

JavaEE5 対応 Web アプリケーションフレームワークの開発

原田雅史[†] 浅見可津志[‡] 土屋 隆[‡]

三菱電機情報技術総合研究所[†] 三菱電機インフォメーションシステムズ生産技術本部[‡]

1. はじめに

Web システムが、技術革新により従来の情報系システムから企業などの基幹系システムの構築に用いられるようになり、顧客から高品質・低コスト・短納期での構築が求められるようになった。我々は、これらの要求を満たすために早期からデファクトスタンダードである Struts をベースに採用したフレームワークの開発に取り組んでいた。フレームワークは、多数の Web システム構築に使用され 10%~30%程度の生産性向上を達成している。

JavaEE(Java Platform Enterprise Edition)5 は、2006 年 5 月に仕様が策定されてから 4 年余り経過し、エンタープライズシステム構築のための標準化された基盤技術として成熟すると共に各ベンダーも実装を提供している。今後、Web システムを構築するための基盤技術として JavaEE が主流になると判断し、従来の Struts ベースのフレームワークの後継として JavaEE5 に対応した Web アプリケーションフレームワークを開発した。

2. JavaEE の特長

JavaEE の前版は、J2EE (Java 2 Enterprise Edition)であり、企業の業務システムや電子商取引システムなどを構築するためのサーバ側の機能に関する標準仕様群である。しかし、高機能な仕様ほど使用方法が複雑で処理性能も遅いといった問題点があった。

これらの反省点を踏まえ JavaEE では、開発容易性と軽量化に重点が置かれた仕様に変更されており、この考え方は、2009 年 12 月に仕様が策定された JavaEE6 にも引き継がれている。

更に、Web システムの標準アーキテクチャである MVC アーキテクチャを高機能な仕様でカバーすると共に、Web サービスなどの新しい技術にも対応

した。また、各種 OSS(Open Source Software)製品も JavaEE に準拠した実装を提供するようになった。

3. JavaEE5 対応フレームワーク

JavaEE5 対応フレームワークは、JavaEE5 を実装したアプリケーションサーバ(JBoss など)、DI(Dependency Injection)コンテナなどを含むベース部分と我々が業務システム構築用に機能追加を行った拡張部分から成る“実行環境”、“開発環境”および“開発プロセス”から構成される(図1.を参照)。本稿では、“実行環境”について述べる。

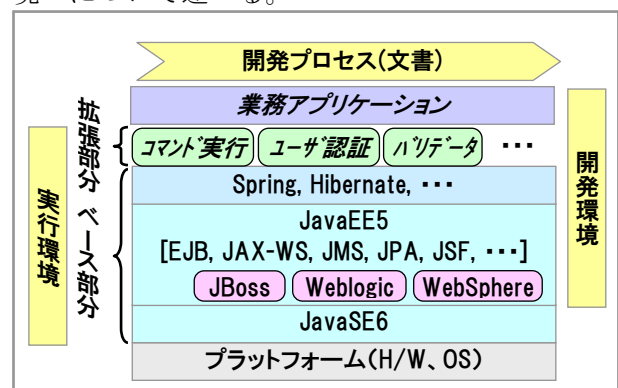


図1. フレームワークの全体構成

3. 1 ベース部分

Web システムの各レイヤーで使用する主な項目は以下である。

□プレゼンテーション層

Struts に対応する JSF を採用し Facelets を用いて実装する。

□ビジネス層

JSF との親和性が高い DI コンテナの Spring を採用し、POJO で実装する。

□データアクセス層

O/R マッパーの標準仕様である JPA を採用し、自製の DAO コード自動生成ツールと連携する。

□システム連携

EJB(セッション Bean)や JMS(Message Driven Bean)をファサードとして使用する。

[†]A Development of the Web Application Framework based on JavaEE5

[†] Masafumi Harada · Mitsubishi Electric Corporation Information Technology R&D Center

[‡] Katsushi Asami, Takashi Tsuchiya · Mitsubishi Electric Information Systems Corporation

3. 2 拡張部分

業務システムを効率よく開発するために、標準では提供されていない機能を拡張部分として開発した。主なものを記載する。

(1) コマンド実行機能

ビジネスロジックを“コマンド”と呼ばれるコンポーネントで実装し、それらを組み合わせで実行する機能である。コンポーネントの組み合わせ方は、XML ファイルに記述する。“コマンド”は、真偽値を返すため例えば、以下の処理を実行することができる。

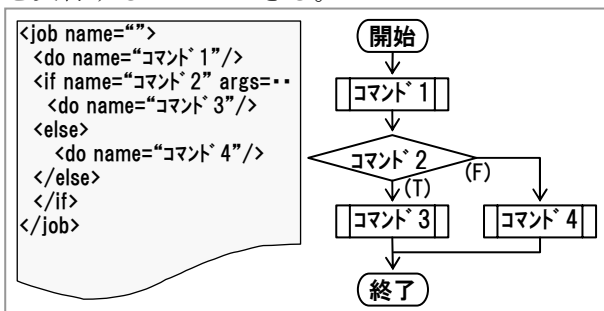


図2. “コマンド”を組み合わせた処理の例

(2) ユーザ認証機能

ユーザからのリクエストが、認証済みのユーザから送信されたものかどうかをチェックし、認証済みの場合にだけ処理を実行し、そうでない場合、指定された画面（例えば、ログイン画面）へ遷移させる機能である。

(3) 日本語対応のバリデータ/コンバータ

日本語に対応したバリデータ（全角文字や半角カタカナの入力値検証、郵便番号や電話番号など日本固有の形式を持つ入力値の検証）やコンバータ（例えば、全角文字⇄半角文字など）の提供。

上記以外に“2重要求抑止機能”，“ブラウザ バック対策機能”，“一覧データのページ切り替え表示機能”，“ファイル アップロード機能”などを提供している。

3. 3 現行のフレームワークからの移行

最後に Struts を使用している現行のフレームワークから JSF を使用している JavaEE5 対応フレームワークへの移行について述べる。仕様上、JSF と Struts は大きく異なるが開発者が同じなため概念が類似しており、個々の構成要素を対応付けることができる（表 1. を参照）。アクションクラスから呼び出される JavaBeans や EJB は、マネージド Bean からも呼び出すこと

が可能であり GUI の記述能力は JSF の方が高いため、表 1. に記載されている下線と斜体の項目を移行すればよい。

表1. JSF と Struts の構成要素の比較

JSF	Struts
<u>Faces</u> サブレット	<u>アクション</u> サブレット
<u>LifeCycle</u> クラス	<u>リクエスト</u> プロセッサ
↑	<u>アクション</u> フォワード
<u>JSF</u> タグ	<u>Struts</u> タグ
<u>UI</u> コンポーネント	—
<u>JSF</u> 設定ファイル	<u>Struts</u> 設定ファイル
<u>マネージド</u> Bean	<u>アクション</u> フォーム Bean
↑	<u>アクション</u> クラス

(注) 下線はプログラマが作成時に使用する項目、斜体はプログラマが作成する項目。

4. おわりに

本稿では、JavaEE5 に対応した Web アプリケーションフレームワークを紹介した。

我々は、これまで Struts をベースにしたフレームワークを開発し、実システム構築に適用することで生産性や品質の向上を実現してきた。その過程でフレームワークとして“実行環境”だけでなく、Eclipse やソースコード自動生成ツールなどの“開発環境”と“開発プロセス”も併せて提供することでより一層の効果を発揮することが経験的に得られている。

今後は、新しいフレームワークに対応した“開発環境”を引き続き整備していくと共に、実プロジェクトに適用することで生産性や品質に関する評価を行う予定である。更に、フレームワークを適用するシステム特性（例えば、ミッションクリティカル性）に合わせた機能拡張を実施する予定である。

参考文献

- [1] JavaEE5 仕様: <http://jcp.org/aboutJava/communityprocess/final/jsr244/index.html>
- [2] JSF 1.2 仕様: <http://jcp.org/aboutJava/communityprocess/final/jsr252/index.html>
- [3] Enterprise JavaBeans 3.0 仕様: <http://jcp.org/aboutJava/communityprocess/final/jsr220/index.html>
- [4] 斉藤賢哉: マスタリング JavaEE5 第2版, 翔泳社(2009)