

J2EE 기반의 웹 애플리케이션을 위한 프리젠테이션 계층 자동생성 템플릿 개발

(Development of Template for Automatic Generation of
Presentation Layer in J2EE-Based Web Applications)

유철중[†] 채정화^{**} 김송주^{**} 장옥배^{***}
(Cheol Jung Yoo) (Jung Hwa Chae) (Song Ju Kim) (Ok Bae Chang)

요약 J2EE(Java™ 2 Platform, Enterprise Edition) 기반의 웹 애플리케이션은 기존의 애플리케이션이 갖는 시간적·공간적 제약을 극복하는 대안으로 대두되었다. 최근 웹 애플리케이션을 보다 빠르고 효율적으로 개발하기 위한 해결책으로 프레임워크를 이용하는 방안들이 다양하게 제기되고 있다. 본 논문에서는 계층구조를 갖는 웹 애플리케이션을 각 계층의 개발자들이 따로 프로그램을 개발할 수 있다는 점에 착안하여 프리젠테이션 계층에서 담당해야 할 여러 처리와 그 유형을 템플릿으로 제시하여 웹 애플리케이션의 프리젠테이션 계층을 보다 빠르게 개발할 수 있도록 한다. 이 템플릿은 사용자가 작성하고자 하는 웹 애플리케이션의 프리젠테이션 계층에 대한 정보를 나타내는 것으로 XML 형태의 문서이며, 이것을 입력받은 코드 생성기는 사용자 정보를 파싱한 후 주어진 웹 애플리케이션 프레임워크를 바탕으로 프리젠테이션 계층의 골격 코드를 자동 생성한다. 이것은 코드 생성기를 이용하여 프레임워크의 핫스팟 클래스들을 상속 받은 골격 코드를 빠르게 생성할 수 있어 보다 효율적으로 웹 애플리케이션을 개발할 수 있음을 의미한다. 본 템플릿과 코드 생성기를 사용함으로써 개발자는 적은 학습량으로 웹 애플리케이션을 개발할 수 있고, 표준화시킨 개발 과정을 내포함으로써 다른 개발자와의 협력작업을 용이하게 하여 웹 애플리케이션을 적시에 개발할 수 있다.

키워드 : 프리젠테이션 계층, 템플릿, 웹 애플리케이션

Abstract Web applications based on J2EE(Java™ 2 Platform, Enterprise Edition) were occurred for solution to overcome the limitations in time and space that the former applications had. Recently, lots of solutions using frameworks are being suggested to develop applications more quickly and efficiently. In this paper, we propose the template for several processes and types, which should be taken in presentation layer of web applications. This idea was based on the fact that web applications developers can concentrate on their specific tasks with independent manner in layered architecture. This template is XML-typed document that shows information about presentation layer of Web applications, which the user wants to compose. This template is inputted to the code generator. After then, the code generator generates skeleton code in presentation layer automatically after parsing information of XML documents. It means that we can develop Web applications more efficiently, by constructing skeleton code which inherits from hot spot classes of framework. Using this template and code generator, developer can develop Web applications with little practice and also is easy to cooperate with other developers to develop them just in time with distributing the standard development process.

Key words : presentation layer, template, Web applications

· 본 논문은 정보통신부 정보통신연구진흥원에서 지원하고 있는 대학기
초연구지원사업(2001-168-2)의 연구결과임

† 종신회원 : 전북대학교 컴퓨터과학과 교수
cjyoo@moak.chonbuk.ac.kr

** 비회원 : 전북대학교 전산통계학과
jhchae@cs.chonbuk.ac.kr

songju@cs.chonbuk.ac.kr

*** 종신회원 : 전북대학교 전자정보공학부 교수
okjang@moak.chonbuk.ac.kr

논문접수 : 2002년 7월 19일
심사완료 : 2002년 11월 26일

1. 서론

웹은 정보를 확산시키는 매체일뿐만 아니라 애플리케이션을 배치하기 위한 플랫폼이기도 하다[1]. 최근 웹 기반 정보 시스템의 급속한 발전은 정보 시스템 전문가의 개발 기술에 대해 중대한 문제를 제시하였다. 기존의 시스템들은 단순하고 대부분 독립형(stand-alone) 애플리케이션이었으나 최근의 애플리케이션은 광범위하고 복잡하며 인터넷이 IT 플랫폼으로 사용되는 전자상거래(e-Commerce) 애플리케이션의 형태로 바뀌었다[2].

J2EE 기반의 웹 애플리케이션은 기존의 애플리케이션이 갖는 시간적·공간적 제약을 극복하는 대안으로 등장하였다[3]. 이러한 웹 기반 정보 시스템에서 기업과 사용자의 요구는 매우 다양하고 빠르게 변화하는데, 이러한 요구를 신속하게 수용하여 고품질의 웹 애플리케이션을 개발하기 위해서는 다양한 웹 기반 기술을 갖춘 고도로 숙련된 IT 전문가가 필수적이다[4].

그러나 실제적으로는 웹 애플리케이션의 개발 과정은 표준화된 기술보다는 개인의 지식이나 개발 습관에 의존적이다[5]. 또한 J2EE 기반의 웹 애플리케이션 구축을 위해서는 개발자가 익혀야 할 명세(specification)와 기술이 너무 방대하고 다뤄야 할 코드가 많다는 점이 어려운 점 중에 하나로 인식되고 있다. 이는 웹 애플리케이션은 단순한 프로그램 차원이 아닌 플랫폼에서부터 애플리케이션과 웹으로까지 이르는 방대하고 복잡한 아키텍처를 형성하고 있기 때문이다. 이렇게 엔터프라이즈급 웹 애플리케이션의 개발 환경뿐만 아니라 요구조건이 다양한 상황에서 웹 애플리케이션을 보다 빠르고 쉽게 개발하기 위한 해결책으로 프레임워크를 이용하는 방안들이 최근 다양하게 제기되고 있다[6].

본 논문은 웹 애플리케이션의 구성 요소 중 하나인 프리젠테이션 계층의 구성을 템플릿화 하여 개발자가 웹 애플리케이션을 보다 빠르고 쉽게 개발할 수 있도록 표준적인 방안을 제시한다. 본 논문의 나머지 부분은 다음과 같이 구성된다. 2장은 관련연구를 알아보고, 3장은 템플릿을 설계한다. 그리고 4장은 코드 생성기에 대해 구현한 것과 프리젠테이션 계층의 골격 코드(skeleton code)를 생성하는 사례를 제시하고, 5장에서는 기존의 프레임워크와의 비교를 통해 성능 평가를 한다. 마지막 6장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련연구

현재의 웹 애플리케이션은 소규모의 비교적 단기간 동안 서비스를 제공하는 것에서부터 인터넷뿐만 아니라

인트라넷 및 엑스트라넷에 해당되는 대규모의 엔터프라이즈 애플리케이션까지 그 세부 정의와 범위가 매우 다양하다. 그러나 클라이언트와 상호작용하는 하부 프로토콜이 HTTP 프로토콜이고 서버에 비즈니스 로직을 가지는 것을 일반적으로 웹 애플리케이션이라고 정의하며, 단순한 웹사이트 혹은 비즈니스 로직을 포함하지 않는 홍보성 홈페이지 등은 그 범주에서 제외된다[7].

J2EE 플랫폼은 계층적 형태로 구성되며 J2EE 플랫폼 기반에서 작성되는 웹 애플리케이션은 일반적으로 n-계층의 구조를 갖는다[3]. 이러한 n-계층 구조는 사용자 인터페이스, 프리젠테이션 로직, 비즈니스 로직과 하부 구조 서비스 및 데이터 계층으로 분리되는데, 이것은 물리적이라기보다는 논리적 즉, 기능적으로 분리되는 구조이다[7]. J2EE 플랫폼 기반의 웹 애플리케이션은 계층적 아키텍처 상에서 그림 1과 같이 하드웨어와 운영체제 및 J2EE 플랫폼 계층 등의 상위 계층에 위치하게 된다.

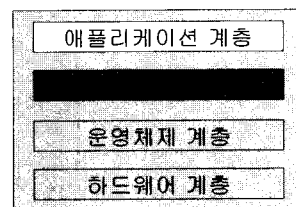


그림 1 J2EE 플랫폼 계층도

이러한 계층은 웹 애플리케이션이 J2EE 플랫폼 위에 존재하는 애플리케이션 계층에 포함되므로 기초를 이루는 하드웨어나 운영체제가 좀 더 빠르고 저렴하며 더욱 안정적인 특성들을 제공하기 위하여 애플리케이션 부분을 재작성하지 않아도 됨을 의미한다. 계층 아키텍처는 사용자의 관점에서 시스템을 바라보는 계층적 구조의 한 가지 특정한 유형이다. 이것은 전면에서 사용자 그리고 후면에는 기본적인 데이터 등으로 전면에서 후면(front-to-back)의 분리를 이끌어낸다. 전형적인 3 계층 아키텍처에서 사용자 인터페이스와 관련된 계층은 비즈니스 로직과 분리된 층에 있고, 비즈니스 로직 계층은 다시 데이터 접근 로직과 분리된 층에 있게 된다. 이 층들은 일반적으로 논리적일뿐만 아니라 물리적인 분리를 암시하기 때문에 서로 다른 기계(machine)에 존재할 가능성을 내포하고 있다. 따라서 웹 애플리케이션은 논리적으로뿐만 아니라 물리적으로도 나뉘어져 자신의 일을 수행하기 위해 다양한 층들 사이에 분산적 통신이 이루어지게 된다.

본 논문에서는 앞에서 언급한 바와 같이 웹 애플리케이션의 계층이 상이한 환경에서 실행될 수 있음으로 인해 각 계층의 전문가들이 따로 프로그램을 개발할 수 있다는 점에 착안하여, 프리젠테이션 계층에서 담당해야 할 여러 처리와 그 유형을 템플릿으로 정확히 제시하여 웹 애플리케이션의 프리젠테이션 계층을 보다 빠르게 개발 할 수 있도록 한다. 이는 결과적으로 웹 애플리케이션이 내포하고 있는 기존의 시스템과의 다른 점들을 개발자가 인식하지 않아도 본 프로젝트에서 제안하는 템플릿을 이용하여 보다 효율적으로 웹 애플리케이션을 개발할 수 있음을 의미한다.

J2EE 플랫폼 기반의 엔터프라이즈 개발환경이 급속히 성장하면서 J2EE 플랫폼 기반의 웹 애플리케이션 프레임워크에 관한 연구도 활발히 진행되고 있다. 현재 제공되고 있는 대부분의 프리젠테이션 계층 프레임워크는 Sun Microsystems(이하 Sun)사의 JSP Model 2 패러다임[8]의 다양한 취향을 나타낸다. Model 2는 Sun사가 웹 애플리케이션을 JSP 중심의 형태로 MVC 패턴을 나타낸 접근법을 말한다.

2.1 Sun Web Application Framework

Sun사의 Sun Web Application Framework는 유동성과 확장성 및 교차적 플랫폼(cross-platform)을 지원하는 엔터프라이즈 애플리케이션을 위한 J2EE 1.3 플랫폼의 기술들을 어떻게 사용하는지 설명하고 있다. 이 프레임워크의 프리젠테이션 계층에 대한 설계의 기본 골격은 Struts 프레임워크와 유사하다. 이 프레임워크에서는 프리젠테이션 계층 이외에 J2EE 플랫폼에서 엔터프라이즈 환경의 웹 애플리케이션을 위한 여러 가지 서버측 기술들을 어떻게 적용할 것인지에 대해 부가하고 있다. 그림 2는 Sun Web Application Framework를 나타낸 것으로 클래스 다이어그램으로 되어 있는 부분이 웹 애플리케이션 프레임워크의 핫스팟 부분이다. 그림 2

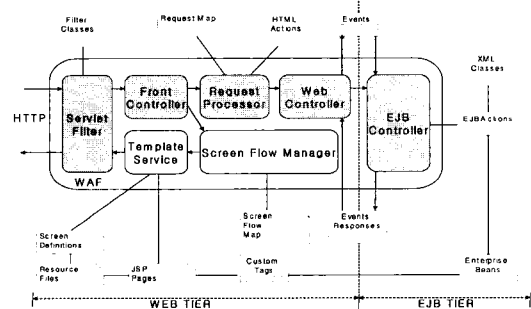


그림 2 Sun Web Application Framework

에서 보는 바와 같이 코드 생성기로 자동 생성할 수 있는 부분은 핫스팟 클래스를 상속받아 서브 클래스를 만드는 방법과 클래스 내에 있는 메소드를 오버라이드 하는 방법 등이다. 또한 기본적으로 이를 활용하기 위해 핫스팟 클래스 이외의 정보가 필요한데 이것은 웹 애플리케이션 프레임워크가 클래스들에게 정보를 전달하기 위해 사용되는 방법과 웹 애플리케이션 프레임워크의 전체적인 문맥이다.

2.2 Barracuda

Barracuda[9]는 복잡한 웹 애플리케이션을 구축하고 유지보수를 손쉽게 하기 위해 재사용 가능한 서버측 컴포넌트로 설계된 프리젠테이션 프레임워크이다. Barracuda에서 생성하는 프리젠테이션 계층은 서블릿을 이용한 프로그래밍 방식의 템플릿 엔진을 사용한다. 그림 3과 그림 4는 각각 Barracuda의 컴포넌트 모델과 이벤트 모델을 보여주고 있다. Barracuda의 컴포넌트 모델에는 템플릿이 하나의 컴포넌트로 간주되고 있다. 이 템플릿은 렌더러(Renderer)를 이용하여 HTML의 Text, Table, List, Action, Input과 같은 기본이 되는 구조들을 조합한 후 화면의 모습을 만들어 낸다. 만들어진 구조는 후에 이벤트가 부가 될 때 이벤트 흐름 처리를 이벤트 모델에 적용하여 만들어 낸다.

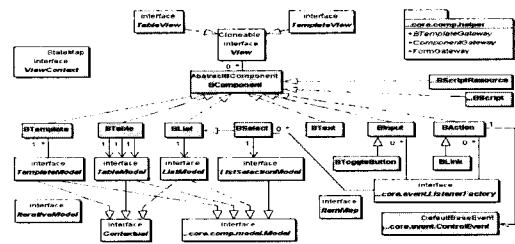


그림 3 Barracuda의 컴포넌트 모델

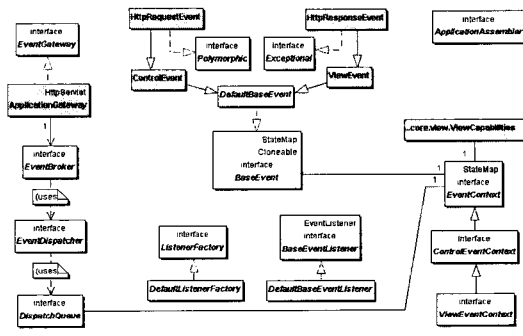


그림 4 Barracuda의 이벤트 모델

2.3 Struts

Struts[10]는 MVC 설계 패러다임을 근간으로 하는 Model 2 아키텍처를 갖는 프레임워크이다. 이것은 웹 애플리케이션에서 기반이 되는 JSP를 위한 흐름을 제어 하는데 초점을 맞추고 있다. 이 프레임워크가 제공하는 것은 컨트롤러 서블릿, JSP 커스텀 태그 라이브러리들 및 XML과 국제화를 지원하는 여러 유틸리티들이다. 그러나 Struts는 아직 컴포넌트 조립과 재사용에 관해서는 언급하지 않고 있다.

2.4 Turbine

Turbine[11]은 Struts와 유사한 프레임워크로서 서블릿(servlet) 기반의 프레임워크로서 Model 2+1이라는 별칭을 갖고 있다. 이것은 Model 2 형태에서 포함하고 있지 않은 보안 관련 문제를 포함하고 있는 모델이라는 의미를 내포하는 것이다. 이 프레임워크도 손쉬운 서버 측 개발을 위해 재사용 가능한 컴포넌트의 모음을 만들어 내는데 목적을 두고 있다. 이를 위해 Turbine은 툴킷 안에 200개 이상의 클래스 모음을 제공하고 있다. Turbine이 제공하는 것들은 MVC 흐름제어, O-R Mapping 도구의 통합, 작업 스케줄러, 지역화 서비스 및 캐싱서비스들이 있으며 이외에도 많은 서비스들을 제공한다. 또한 Turbine은 WebMacro[12], Freemarker 및 Velocity[13]와 같은 템플릿 엔진과의 통합성도 제공하고 있다.

이 외에도 Roja 프레임워크[14]는 시스템의 복잡성과 비즈니스의 요구사항 수준에 따라 경량급, 중(中)량급 및 중(重)량급 전자상거래 애플리케이션을 개발할 수 있는 각각의 솔루션(solution)을 제공하였다. 논문 [15]에서는 웹 애플리케이션을 빠르게 개발하기 위한 방법으로 구조적이며 유연한 스크립팅 프레임워크를 제안하였다.

3. 템플릿 설계

본 논문의 템플릿 및 코드 생성기는 웹 애플리케이션의 프리젠테이션 계층에 대한 모델링과 설계 및 구현 기능 등을 수행한다. 본 논문에서 제안한 XML 형태의 템플릿은 모델링 파일로서 코드 생성기의 입력이 된다. 코드 생성기는 주어진 웹 애플리케이션 프레임워크를 이용하여 웹 애플리케이션의 프리젠테이션 계층을 생성한다. 그림 5는 프리젠테이션 계층의 생성과정을 도식화한 것이다. 즉, 템플릿은 사용자가 작성하고자 하는 웹 애플리케이션의 프리젠테이션 계층의 정보를 나타내며, 이 정보를 입력받은 코드 생성기는 주어진 웹 애플리케이션 프레임워크를 바탕으로 프리젠테이션 계층을 자동 생성하는 기능을 수행한다. 즉, 설계된 템플릿은 코드

생성기의 입력으로 사용되어 사용자 정보를 파싱(parsing)한 후 주어진 웹 애플리케이션 프레임워크에 근거하여 웹 애플리케이션의 프리젠테이션 계층의 골격 코드를 생성한다.

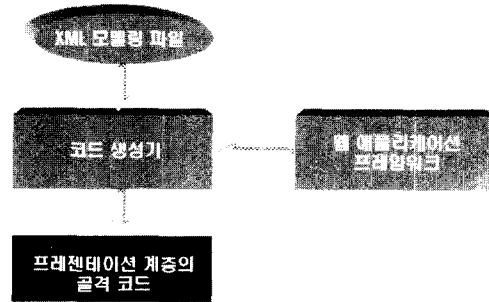


그림 5 템플릿 및 코드 생성기의 구조

제안된 템플릿 형태 및 요소는 프레임워크를 재사용하여 개발하고자 하는 개발자 입장을 고려하여 개발자 입장에서 필요한 내용을 추출하고, 여기에 웹 애플리케이션 특징을 부가하여 XML 형태의 모델 파일을 설계하였다. 템플릿의 제안 및 적용을 위한 과정을 도표화하면 그림 6과 같다.

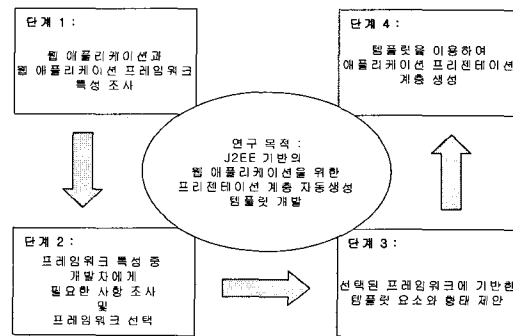


그림 6 템플릿의 제안 및 적용 단계

3.1 템플릿 설계를 위한 개요 작성

개발자들에게 필요한 프레임워크의 개요를 나타내기 위해서는 앞에서 언급한 바와 같이 프레임워크의 목적과 기능 및 개발적 구조도 등이 필요하다. 이 개요는 일반적으로 자연어로 표현되며 때때로 UML 다이어그램 등을 통해 표현되기도 한다.

개요에서 서술되어 있는 사항은 개발자가 빠르게 프

레이아웃을 익힐 수 있도록 하는데 그 목적이 있다. 기술해야 될 내용을 살펴보면 다음과 같다.

• 프레임워크의 목적

프레임워크가 어떠한 목적으로 설계되어 있는지 명확히 기술한다. 즉, 프레임워크가 해결하고자 하는 도메인이 명확히 기술되어야 하고 어떠한 기능을 지니고 있는지도 서술되어야 한다.

• 초기 파라미터

웹 애플리케이션이 수행될 때 애플리케이션의 환경변수 역할을 한다. 예를 들면, 웹 애플리케이션이 지원하는 나라의 언어 종류나 기본 문자 셋 등을 지정할 수 있다.

• 화면 선택에 대한 입력 형태

화면을 선택할 때 사용되는 형태를 말한다. J2EE 플랫폼에서는 화면 선택을 위해 직접적인 화면 파일을 지칭하기보다 별칭(alias)을 통해 제어를 선택하고 그 제어가 화면을 선택하는 방식으로 이루어져 있다.

• 이벤트 생성 형태

이벤트란 서버 측에서 수행되는 비즈니스 로직이 필요한 경우에 생성된다. 이러한 이벤트는 전통적인 애플리케이션에서 지칭하는 이벤트와 비슷하다. 이벤트가 생성되면 이벤트의 명칭과 이벤트에 적용될 파라미터가 필요하게 된다. 이러한 J2EE 플랫폼 기반의 웹 애플리케이션에서 이벤트의 명칭과 파라미터를 넘겨주는 방식은 일반적 웹 애플리케이션 작동방식과 같다. 그러나 웹 애플리케이션과 클라이언트-서버 애플리케이션의 차이점에서 알 수 있듯이 이벤트를 처리하는 방식은 기존의 방식과는 다르다. 즉, 기존의 전통적인 클라이언트-서버 애플리케이션에서는 컴포넌트가 이벤트 처리 부분을 포함하고 있지만 웹 애플리케이션에서 컴포넌트라 함은 단지 HTML 등의 메타언어일 뿐이고 이벤트를 처리하는 로직을 가지고 있지 않기 때문이다. 따라서 이러한 이벤트를 처리하는 방식에 대한 기술이 필요하다.

프레임워크 개요 작성시 언급되어야 할 사항은 다음과 같다. 프레임워크 개요 작성시 서술되는 사항은 필수 사항과 권고 사항으로 나뉜다.

- 프레임워크 개요 작성시 서술되어야 할 필수 사항은 다음과 같다.

- 1) 프레임워크의 적용 도메인
- 2) 프레임워크의 적용 범위
- 3) 프레임워크의 핫 스팟 부분
- 4) 프레임워크의 이벤트 처리 메커니즘

- 프레임워크 개요 작성시 서술될 수 있는 권고 사항은 다음과 같다.

- 1) 프레임워크의 개괄적 구조도
- 2) 프레임워크의 중요 부분 UML 클래스 다이어그램

3.2 프레임워크 기반 템플릿 설계

애플리케이션 개발자들은 프레임워크를 이용하여 원하는 애플리케이션을 작성하기 위한 방법을 알기를 원한다. 이것은 프레임워크의 작동 방식의 원리가 아니라 작동 방법을 알기 위한 목적 지향적 행동이다. 애플리케이션 개발자들은 적절한 핫스팟(hot spot)과 그것들을 어떻게 고객화(customization) 할 것인가에 대해 알기를 원한다. 즉, 상속받을 클래스들과 오버라이드하기 위한 방법 및 클래스들간의 조합 방법 등에 대한 것을 알고자 한다. 이러한 내용들은 템플릿 설계의 기반이 된다.

본 논문에서는 템플릿의 구성 요소와 구조를 정의하는데 있어서 XML 형식을 이용하였다. XML 형식 문서는 새로운 웹 애플리케이션의 코드를 자동으로 생성하기 위한 도구를 개발 및 응용하는데 있어서 쉽게 분석되고 적용될 수 있는 특징을 갖는다. 이러한 문서화를 위해 파악해야 하는 단계를 다음과 같이 정의할 수 있다.

- (단계 1) 웹 애플리케이션 프레임워크 개요 파악
- (단계 2) 역할에 따라 넘겨줄 파라미터 형식 파악
- (단계 3) 웹 애플리케이션 초기 파라미터 파악
- (단계 4) 핫스팟 부분의 클래스 요소 파악
- (단계 5) 이벤트 처리를 위한 중요 클래스 파악

다음의 표 1은 사용된 태그의 종류와 그 의미를 보여주고 있다.

표 1 태그의 종류와 의미

태그	의미
document	문서의 루트노드
intent	문서의 목적 서술
roles	담당 역할을 나열하는 태그의 부노드
role	담당 역할 서술
target	담당 클래스 서술
input	입력 파라미터 형식 서술
initParameters	초기 파라미터들을 나열하는 태그의 부노드
parameter	파라미터 서술
hotspotsuperclass	핫스팟을 나열하기 위한 태그의 부노드
tiers	핫스팟이 위치 하는 계층을 나열
tier	핫스팟의 위치하는 계층 서술, EJB 계층과 Web 계층
class	핫스팟의 클래스 서술
events	핫스팟 중 이벤트 담당하는 클래스들 나열
class	이벤트 담당하는 주요 클래스 서술

XML로 문서화하기 위해 앞에서 제안한 5 단계를 거치게 되는데 각 단계에서 서술하게 되는 태그의 내용은 다음과 같다.

(단계 1) <intent> 태그로 표현될 내용을 작성하게 된다. 여기에서는 Main WAF documentation이라고 표현하였다. 이 부분은 템플릿에 의하여 생성되는 애플리케이션의 특징을 나타내며, 코드 생성기에서 골격코드를 생성할 때 실질적으로 사용되지는 않는다.

(단계 2) <roles> 태그로 표현될 내용을 작성하게 된다. <roles> 태그는 프리젠테이션 계층에서 이루어지는 세 가지 동작 역할을 정의하게 된다. 첫째 화면 뷰를 설정하고, 둘째 비즈니스 로직에 의해 MVC에서의 모델이 변화되었을 때 보여줄 뷰를 설정하며, 셋째 비즈니스 로직이 수행되게 하는 실제 이벤트에 대하여 정의하게 된다. 따라서 <role> 태그는 세 번 반복된다.

(단계 3) <initParameters> 태그로 표현될 내용을 작성하게 된다. 여기서는 웹 애플리케이션 프레임워크의 초기 파라미터 변수를 정의한다.

(단계 4) <hotspotsuperclasses> 태그로 표현될 내용을 작성하게 된다. 여기서는 여러 핫스팟 요소 중 실제 클래스들만을 파악하여 작성한다. <initParameters> 태그와 <hotspotsuperclass> 태그는 프레임워크를 확장시킬 때 실질적으로 사용되는 태그이다. 이 태그를 통해 핫스팟 클래스를 상속받은 서브클래스와 이벤트 클래스 등을 생성하게 되고 전체 웹 애플리케이션에 적용될 초기 파라미터를 프레임워크에 넘겨주게 된다.

(단계 5) <event> 태그로 표현될 내용을 작성하게 된다. 이 태그가 표현하는 클래스는 핫스팟 부분이기도 하지만 웹 애플리케이션의 특성상 중요한 메커니즘이 담겨 있는 부분이기에 때문에 따로 처리하기 위해 특별히 강조하여 작성한다.

개발자는 담당 역할에 따른 클래스에 넘겨질 파라미터와 웹 애플리케이션에서 사용될 초기 파라미터, 핫스팟 부분 중 서브 클래스를 원하는 클래스에 대한 정보 및 액션 중 이벤트로 활용될 내용만을 파악하면 된다. 이렇게 파악된 내용을 템플릿으로 변환한다. 템플릿은 XML 문서 형태로 나타낸다. 그림 7은 위에 서술한 5 단계의 과정을 거친 문서를 나타내며, 그림 8은 템플릿화하기 위해 사용된 XML의 DTD(Document Type Definition)를 정의한 것이다. 즉, 본 논문에서 코드 생성기의 입력이 되는 템플릿은 프리젠테이션 계층에서 필요로 하는 요소를 정의하고 있는데, 첫째 프리젠테이션 계층이 담당하는 세 가지 요소를 나타내고, 둘째 프리젠테이션 계층 중 J2EE 플랫폼 아키텍처에서 웹 계층과 EJB 계층

의 역할에 따른 핫스팟 부분의 구성요소를 알려주며, 셋째 웹의 액션 방법이 기존의 클라이언트-서버 환경에서 사용된 이벤트 처리 방식으로 변환되는 웹 애플리케이션의 특성을 나타내고 있다.

```

<?xml version = "1.0"?>
<document level = "WAF">
<intent>Main WAF documentation</intent>
<roles>
<role name ="screen">
  <target name = "TemplateServlet" />
  <input form = "*.screen" />
</role>
<role name = "do">
  <target name = "MainServlet" />
  <input form = "*.do" />
</role>
<role name = "event">
  <target name = "Event" />
  <input form = "actionType=actionParameter" />
</role>
</roles>
<initParameters>
<parameter>default_language</parameter>
<parameter>languages</parameter>
</initParameters>
<hotspotsuperclasses>
<tiers>
<tier name = "web">
<class>WebClientController</class>
<class>HTMLAction</class>
</tier>
<tier name = "ejb">
<class>EJBClientControllerEJB</class>
<class>EJBAction</class>
</tier>
</tiers>
<event>
<class>Event</class>
</event>
</hotspotsuperclasses>
</document>

```

그림 7 웹 애플리케이션 프레임워크의 XML 문서

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT class (#PCDATA)>
<!ELEMENT document (intent, roles, initParameters, hotspotsuperclasses)>
<ATTLIST document
  level CDATA #REQUIRED
>
<!ELEMENT event (class)>

```

```

<!ELEMENT hotspotsuperclasses (tiers, event)>
<!ELEMENT initParameters (parameter+)>
<!ELEMENT input EMPTY>
<!ATTLIST input
  form CDATA #REQUIRED
>
<!ELEMENT intent (#PCDATA)>
<!ELEMENT parameter (#PCDATA)>
<!ELEMENT role (target, input)>
<!ATTLIST role
  name (do | event | screen) #REQUIRED
>
<!ELEMENT roles (role+)>
<!ELEMENT target EMPTY>
<!ATTLIST target
  name (Event | MainServlet | TemplateServlet)
  #REQUIRED
>
<!ELEMENT tier (class+)>
<!ATTLIST tier
  name (ejb | web) #REQUIRED
>
<!ELEMENT tiers (tier+)>
    
```

그림 8 템플릿의 XML DTD

반면, 그림 9는 본 논문에서 코드 생성기에 적용하기 위해 선정한 Sun Web Application Framework를 나타내는 것으로 클래스 다이어그램으로 되어 있는 부분이 웹 애플리케이션 프레임워크의 핫스팟 부분이다. 그림에서 보는 바와 같이 코드 생성기로 자동 생성할 수 있는 부분은 핫스팟 클래스를 상속받아 서브 클래스를 만드는 것과 클래스 내에 있는 메소드를 오버라이드하는 방법 등이다. 또한 기본적으로 이를 활용하기 위해 핫스팟 클래스 이외의 정보가 필요한데, 이것은 웹 애플리케이션 프레임워크가 클래스들에게 정보를 전달하기 위해 사용되는 방법과 웹 애플리케이션 프레임워크의

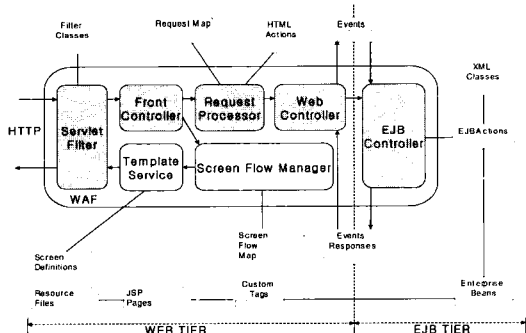


그림 9 적용된 웹 애플리케이션 프레임워크

전체적인 문맥이다.

4. 코드 생성기의 구현

4.1 코드 생성기

코드 생성기는 총 5개의 클래스로 구성되어 있으며 각각의 기능은 표 2와 같다. 또한 그림 10은 이러한 클래스간의 관계 및 상속성 등을 표현한 클래스 다이어그램이다.

표 2 코드 생성기의 클래스

클래스 이름	기능
WAGen	템플릿을 이용하여 프리젠테이션 계층을 생성하는 메인 클래스
TempletImport	템플릿을 DOM 객체로 읽어들이며, 템플릿의 형태가 유효한 것인지 판단하는 역할을 담당하는 클래스
TempletView	사용자에게 템플릿을 보여주는 클래스
CodeGen	읽어들인 템플릿과 프레임워크 저장소에서 읽어들이는 웹 애플리케이션 프레임워크를 이용하여 프리젠테이션 계층의 소스를 생성하는 클래스
FrameworkImport	코드 생성을 위해 필요한 정보 중의 하나인 프레임워크를 선택하여 읽어내며, 추후 다른 프레임워크도 이용할 수 있도록 확장성을 위해 설계된 클래스

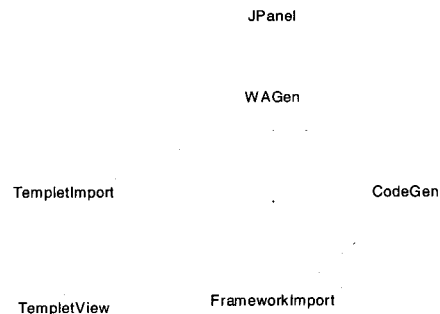


그림 10 코드 생성기의 클래스 다이어그램

그림 11은 코드 생성기를 실행시킨 위저드(wizard)를 보여준다. 화면의 왼쪽 패널(panel)은 템플릿의 구조를 트리 형태로 보여주고, 오른쪽 패널은 실제 프리젠테이션 계층의 생성과정을 나타내는 에디터 패널이다. 본 논문의 코드 생성기는 XML 형태의 템플릿을 읽어들이어 파싱하는 부분과 웹 애플리케이션을 선택하는 부분 및 프리젠테이션 계층의 소스를 생성하는 부분으로 이루어져 있다.

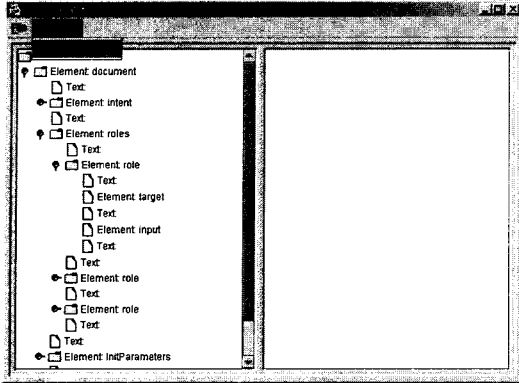


그림 11 코드 생성기의 위저드 실행 화면

그림 12는 코드 생성기의 WAGen 메인 클래스 코드의 주요 부분을 나타내고 있다. 이 코드에서 WAGen 클래스는 생성기의 모습을 만들어내고 템플릿 파일을 이용하기 위한 이전단계의 작업과 메뉴 생성을 담당하고 있다. 각 메뉴에 의하여 템플릿을 읽어들이고 웹 애플리케이션의 프리젠테이션 계층을 생성한다. 각 메뉴의 이벤트 리스너는 이러한 행동을 위한 TempletImport 클래스와 CodeGen 클래스에 이벤트를 전달한다. 코드 생성기는 [File] 메뉴와 [Generate] 메뉴로 구성되어 있다. 여기서 [File] 메뉴는 템플릿을 읽어들이거나 또는 다른 템플릿 파일을 읽어들이는 역할을 담당한다. 그리고 [Generate] 메뉴에서는 특정 웹 애플리케이션 프레임워크를 선택하여 이를 기반으로 템플릿 파일의 정보를 이용하여 웹 애플리케이션의 프리젠테이션 계층의 골격 코드를 생성한다.

```
public class WAGen extends JPanel
{
    // 전역 변수 선언
    static Document document;

    static final int windowHeight = 460;
    static final int leftWidth = 300;
    static final int rightWidth = 340;
    static final int windowWidth = leftWidth + rightWidth;

    public WAGen()
    {
        // Boarder 생성
        .....
        // 트리 설정
        JTree tree = new JTree(new DomToTreeModel Adapter());
```

```
// 왼쪽 패널 설정
JScrollPane treeView = new JScrollPane(tree);
treeView.setPreferredSize(
    new Dimension( leftWidth, windowHeight ));

// 오른쪽 패널 설정
JEditorPane htmlPane = new JEditorPane("text/
html", "");
.....

// 양 패널을 구분짓는 스플릿 패널 설정
JSplitPane splitPane =
    new JSplitPane( JSplitPane.HORIZONTAL_SPLIT,
        treeView,
        htmlView );
.....

// Add GUI components
this.setLayout(new BorderLayout());
this.add("Center", splitPane );
} // constructor

public static void main(String argv[])
{
    if (argv.length != 1) {
        System.err.println("사용법이 틀렸습니다.");
        System.exit(1);
    }
    makeFrame();

    public static void makeFrame() {
        // Set up a GUI framework
        JFrame frame = new JFrame("WA Generator");
        frame.addWindowListener(
            new WindowAdapter() {
                public void windowClosing(WindowEvent e)
                {System.exit(0);}
            }
        );

        // 파일메뉴바 설정
        boolean inStandalone = false;

        JMenuBar jMenuBar1 = new JMenuBar();
        .....
        // 여기까지 메뉴바 생성 및 선언

        // 파일메뉴 초기화
        menuFile.setMnemonic('F');
        .....
        // Generate 메뉴초기화 끝
        // 메뉴바 등록
        jMenuBar1.add(menuFile);
        .....
        // 이벤트 처리 시작
        jmiExit.addActionListener(new java.awt.event.
            ActionListener() {
```



```

public void actionPerformed(ActionEvent e) {
    jmiExit_actionPerformed(e);
}
});
.....
/* 이벤트 처리 완료, 이벤트란 생성기에서 소스를
   생성시키는 이벤트 등을 포함
   이벤트는 이벤트 처리하는 새로운 클래스를
   인스턴스화 함으로써 처리 */

// 트리과 뷰를 설정하고 이것들을 화면에 표시
final WAGen wagen = new WAGen();
.....
} // makeFrame 완료
}
    
```

그림 12 WAGen 메인 클래스 코드의 주요 부분

4.2 구현 사례

4.2.1 템플릿을 활용한 웹 애플리케이션 개발 과정

본 논문에서 제안하고 있는 템플릿은 프레임워크 자동 코드 생성기의 입력 파일로 모델링 되었다. 실제 XML 문서를 활용하여 웹 애플리케이션의 프리젠테이션 계층에 대한 골격 코드를 생성하는 과정과 생성되는 골격 코드를 보여주기 위해 간단한 웹 애플리케이션 예를 들고자 한다.

이 웹 애플리케이션은 영어로 되어 있는 웹 애플리케이션의 화면을 한국어로 되어 있는 화면으로 변환하는 애플리케이션이다. 애플리케이션은 영어와 한국어 두 가지로 되어 있는 JSP 파일과 HTML 파일로 이루어져 있고 Locale을 한국어로 변환하면 각각의 영문 JSP 파일과 HTML 파일에 대응하는 한국어 JSP 파일과 HTML 파일을 읽어들이어 화면에 표시하며 그 반대의 경우도 성립한다. 따라서 이 웹 애플리케이션은 ChangeLocale이라는 단 하나의 이벤트를 수행하게 된다. 그림 13은 이러한 웹 애플리케이션의 템플릿 XML 문서이며, 그림 14는 프리젠테이션 계층을 생성하기 위해 사용된 프레임워크의 클래스 다이어그램을 나타낸다.

```

<?xml version="1.0"?>
<document level="Application">
<intent>Main WAF documentation</intent>
<roles>
<role name="screen">
<target name="TempletServlet" />
<input form="*.screen" />
<elements>
<element></element>
</elements>
    
```

```

</role>
<role name="do">
<target name="MainServlet" />
<input form="*.do" />
<elements>
<element>ChangeLocale</element>
</elements>
</role>
<role name="event">
<target name="Event" />
<input form="changelocale=*" />
<elements>
<element>euc_kr</element>
<element>eu_en</element>
</elements>
</role>
</roles>
<initParameters>
<parameter>default_language</parameter>
<parameter>languages</parameter>
</initParameters>
<hotspotsuperclasses>
<tiers>
<tier name="web">
<class>WebClientController</class>
<class>HTMLAction</class>
</tier>
<tier name="ejb">
<class>EJBClientControllerEJB</class>
<class>EJBAction</class>
</tier>
</tiers>
<event>
<class>Event</class>
</event>
</hotspotsuperclasses>
<extendedclasses>
<tiers>
<tier name="web">
<class>ChangeLocaleHTMLAction</class>
</tier>
<tier name="ejb">
<class>ChangeLocaleEJBAction</class>
</tier>
</tiers>
<event>
<class>ChangeLocaleEvent</class>
</event>
</extendedclasses>
</document>
    
```

그림 13 웹 애플리케이션 템플릿

이 애플리케이션은 ChangeLocale이라고 하는 하나의 이벤트를 가지고 있으므로 템플릿을 적용하여 생성기

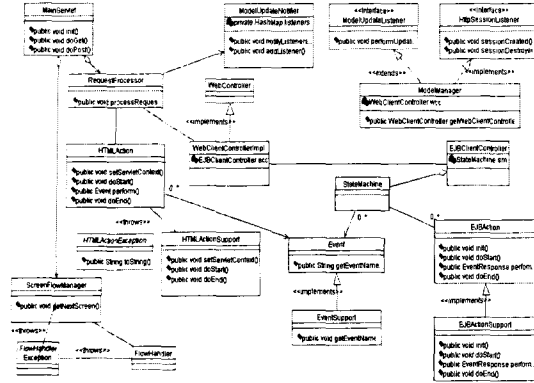


그림 14 웹 애플리케이션 프레임워크의 클래스 다이어그램

를 통해 애플리케이션의 골격 코드를 생성하면 프레임워크의 핫스팟 부분의 클래스들을 상속받아 ChangeLocale에 관련된 골격 코드를 자동으로 생성해준다. 다음의 그림 15부터 그림 18까지는 상속받은 핫스팟 클래스들에 대한 상세 코드를 나타내고 있다. 그림 15는 웹에서 이벤트로 변환되기 이전에 처리하게된 액션에 관련된 핫스팟의 추상 클래스이고, 그림 16과 17은 액션을 이벤트로 변환하여 이벤트 형식을 정의하는 클래스이다. 또한 그림 18은 이벤트에 대한 비즈니스 로직을 처리하기 위해 EJB 컴포넌트를 인스턴스화시키기 위한 EJB 계층의 액션을 정의한 추상 클래스이다.

```
<?xml version="1.0"?>
<document level="Application">
<intent>Main WAF documentation</intent>
<roles>
<role name="screen">
  <target name="TempletServlet" />
  <input form="*.screen" />
  <elements>
  <element></element>
  </elements>
  </role>
</roles>
</document>
```

그림 15 HTMLActionSupport

```
public class EventSupport implements Event {
  public String getEventName() {
    return null;
  }
}
```

그림 16 EventSupport

```
public abstract class EventResponseSupport
implements EventResponse {
  private Object payload = null;
  public EventResponseSupport(Object payload) {
    this.payload = payload;
  }
  public Object getPayload() {
    return payload;
  }
}
```

그림 17 EventResponseSupport

```
public abstract class EJBActionSupport
implements java.io.Serializable, EJBAction {
  protected StateMachine machine = null;
  public void init(StateMachine machine) {
    this.machine = machine;
  }
  public void doStart() {}
  public void doEnd() {}
}
```

그림 18 EJBActionSupport

그림 19부터 그림 21까지는 템플릿을 입력받아 코드 생성기를 통해서 생성되는 골격 코드 중 몇 가지 주요 클래스들의 모습이다. 그림 19는 웹에서의 액션을 처리하는 ChangeLocale에 대한 코드이고, 그림 20은 ChangeLocale에 대한 이벤트를 정의하는 코드이다. 또한 그림 21은 ChangeLocale이라는 이벤트를 수행하기 위한 비즈니스 로직을 처리하는 EJB 컴포넌트를 인스턴스화하기 위해 사용되는 코드이다. 이러한 코드는 코드 생성기가 생성한 골격 코드에 약간의 코드를 추가하여 웹 애플리케이션의 프리젠테이션 계층을 완성할 수 있음을 보여 주고 있다.

```
public final class ChangeLocaleHTMLAction
extends HTMLActionSupport {
  // 아래 부분에 필요한 코드 작성
  public Event perform(HttpServletRequest request)
  throws HTMLActionException {
    // 아래 부분에 필요한 코드 작성

    HttpSession session = request.getSession();
    if (/* 이곳에 인자 작성 */) {
      // 아래 부분에 필요한 코드 작성
      return new ChangeLocaleEvent();
    } else {
```

```

        throw new
        HTMLActionException("ChangeLocaleAction:
        Exception");
    }
}
    
```

그림 19 ChangeLocaleHTMLAction

```

public class ChangeLocaleEvent extends EventSupport {
    // 아래 부분에 필요한 코드 작성
    public ChangeLocaleEvent() {
    // 아래 부분에 필요한 코드 작성
    }
    // 아래 부분에 필요한 코드 작성

    public String getEventName() {
        return "java:comp/env/event/ChangeLocaleEvent";
    }
}
    
```

그림 20 ChangeLocaleEvent

```

public class ChangeLocaleEJBAction extends
EJBActionSupport {
    public EventResponse perform(Event e) throws
    EventException {
        ChangeLocaleEvent ce = (ChangeLocaleEvent)e;
        machine.setAttribute(/* 이곳에 인자 작성 */);
        return null;
    }
}
    
```

그림 21 ChangeLocaleEJBAction

5. 비교 및 평가

본 연구의 결과인 템플릿 및 코드 생성기를 Struts 프레

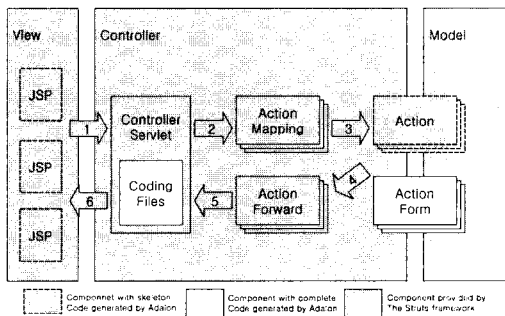


그림 22 Struts Framework Generator Overview

임워크[9]를 기반으로 골격 코드를 생성하는 Struts Framework Generator의 기능과 비교하였다. Struts Framework Generator의 기능에 대한 개요도는 그림 22와 같다.

Struts Framework Generator는 전자 상거래 프로세스를 설계하는 도구인 Synthis사의 Adalon이라는 도구의 한 기능이다. 이 프레임워크 생성기는 Adalon 도구에서 설계한 애플리케이션의 모델을 이해하기 쉽고 고객화하기 쉬운 형태의 구현물을 Struts Framework를 기반으로 골격 코드로 생성한다. 이렇게 생성하는 골격 코드의 종류는 다음과 같다.

- Action Servlet
- Action classes, Form classes, Business logic classes
- Struts Configuration and Web XML Files
- JSP for Client Test

이 생성기는 Adalon 도구에서 설계된 프로세스 모델을 통하여 코드를 생성하는데 이러한 모델은 본 논문에서 정의하고 설계한 템플릿과 같은 의미를 지닌다. 그러나 이 도구에서는 J2EE 플랫폼 기반에서의 개발 프로세스 모델을 따르는 것이 아니고 Synthis사에서 만들어 낸 프로세스 모델을 따르고 있다. 따라서 Struts 프레임워크 생성기는 Synthis사에서 만들어 낸 모델을 Struts 프레임워크에 매핑시키는 방법을 이용하고 있다. 또한 이 생성기에서 생성되는 Action Servlet 등의 코드는 실제 프레임워크를 상속한 골격 코드가 아니라 프레임워크와 매핑하기 위해 인터페이스 형태의 스템프(stub) 코드를 제공하는 형태로 되어 있다. 이 코드 생성기는 프레임워크를 직접 이용하는 방식으로 되어 있지 않다는 것이 본 논문의 코드 생성기와 다른 점이라 할 수 있다. 즉, 본 논문의 생성기는 Synthis사에서 채택한 방식이 아니라 프레임워크의 핫스팟을 직접 이용하는 방식으로 되어 있다. 이것은 개발자에게 프레임워크의 핫스팟을 알려주고 그 부분에 대한 여러 파라미터를 입력하여 프레임워크를 재사용을 용이하게 해주고 있다. 또한 매핑하는 방식이 아니기 때문에 프레임워크에 알맞은 골격 코드를 생성한다. 이러한 방식에는 각각 장·단점이 존재하게 된다. 전자의 방식은 여러 가지 다른 프레임워크를 알지 못하더라도 단 하나의 프로세스 모델만을 알면 여러 프레임워크에 매핑시켜 프레임워크를 재사용할 수 있다는 장점이 있다. 그리고 후자의 방식은 전자의 방식을 사용할 때 간과하기 쉬운 프레임워크 자체의 설계 의도를 모두 사용할 수 있다는 장점이 있다. 프레임워크를 설계할 때에는 여러 가지 의도를 가지고

496-497, 1997.

[7] J. Conallen, "Building Web Applications with UML," Addison-Wesley Publishing Company, 1999.

[8] Understanding JavaServer Pages Model 2 Architecture, <http://www.javaworld.com/>

[9] Barracuda: Open Source Presentation Framework, <http://barracuda.enhydra.org/>, 2001.

[10] Struts: Open Source Presentation Framework, <http://jakarta.apache.org/struts/index.html>

[11] Turbine: Open Source Presentation Framework, <http://jakarta.apache.org/turbine/>

[12] Web Macro: Open Source Presentation Framework, <http://www.webmacro.org/>

[13] Velocity: Open Source Presentation Framework, <http://jakarta.apache.org/velocity/>

[14] Source Code for Roja Framework and Sample Project, <http://rojaware.wo.to>

[15] V. Tam, W. K. Foo, and R. K. Gupta, "A Fast and Flexible Framework of Scripting for Web Application Development: A Preliminary Experience Report," 2000.



김 송 주

1998년 전북대학교 화학공업공학과 졸업 (공학사). 2002년 전북대학교 대학원 전산통계학과 졸업(이학석사). 관심분야는 소프트웨어공학, 에이전트, J2EE, 통합매시징 시스템 등

장 옥 배

정보과학회논문지 : 컴퓨팅의 실제
제 9 권 제 1 호 참조



유 철 중

1982년 전북대학교 전산통계학과 졸업 (이학사). 1985년 전남대학교 대학원 전산통계학과 졸업(이학석사). 1994년 전북대학교 대학원 전산통계학과 졸업(이학박사). 1982년~1985년 전북대학교 전자계산소 조교. 1985년~1996년 전주기전 여자대학 전자계산과 전임강사~부교수. 1997년~현재 전북대학교 자연과학대학 컴퓨터학과 전임강사~조교수. 관심분야는 소프트웨어공학, 에이전트공학, 컴포넌트기술, 분산객체기술, GNSS (GPS), GIS, 멀티미디어, 인지과학 등



채 정 화

1992년 군산대학교 컴퓨터학과 졸업 (이학사). 1999년 전북대학교 교육대학원 전자계산교육전공 졸업(교육학석사). 2002년 전북대학교 대학원 전산통계학과 박사과정 수료. 관심분야는 소프트웨어 컴포넌트 기술, 객체지향 기법, 웹 애플리케이션 기술 등