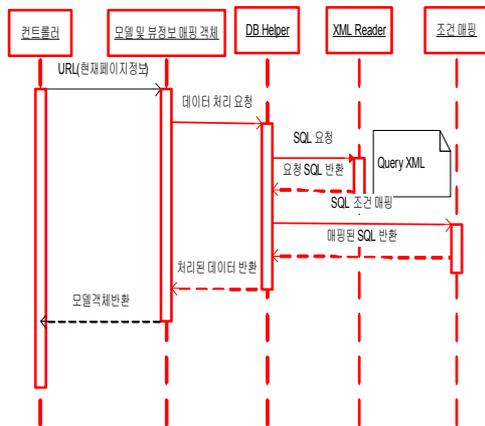


그림 6 모델 및 뷰 연동 프로세스

현재 페이지의 URL 정보를 가진 모델 및 뷰정보 매핑 객체는 지정된 XML에서 해당 SQL을 읽고 조건을 매핑하여 데이터를 처리한다. 그리고 처리된 결과를 가진 해당 모델을 반환한다. 모델에는 이미 뷰의 정보도 포함되어 있어야 한다.

소스 코드를 구현 하기전에 각 정보를 XML로 보관하였다. domain.xml 은 asp에 해당하는 도메인을 관리 하는 xml이다. view.xml은 뷰와 보털을 연계 하여주는 XML이다.

<코드 1> domain.xml

```
<?xml version="1.0" encoding="utf-8"?>
<statements>
<domain name="www.test.co.kr">
asp.test.model.TestModelController
</domain>
</statements>
```

domain.xml에서 name의 값은 유일해야 한다. name 아 큐먼트는 해당 도메인을 나타낸다. domain 태그의 값은 해당 도메인을 컨트롤해주는 모델 컨트롤러이다. ASP 프레임 워크는 설정해주 모델 컨트롤러를 통하여 비즈니스 로직을 처리하여 준다.

<코드 2> view.xml

```
<?xml version="1.0" encoding="utf-8"?>
<datas>
<data id="/ASPframework/boardList.htm">
<model>getBoardListModel</model>
<view>boardList</view>
</data>
</datas>
```

<코드 3> query 관리 xml

```
<?xml version="1.0" encoding="utf-8"?>
<statements>
<statement name="test">
select test_nm from test_t
</statement>
<statement name="getBoard">
select b.board_seq board_seq,
b.boardTitle boardTitle,
b.boardData boardData,
b.insert_dt insert_dt
from domain_t a,
board_t b
where a.domain_nm = ?
and b.domain_id = a.domain_id
and b.board_id = ?
order by b.board_seq
</statement>
</statements>
```

view.xml은 모델과 뷰를 연계의 값을 관리 하여주는 xml이다. 이 정보는 각 도메인에서도 따로 관리가 되며 전체적으로 뷰어 관리가 되어진다. id값은 유일해야 하며 id값은 해당 uri를 나타낸다. model 태그안의 값은 도메인에서 설정한 모델컨트롤러의 메소드를 나타낸다. view 태그는 메소드에서 연계 시켜주는 jsp의 이름이다.

query를 관리 하여주는 태그도 관리하여 준다. xml로 query를 관리 하여 줌으로서 데이터의 유연성 있는 관리가 가능하여진다.

스프링의 기능을 확장한 ASP프레임워크는 ASPController를 이용하여 구조도에 해당하는 프로세스를 처리한다.

ASPController는 1단계로 도메인을 분석하여 도메인에 해당하는 모델컨트롤러정보를 가져와 2단계인 Model Mapping 객체를 생성 시키고 필요객체를 넘겨준다. 마지막 단계로 생성된 모델컨트롤러를 통해 해당 모델 객체를 넘겨받아 스프링 컨트롤러에 넘겨준다. 개발자는 시스템 개발시에 이부분을 신경 쓸 필요가 없어진다.

<코드 4> 모델과 view 연계

```
public ModelAndView
    getHelloModel(String view){
    HashMap[] datas =
    this.getHashValue("test", null, null);
    ModelAndView mav =
    new ModelAndView();
    mav.setViewName(view);
    mav.addObject("datas",datas);
    return mav;
}
```

개발자는 ModelController 객체를 상속받아 표10과 같이 개발하면 해당 데이터 및 뷰를 연동할 수 있다.

IV. 실험 및 평가

이 장에서는 개발 되어진 프레임워크를 통하여 ASP 형식의 게시판 개발을 실험을 하였다. 실험들을 통하여 뷰의 적용시간, 프레임워크의 적용시 소스 라인수, 유지보수시의 변경 파일수 및 변경 라인수를 측정해 보았다

1) 실험 방법

이 논문에서는 가상 도메인을 추가시키고 가상 도메인에 해당하는 게시판을 개발하여 기존 개발방법에 비해 개선된 점을 비교 평가 하였다. 프레임워크를 통하여 개발되어진 게시판은 각기 다른 도메인과 다른 uri를 가지고 있다. 각기 다른 도메인을 가지고 실험을 하기 위하여 윈도우의 system32\drivers\etc\hosts 파일에 url을 추가 하여 주었다. 다음은 도메인을 설정 하였다.

<코드 5> hosts 추가 도메인

127.0.0.1	www.test.co.kr
127.0.0.1	www.test2.co.kr
127.0.0.1	www.test3.co.kr

설정된 도메인으로 개발되어진 각각의 게시판들은 실험 평가에서 사용되어졌다.

2) 실험 평가

첫 번째 실험은 기존에 개발 되어진 모듈을 통하여 새로운 도메인과 뷰를 적용 하였을때의 적용 시간을 측정하기 위한 실험을 하였다.

게시판은 ModelController를 상속받은 TestModelController를 통해 게시판에서 사용할 비즈니스 로직인 모델을 개발 하였다. TestModelController는 domain.xml에서 설정하여 주었다. 모델을 설정하고 뷰단의 jsp파일들을 개발하여 도메인에 따라 서로 다른 디자인이 적용되는것을 테스트 하였다.

그리고 각 도메인에 해당되게 개발되어진 게시판은 그림 7 과 그림 8이다.



그림 7 www.test.co.kr 게시판



그림 8 www.test2.co.kr 게시판

프레임워크를 통하여 게시판을 개발 할 때에 별다른 문제 점은 발견 되지 않았으며 다음과 같이 개발 속도가 매우 향상된 것을 알 수 있었다.

Service	minutes
www.test.co.kr 게시판	32
www.test2.co.kr 게시판	10

서로 다른 도메인을 가진 게시판을 개발 하는데 있어서 두 번째로 개발되어진 게시판은 현재 개발한 프레임워크로 개발 할 경우 첫 번째로 개발한 게시판에 비해 약 3배정도 빠르게 개발이 되어졌다. 개발할 시에 query를 약간 수정한 시간과

지 생각한다면 매우 많이 단축 졌다는 것을 알 수 있다.

두 번째 실험으로서 기존에 프레임워크를 이용하지 않고 개발 되어진 게시판을 프레임워크를 사용하여 재개발을 하여 보았다. 개발되어진 게시판의 리스트 모델부분의 소스를 비교 하면 표7과 같이 나타난다.



그림 9 프레임워크 이용 개발 게시판

<표 7> 프레임워크 적용 비교 표

프레임워크 적용	Count of Line
N	131
Y	38

실험결과에서와 같이 프레임워크 적용시 기존 소스코드의 라인수에 비해 29%정도의 소스가 줄어드는 것을 확인 할 수 있었다.

세 번째 실험으로서 게시판 리스트에 추가적인 컬럼을 추가 하면서 변경 되어지는 소스를 측정해 보았다. 기존의 방식은 비즈니스 로직에서 모델의 소스 파일만을 수정하고 재 컴파일을 해야지만 뷰파일에서 적용할 수가 있었다. 반면에 개발되어진 프레임워크에서는 Query가 저장되어 있는 xml 파일과 뷰 파일만을 수정하면 간단하게 적용 할 수 있었다.

<표 8> 유지보수시 수정사항 비교

프레임워크 적용	재 컴파일	수정 파일수
N	Y	3
Y	N	2

V. 결론 및 향후 연구

ASP 시스템을 개발하기 위해서는 ASP 개념부터 시작해서 ASP 적합한 시스템 구조를 설계를 해야만 한다. 설계를 한후 시스템을 구현할 때에 초기 작업은 많은 시간과 노력을 소비 할 수밖에 없을 것이다. 하지만 본 논문에서 구현되

어진 방식으로 프레임워크를 구현하여 사용한다면 설계의 시간뿐만이 아니라 개발의 시간도 단축되어져서 경쟁력 있는 제품 개발에 좀 더 유리한 고지에 위치할 것이다.

하지만 현재 설계 되어진 부분만으로 ASP의 모든 부분을 소화 했다고 하기에는 많이 미흡할 것이다. 좀 더 많은 기능으로서 추가 적인 부분들이 개발되어져야 할 것이다. 그리고 xml과 디렉토리 구조를 관리 하여주는 관리 프로그램으로 현 시스템을 보완하여야 할 것이다.

참고문헌

- [1] 임춘성, 양정환, ASP 산업의 이해와 전망, 지역정보화, 2000
- [2] 민경훈, Java를 이용한 ASP 방식의 회계정보 시스템 구현, 서울산업대학교 석사논문, 2006
- [3] 중소기업 정보화 길잡이, 정보통신부, 2003
- [4] 문대원, 선우종성, 구자환, 박종문, 신신애, 이병욱 시스템 개발방법론 적용기준에 관한 연구, 한국전산원, 1997
- [5] E. Gamma, R. Helm, R. Johnson, and J. Vlissades. Design Patterns: Elements of Reusable Software Architecture. Addison-Wesley, 1995
- [6] M. Fayad, D. Schmidt, R. Johnson, Building Application Frameworks, John Wiley & Sons, Inc, 1999
- [7] Booch, The Unified Modeling Language User Guide, Addison-Wesley, 1999
- [8] Maurizio Morisio, etc., Framework Based “Software Development: Learning as an InvestmentFactor”, First Workshop on Economics-Driven Software Engineering Research, EDSE-1, 1999
- [9] Struts: Open Source Presentation Framework, <http://jakarta.apache.org/struts/>
- [10] Spring: Open Source IOC Framework, <http://www.springframework.org/>
- [11] 최범균, “웹 개발자를 위한 스프링2.5 프로그래밍”, 2008
- [12] 강경수, “결합도가 약한 아키텍처를 위한 대안 기술, 스프링”, 마이크로소프트웨어, 2004
- [13] Malcolm Davis, Struts, 오픈 소스 MVC 구현, <http://www-128.ibm.com/developerworks/kr/library/j-struts/>