

자바 오픈 소스 프레임워크 기반의 소프트웨어 개발 문서 표준안

진혜영*, 류성열**

*숭실대학교정보과학대학원 소프트웨어공학과

**숭실대학교 IT 대학 컴퓨터학부 교수

e-mail : jeonhye0@ssu.ac.kr

Standard for Java Open Source Framework Oriented Software Development Documentation

Hye-Young Jeon*, Sung-Yul Rhew**

*Dept. of Computer Science, Soongsil University Graduate School of Information Science

**Professor, Dept. of Computer Science, Soongsil University

요 약

자바 프로그래밍언어를 이용한 개발이 많아지면서 사용할 수 있는 소프트웨어도 많이 배포되고 있다. 오픈 된 자바에 관련된 소프트웨어는 특정 부분에서 사용하는 기능을 개발한 것도 있지만 개발환경을 제공하는 프레임워크가 많은 부분을 차지한다. 개발자는 소프트웨어를 개발할 때 하나이상의 프레임워크를 선정하여 소프트웨어 아키텍처를 설계하게 되는데 각 프레임워크의 역할은 모든 프로그램이 동일하지는 않다. 그래서 프레임워크간의 관계 및 역할 등의 정보가 문서화가 이루어져서 개발자 뿐 아니라 사용자에게 정보를 제공해야 한다. 특히 오픈 소스는 빈번하게 업그레이드되어 변화하는 프레임워크 정보를 이해하지 못하면 매우 혼란스럽게 된다. 본 연구는 자바 오픈 프레임워크를 이용하여 소프트웨어를 개발하려고 할 때 프레임워크의 정보 관리 및 소프트웨어 아키텍처를 설계 후 작성해야 하는 문서의 표준안을 제시한다. 이는 개발자는 개발된 소프트웨어의 구조를 표준화된 문서로 배포하면 사용자는 그 문서를 활용하여 소프트웨어의 구조 뿐 아니라 오픈 소스 프레임워크의 역할과 환경정보를 빠르게 이해하고 향후 효과적인 유지보수를 하는데 도움을 줄 수 있다.

Open Source Framework; Development Documentaion; J2EE Architecture

1. 서론

자바 프로그래밍언어를 이용한 개발이 많아지면서 사용할 수 있는 소프트웨어도 많이 배포되고 있다. 배포된 소프트웨어 중에 특정 기능을 구현한 소프트웨어는 요구사항의 다양성 때문에 그대로 사용하지 못하는 경우가 대부분이다. 이에 반해 프레임워크는 프로그램 디자인 패턴을 이용하여 개발 환경을 제공하는 틀로서 여러 분야에서 사용되고 있다.

2007년 8월 25일 현재 소스포지에서도 알 수 있듯이 자바 관련 프로젝트를 살펴보면 14516 개 중 JBoss.org, jEdit, Hibernate, Spring Framework, Eclipse Checkstyle Plug-in 등이 활발하게 활동하고 있는데 대부분이 개발에 성능을 높이기 위한 프레임워크나 에디터임을 확인해 볼 수 있다. 특히 아파치 그룹에서 오픈한 자바 프로젝트도 iBATIS, Jackrabbit, Cocoon, Struts 등이 프레임워크에 관한 소프트웨어에 대한 정보를 확인해 볼 수 있다.

프레임워크는 각 고유한 기능을 가지고 있다. 그래서 개발 시 하나의 프레임워크를 사용하는 경우도 있지

만 대부분은 여러 프레임워크의 특징을 활용하여 개발을 하게 된다. 이렇다 보면 프레임워크 간의 특징적인 기능도 있지만 중복되는 기능도 있어서 설계자가 프레임워크간의 관계 및 정의를 명확하게 하지 않으면 개발 시 오히려 방해가 될 수 있다. 특히 빈번하게 업데이트 되는 프레임워크 때문에 정보를 관리하지 않으면 효과적으로 개발하기 위해서 사용된 프레임워크는 오히려 혼란을 줄 수 있다.

본 연구는 자바 오픈 프레임워크를 이용하여 소프트웨어를 개발하려고 할 때 프레임워크의 정보 관리 및 소프트웨어 아키텍처를 설계 후 작성해야 하는 문서의 표준안을 제시한다. 이는 개발자는 개발된 소프트웨어의 구조를 표준화된 문서로 배포하면 사용자는 그 문서를 활용하여 소프트웨어의 구조 뿐 아니라 오픈 소스 프레임워크의 역할과 환경정보를 빠르게 이해하고 향후 효과적인 유지보수를 하는데 도움을 줄 수 있다.

2. 관련연구

이번 연구는 오픈 소스 프레임워크의 종류와 이를 배

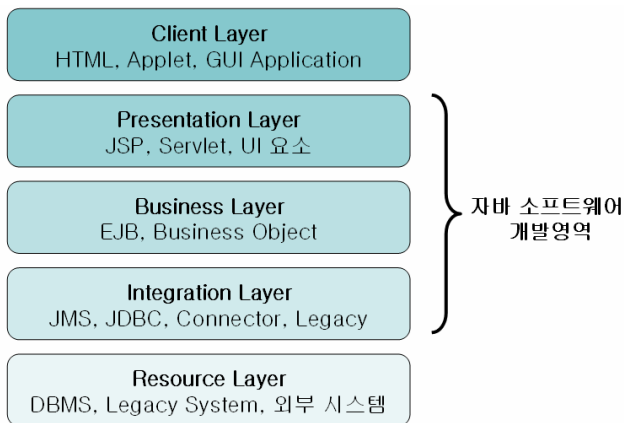
치할 수 있는 J2EE Architecture Design Patterns 을 이해한다. 오픈 소스 아키텍처를 설계할 수 있는 방법론을 알아보고 현재 오픈 소스 프레임워크를 활용하여 설계된 아키텍처를 확인한다.

2.1 Open Source Framework

아파치 그룹, 소스포지, KLDAP 등의 사이트를 보면 오픈 소스 프로젝트 정보를 살펴 볼 수 있다. 특히 자바에 관련된 프로젝트는 주로 개발을 좀더 효율적으로 하고 디자인 패턴을 구현해 놓은 프레임워크가 대부분이다. 프레임워크 관련 소프트웨어를 살펴보면 JBoss, Hibernate, Spring, Ant, Beehive, iBATIS, Jackrabbit, Struts, Tapestry 등 매우 다양하다. 이 중 Struts 는 MVC Model2 디자인 패턴기반으로 개발할 수 있는 환경을 제공하는 프레임워크로 가장 활성화되다.

2.2 J2EE Architecture Design Patterns

J2EE Platform 을 기반으로 소프트웨어를 개발할 때 일반적으로 발생될 수 있는 디자인 패턴 정보를 제공하고 있다. 이를 J2EE Architecture Design Patterns 이라고 한다.



(그림 1) J2EE Architecture Layer

J2EE Architecture Design Patterns 에서 Client Layer 와 Resource Layer 를 제외한 Presentation Layer, Business Layer, Integration Layer 를 자바 소프트웨어 개발영역으로 하고 소프트웨어 아키텍처를 설계한다.

자바 개발 영역에 맞는 Open Source Framework 이다.

- ◆ Presentation Layer
 - : MVC Model 2, Struts, Velocity, SiteMesh, JSF
- ◆ Business Layer
 - : Spring, HiveMind, EJB-SessionBean
- ◆ Integration Layer
 - : Hibernate, EJB-EntityBean, MessageDrivenBean, iBatis, SpringDAO, JDO, Castor

2.3 Architecture Framework Method

아키텍처를 설계하기 위한 개발 방법론은 Zachman's EA(Enterprise Architecture) Framework, TOGAF(The Open

Group Architecture Framework) EA, RUP's 4+1 Views, UMLComponents, TTAS.KO-11.0030 등이 있다. 이러한 방법론은 컴포넌트 기반으로 아키텍처 설계의 중요성 및 방법에 대해 정의하고 있다. 이 중 TOGAF 는 미국 방성의 TAFIM(Technical Architecture for Information Management)을 비 영리 단체인 오픈 그룹에서 기술 이전을 받아 이를 산업계의 특성에 맞도록 재구성한 것이다. 이러한 TOGAF 를 포함하여 ADM(Architecture Development Method)는 프레임워크 아키텍처 방법론을 수립하였다.

ADM 은 다음 7 단계의 절차로 구성된다.

- ◆ Phase 1: Initiation and Framework
- ◆ Phase 2: Baseline Description
- ◆ Phase 3: Target Architecture
- ◆ Phase 4: Opportunities and Solutions
- ◆ Phase 5: Migration Planning
- ◆ Phase 6: Implementation
- ◆ Phase 7: Architecture Maintenance

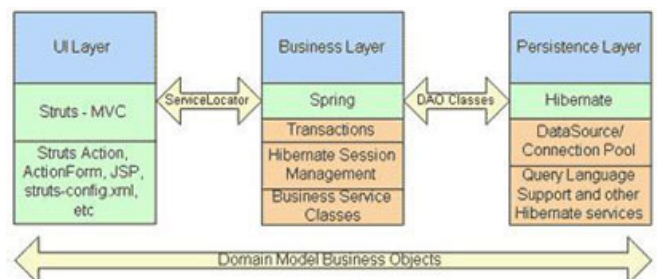
Phase 6 에서는 시스템을 구현하고 배포하는데 개별적인 프로젝트의 범위, 기술적인 아키텍처의 관점에서의 요구사항, 아키텍처 관련 계약 사항 등에 대한 문서화를 한다. Phase 7 에서는 기술적인 환경 및 비즈니스 환경 변화에 대한 모니터링을 제공한다.

ADM 은 IT 아키텍처를 설계하는 방법론으로서 검증되었고 여러 분야에서 사용되고는 있지만 전체적인 설계를 위한 프로세스의 제안으로 세부적인 내용은 별도의 프로세스 정의가 필요하다.

2.4 Open Source Framework Architecture

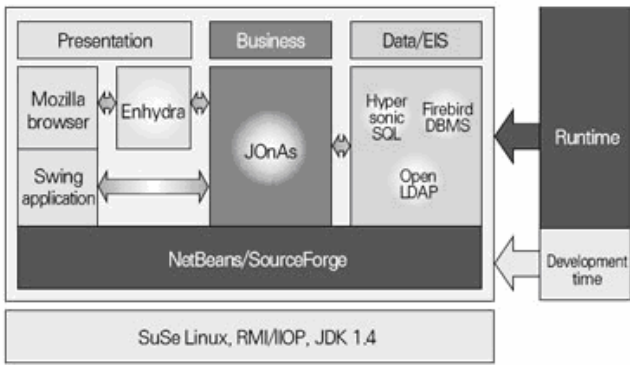
일반적으로 자바의 Open Source Framework 를 이용하여 아키텍처를 설계하려고 할 때 함 어려움이 많았다. 어떤 영역에서 어떤 프레임워크를 선정하는데 시 모든 프레임워크를 분석 후에 사용해야 한다는 막막한 점이 있다.

다음과 오픈 프레임워크를 이용하여 설계된 아키텍처로 여러 분야에서 사용되거나 연구되고 있다.



(그림 2) OnJava's Architecture

(그림 2)는 오라일리가 운영하는 OnJava.com 에 소개된 별도의 프레임워크로 이미 널리 알려져 있는 Struts, Spring, Hibernate 를 기반으로 아키텍처를 설계하였고 가장 보편화된 구조이다.

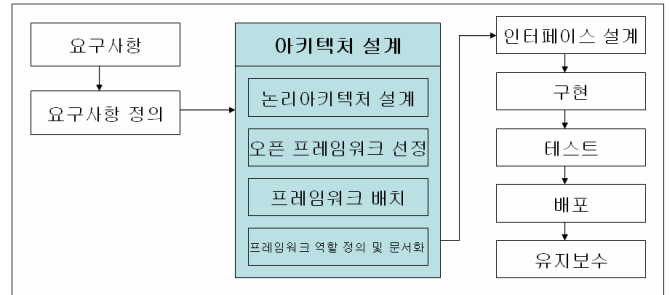


(그림 3) EJOSA

EJOSA(Enterprise Java Open Source Architecture)는 오픈 소스 및 프레임워크를 이용하여 자바에서 엔터프라이즈 환경을 구축할 수 있는 아키텍처 구조이다. 이전에는 개발 프레임워크를 중심으로 설계되었다면 EJOSA는 WAS, 개발도구 등을 모두 제공하였다.

기 위한 프레임워크를 직접 개발하기 보다는 오픈 된 프레임워크를 선별하여 배치하여 아키텍처를 설계한다.

오픈 프레임워크를 이용하여 아키텍처를 설계하는 일련의 과정을 UML Component, ADM 을 기반으로 다음과 같이 도식화 하였다.



(그림 5)J2EE Architecture 기반 오픈 소스 프레임워크 설계 프로세스

| | | |
|-------|---|--------------|
| 활용 샘플 | OpenAuction | Garasani KMS |
| 프레임워크 | VSSH(Velocity Structs Spring Hibernate) | |
| 개발 환경 | OpenAuction | Garasani KMS |
| 동작 환경 | Apache | Tomcat |
| | JBoss | HSQLDB |
| 방법론 | OpenCBD | |

(그림 4) VSSH

이전의 오픈 프레임워크를 이용한 아키텍처 설계와 달리 VSSH(Velocity Structs Spring Hibernate)등 프레임워크를 중심으로 방법론, 동작환경, 개발환경, 활용 샘플 등 다양한 정보를 제공하고 있다. 특히 VSSH 포럼으로 여러 개발자가 아키텍처 설계에 참여 하고 있다.

3. 자바 오픈 소스 프레임워크 기반의 소프트웨어 개발 문서 표준안

관련연구에서도 살펴 볼 수 있듯이 오픈 프레임워크를 이용하여 소프트웨어 아키텍처를 설계할 수 있는 방법은 매우 다양하다. Struts 와 같이 항상 언급 되는 기술도 있지만 초기엔 Hibernate 를 기반으로 설계되었지만 최근엔 iBatis 를 연계한 프로젝트가 많이 발생하고 있다.

오픈 프레임워크를 이용하여 아키텍처를 살펴 보면 공통점은 개발 영역이 J2EE Architecture Layer 를 기반으로 작성되었다는 것이다. 그래서 업무에 맞는 프레임워크를 선택하고 J2EE Architecture 를 기반으로 각 Layer 에 정의된 Design Patterns 에 따라 설계하면 보다 쉽게 작성할 수 있다.

소프트웨어를 본격적으로 개발하기 전에 아키텍처를 설계해야 한다. 이번 연구에서는 아키텍처를 설계하

3.1 논리아키텍처 설계

정의된 요구사항을 기반으로 논리 아키텍처를 설계한다. 논리 아키텍처란 개발 환경이나 시스템이 정해지지 않았을 때 업무의 흐름을 도식화 함으로써 세부 업무를 정의하기 전 개발환경 선정, 오픈 프레임워크 선정에 기준이 된다. 이때 작성되는 문서는 컴포넌트 명세서라고 한다.

컴포넌트 명세서 세부 요소 다음과 같다.

- ◆ 컴포넌트명
- ◆ 컴포넌트 개요
- ◆ 컴포넌트 배경도(외부 상호작용)
- ◆ 관련 요구사항
- ◆ 전제조건
- ◆ 우선순위
- ◆ 사용자(Actor)
- ◆ 관련 요구사항

3.2 오픈 프레임워크 선정

오픈 프레임워크에는 고유한 기능과 특징이 있다. 이 과정에서는 업무에 맞는 오픈 프레임워크를 선정하여 이후 아키텍처를 설계할 때 제공할 정보를 관리한다. 오픈 프레임워크 선정 단계에서는 오픈 프레임워크 기능 명세서를 작성한다.

프레임워크 기능 명세서 세부 요소 다음과 같다.

- ◆ 프레임워크 이름
- ◆ 개발 영역 분류(J2EE Architecture Layer)
- ◆ 적용 가능한 디자인 패턴(표 1, 2, 3 참고)
- ◆ 출처
- ◆ 버전
- ◆ 참고문서
- ◆ 구현 컴포넌트(.jar)
- ◆ 사용성

다음은 Presentation Layer, Business Layer, Integration

Layer 등 J2EE Architecture 에서 영역에 해당하는 Design Patterns 이다.

<표 1> Presentation Layer Design Patterns

| 패턴명 | 개요 |
|------------------------|---|
| Intercepting Filter | 요청에 대한 사전 처리와 사후처리 |
| Front Controller | 요청 핸들링 관리하기 위한 중앙 집중 컨트롤러 제공 |
| Context Object | 상태를 여러 애플리케이션에서 공유할 수 있도록 특정 프로토콜에 한정되지 않는 방식으로 캡슐화 |
| Application Controller | 액션과 뷰 관리를 중앙 집중하고 모듈화 |
| View Helper | 프레젠테이션 모델 로직과 관련 없는 로직을 Helper 컴포넌트로 캡슐화 |
| Composite View | 최소 단위의 하위 컴포넌트를 집합 뷰로 생성 |
| Service to Works | 디스패처 컴포넌트를 Front Controller와 view Helper 패턴과 함께 조합 |
| Dispatcher View | Controller와 view Helper를 가지고 디스패처 컴포넌트를 형성하며 뷰 처리가 끝날때까지 여타 활동 지연 |

<표 2> Business Layer Design Patterns

| 패턴명 | 개요 |
|---------------------------|---|
| Business Delegate | 비즈니스 서비스 접근을 캡슐화 |
| Service Locator | 서비스와 컴포넌트 검색을 캡슐화 |
| Session Facade | 비즈니스 tier 컴포넌트를 캡슐화하고 원격 클라이언트에게 큰 단위 서비스(coarse-grained)를 제공 |
| Application Service | 단일한 서비스 레이어를 제공하기 위하여 동작을 하나로 모아 제공 |
| Business Object | 객체 모델을 이용하여 비즈니스 데이터와 로직을 서로 분리 |
| Composite Entity | 로컬 엔티티 빈과 POJO를 이용하여 Persistent Business Object를 구현 |
| Transfer Object | tier 사이의 데이터를 전송 |
| Transfer Object Assembler | 여러 데이터소스에서 복합 Transfer Object를 모음 |
| Value List Handler | 데이터 조회를 처리하고 결과를 임시 저장하며 결과 집합을 뒤져 필요한 항목을 선택하는 역할 |

<표 3> Integration Layer Design Patterns

| 패턴명 | 개요 |
|--------------------|-------------------------------------|
| Data Access Object | 영구 저장소 접근을 추상화하고 캡슐화 |
| Service Activator | 메시지를 수신하고 프로세스를 비동기 방식으로 호출 |
| Domain Store | Business Object를 위한 투명한 영구 메커니즘을 제공 |
| Web Service Broker | XML과 웹 프로토콜을 이용하여 하나 이상의 서비스 제공 |

3.3 프레임워크 배치

프레임워크 기능명세서를 기반으로 소프트웨어 개발에 필요한 프레임워크를 선정하여 배치한다. 이때 프레임워크간의 관계에 대한 설정이 명확하게 하고 작성되는 문서는 프레임워크 배치 명세서를 작성한다.

프레임워크 배치 명세서 세부 요소 다음과 같다.

- ◆ 프레임워크 이름
- ◆ 관련 컴포넌트
- ◆ 업무 영역
- ◆ 관련 라이브러리

3.4 프레임워크 역할 정의 및 문서화

배치된 프레임워크를 기반으로 프로젝트를 진행시 배포해야 하는 명세서를 정리하고 작성한다. 프레임워크 배치 명세서를 기반으로 세분화된 아키텍처 설계 명세서를 작성하고 개발환경 명세서, 프레임워크 관련 문서 관리, 용어 사전을 작성한다.

개발환경 명세서 세부 요소 다음과 같다.

- ◆ OS

- ◆ JVM
- ◆ Editor
- ◆ WAS(Web Application Server)
- ◆ Builder(Ant)
- ◆ 프레임워크
- ◆ UI 환경

프레임워크 관련 문서 관리 세부 요소 다음과 같다.

- ◆ 프레임워크 관련 라이브러리
- ◆ 프레임워크 관련 문서

용어 사전 세부 요소 다음과 같다.

- ◆ 용어명
- ◆ 용어내용
- ◆ 출처

4. 결론 및 향후 연구과제

이번 연구에서는 소프트웨어를 개발할 때 설계되는 아키텍처를 오픈 프레임워크를 이용하여 작성할 때 고려해야 하는 요소와 이후 작성해야 하는 문서의 표준안에 대한 연구를 하였다. 표준화된 문서안을 기반으로 소프트웨어를 개발 환경에 대한 아키텍처 구조에 관한 정보를 관리하면 개발 시 일관된 정보를 제공할 수 있고 향후 유지 보수시 이해를 쉽게 함으로 작업에 효율성을 높일 수 있다.

향후 오픈 프레임워크를 선정하기 위한 방안에 대한 연구와 좀더 상세한 문서표준안에 대한 연구 및 사례연구가 요구된다.

참고문헌

- [1] John Cheesman and John Daniels, UML Components, Addison Wesley, 2000
- [2] Eric Jendrock, Jennifer Ball, Debbie Carson, Ian Evans, Scott Fordin, Kim Haase, "The Java™ EE 5 Tutorial Third Edition", For Sun Java System Application Server Platform Edition 9
- [3] IEEE Std 1063-2001(Revision of IEEE Std 1063-1987), IEEE Standard for Software User Documentation
- [4] Deepak Alur, Dan Malks, John Crupi, "Core J2EE Patterns" Prentice Hall, 2003
- [5] 최일우, 류성열, 이남용 " J2EE 패턴기반의 컴포넌트개발설계", 한국전자거래학회 논문지 제 7 권 3 호, 2002.12
- [6] TTAS.KO-11.0030, 소프트웨어 컴포넌트 개발 산출물(standard for software component development artifacts specification), 2003.12
- [7] 배정미, RUP 테일러링을 통한 J2EE 플랫폼의 소프트웨어 아키텍처 설계, 숙명여자대학교 대학원, 2006.02
- [8] 김승권, 이종하, 오픈소스를 이용한 시스템 통합, 마이크로소프트웨어, 2004.08
- [9] TOGAF, <http://www.opengroup.org/togaf/>, 2004.
- [10] J2EE Architecture, <http://java.sun.com>
- [11] Open Source Framework, <http://kldp.net/>
- [12] Open Source Framework, <http://sourceforge.net/>
- [13] Open Source Framework, <http://www.apache.org/>