

# Web コンピューティングシステムにおけるフレームワーク開発と適用事例紹介

五十嵐政志<sup>†</sup> 土屋隆<sup>†</sup> 吉村礼子<sup>†</sup> 原田雅史<sup>†</sup>

三菱電機株式会社<sup>‡</sup>

## 1 はじめに

Java や Web アプリケーションサーバを使った Web コンピューティングシステムの業務アプリケーション構築の生産性を向上させるために、数年前から以下を狙いとするアプリケーションフレームワークの開発を行っている。

- ・業務部品の再利用性の向上  
業務を部品化する事により、再利用性を向上させる。
- ・ミドルウェア利用技術のノウハウの隠蔽  
Web コンピューティングシステム開発ではいくつかのミドルウェアが利用される。ミドルウェアを使いこなすには時間がかかり熟練が必要である。正しく使わないと、いろいろな問題を引き起こす。このような、ミドルウェアを正しく使いこなすノウハウをフレームワークの中に隠蔽して、業務アプリケーション設計の担当者が詳細なノウハウを知らなくてもミドルウェアを使いこなせるようにする。

本稿では、このようなフレームワークの概要と適用事例について述べる。

## 2 フレームワークの概要

今回紹介するフレームワークを利用したソフトウェア構成例を図 1 に示す。この図の密度の濃い網掛けがフレームワークを示しており、以下の 2 種類のフレームワークが含まれている。

- ・Web システム向け MVC(Model View Controller)フレームワーク  
Java アプリケーションサーバ上の Web アプリケーションを構成する MVC モデルを使ったフレームワーク。
- ・システム間連携 XML フレームワーク  
XML を使ってシステム間連携を行うためのフレームワーク

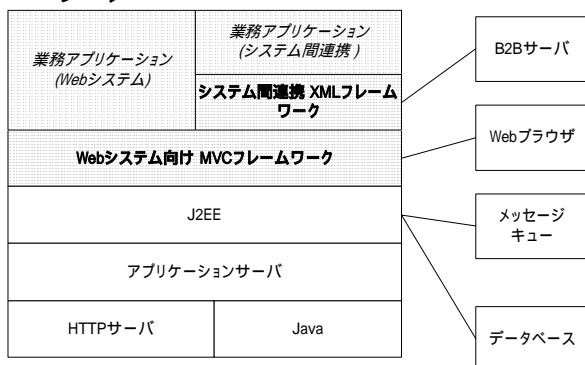


図 1 フレームワークを利用したソフトウェア構成例

Developing Framework and Introduction of its Application in Web Computing System

<sup>†</sup>Masashi Igarashi, Takashi Tsuchiya, Ayako Yoshimura, Masafumi Harada

<sup>‡</sup>Mitsubishi Electric Corporation

## 2.1 Web システム向け MVC フレームワーク

Web システム向けフレームワークの構成を図 2 に示す。

このフレームワークの特長は以下の通りである。

- ・基本的なアーキテクチャとして、MVC モデルを採用している。このため業務アプリ設計者は、主に Model のビジネスロジックコンポーネントと、データアクセスコンポーネント、および、View の JSP(Java Server Pages)を開発するだけで、システムを構築する事ができる。
- ・サーバソフトウェアのコンポーネント化のために、EJB を採用し、EJB の基底クラスとユーティリティクラスを提供して、フレームワークが EJB を扱う方法を標準化している。これにより、業務アプリ開発者が EJB を意識せず簡単に扱える。そして EJB のメリット(トランザクション管理やセキュリティに関するコードの業務ロジックからの排除 etc.)を利用することができる。
- ・Controller では、図 2 に示すようにデータベース上の定義(実行定義、処理定義)に従って、状態遷移制御と Command パターンを組み合わせたコンポーネントの実行制御によりアプリケーションを実現している。このため、データベース上の定義を変更するだけで、コンポーネントの処理順序を変更でき、アプリケーション処理内容の変更にプログラムコードの変更が必要ない。

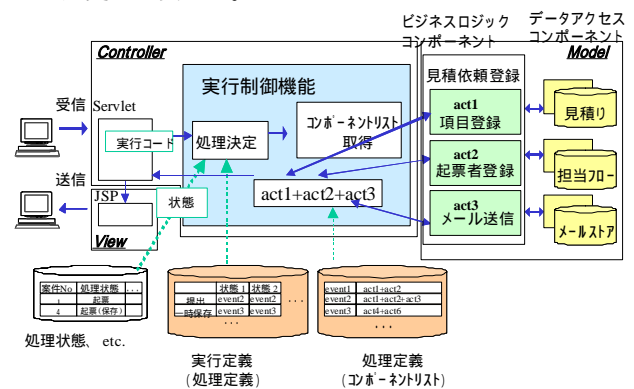


図 2 Web システム向け MVC フレームワークの概要

## 2.2 システム間連携 XML フレームワーク

図 3 にシステム間連携 XML フレームワークの構成を示す。このシステムの特長を以下に示す。

- ・業務アプリケーション開発者は、XML メッセージを処理するために 2 つの設定を行うだけでよく、複雑な XMLParserAPI を扱う必要がない。設定内容は、"(1)送受信情報の設定を行うロジックの開発"、"(2)設定された情報を送受信する XML メッセージフォーマットへのマッピングの設定"である。
- ・このように設定を 2 つに分けた事により、XML メッセージのフォーマットの変更や、別のフォーマット

に柔軟に対応できる。

- ・ 図 3 のシステム間連携 XML フレームワークは、Web システム向けフレームワーク上のアプリケーションとして構成されており、XML 送受信の際に実行される業務部品を共用できる。例えば、Web ブラウザからの見積処理と、取引先のシステムからの見積 XML メッセージにより実行される見積処理を共用できる。

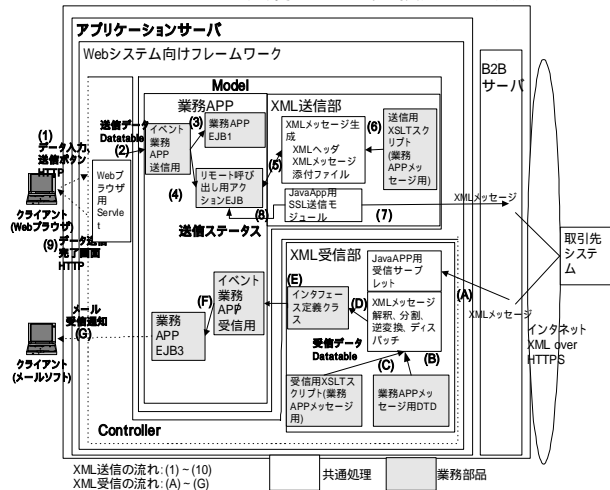


図 3 システム間連携 XML フレームワーク  
3 フレームワークの適用事例

### 3.1 事例 1 「WebEDI 資材見積りシステム」

#### (1) 業務アプリケーションの概要

図 4 に示すように Web ブラウザおよびメールを使って、社内の設計担当者・資材担当者、および取引先の担当者間で資材見積り業務を行う。

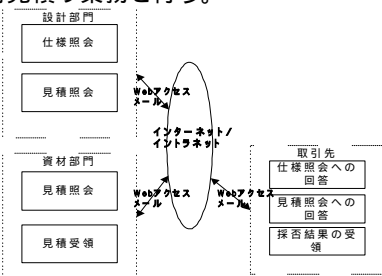


図 4 WebEDI 資材見積りシステム

#### (2) 課題と解決策

図 4 のシステムに Web システム向け MVC フレームワークを適用してシステムを構成した。

- ・ 業務アプリ設計担当者は、オブジェクト指向、Java、アプリケーションサーバ共未経験だったが、業務アプリケーションを設計・製造する事ができた。
- ・ 業務システムでは、業務状態の制御が必要である。図 4 のシステムにおいても見積り案件に対する「見積回答前」、「回答済み」などの状態の管理が必要となった。これを、フレームワークの Controller の提供する状態遷移制御により管理する事ができた。
- ・ フレームワークの開発の際に、パイロットプロジェクトにより、業務システムに適用する際の検証を行った。そして、実プロジェクトに適用する前に、アプリケーションサーバを使う際に生じる問題 (性能問題、漢字文字化問題) などが判明し、フレームワ

ークに対策のための機能を盛り込む事ができた。

- ・ 試験段階でアプリケーションサーバが原因の不具合が生じて、プログラムコードによる対策が必要になった。しかし、フレームワークの提供する共通部品でこれを吸収する事ができ、業務アプリケーションを修正しなくても問題を解決する事ができた。

### 3.2 事例 2 「E マーケットプレース実証実験システム」

#### (1) 適用した業務アプリケーションの概要

国をまたがった HUB 構成の E マーケットプレースの実証実験システム。このシステムでは業務として、Web ブラウザから企業検索 / 見積もり / 受発注 / 公開公募などの業務を行う。これらの業務で、他の E マーケットプレースに所属する取引先が指定された場合にのみ、E マーケットプレース同士の XML を使ったシステム間連携が行われる。例えば MP1 に属する企業 E11 が、MP2 に属する企業 E21 と見積りを行う場合には、国 1 HUB1 を通じて XML によるシステム間連携が行われる。

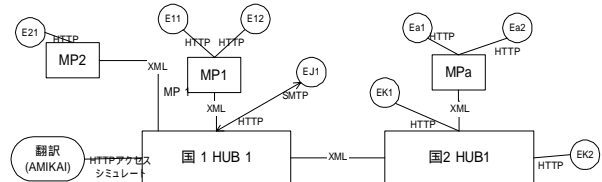


図 5 E マーケットプレース実証実験システム

#### (2) 課題と解決策

図 5 のシステムに対して Web システム向け MVC フレームワークとシステム間連携 XML フレームワークを適用した。

- ・ フレームワークが提供する XML メッセージを操作する部品により、XML Parser を直接使う場合の冗長なコードを削減する事ができ、9 種類の XML メッセージを処理する業務ロジックの製造を 2 週間程度で行う事ができた。
- ・ マーケットプレース間の XML 通信業務プロトコルにおいて、到達保証のために後から ACK メッセージ返信の開発が必要になった。これらの変更を、ACK メッセージに対するフレームワークの (1), (2) の定義と、処理定義の変更だけで行う事ができ、開発量を最小にする事ができた。
- ・ 同じ種類の業務を Web ブラウザから行う場合と、XML メッセージで他のシステムから行う場合に、共通の業務部品を使う事ができた。

## 4 まとめ

「Web システム向け MVC フレームワーク」は、他に 4 つのプロジェクトに適用した。

「システム間連携 XML フレームワーク」は、現在、EAI/B2B システム向けミドルウェアとして、ブラッシュアップを行っている。フレームワークを使う事により、開発量の削減や、ミドルウェアの使い方の誤りによる不具合の低減などの効果を確認する事ができた。

## 参考文献

- [1] Ralph E. Johnson、中村弘明、中山裕子、吉田和樹、  
「パターンとフレームワーク」、共立出版