

標準フレームワークを利用したJava開発

Javaアプリケーションサーバーを用いたWebシステムの開発案件が急増している。これらの案件では、開発に高い品質とスピードが求められる一方で、熟練した技術者が必須という課題もある。本稿では、このような課題を解決し、開発をサポートする技術的な仕組みであるNRI（野村総合研究所）のフレームワークについて紹介する。

Javaアプリケーションサーバーの課題

Webシステムの開発において、WebLogic（BEA Systems社の製品）やWebSphere（IBM社の製品）などのJavaアプリケーションサーバーと呼ばれるソフトウェア群を利用する事例が急増している。Javaアプリケーションサーバーを利用する大きな理由は、これによって開発されたシステムの「耐障害性」と「移植性」の高さにある。Javaは言語レベルでこの2つをサポートしており、Javaアプリケーションサーバーを適切に利用すれば、高品質のシステムを短期間で構築することができるのである。

一方、Javaを用いたシステム開発には、オブジェクト指向技術に習熟した技術者が必須というのが定説である。そのため、技術者の確保ができないという理由で、Javaアプリケーションサーバーの導入に二の足を踏む例も多かった。

オブジェクト指向技術適用時の課題解決方法

NRIでは、長年にわたって業務系システムの開発にオブジェクト指向技術を利用してきた経験から、とくに中規模から大規模な業務システム開発では、「熟練者がフレームワー

ク（全体構造を規定する部品）を開発し、通常の技術者（オブジェクト技術未経験者）がそれを用いてアプリケーションを開発する」という方法が、テストや維持管理のフェーズを通じて最も効率的であると考えている。

このとき、フレームワークに業務寄りの機能を含め過ぎると、新規開発者の教育のコストが高くなり、またシステム開発の後期の工程で熟練者に作業が集中してしまうという問題が起きる。そこで、フレームワークでは「画面表示・遷移処理」、「チェック処理などの業務共通処理」、「障害処理などの基盤系処理」、「電文編集処理」などの汎用機能を提供し、個別の業務処理はアプリケーションで実装するのが効果的である。このようなフレームワークを利用すれば、オブジェクト指向技術に習熟した技術者の数が少なくても、Javaアプリケーションサーバーの導入が容易となる。

また、Webシステム開発時には、画面処理と業務処理が混在した保守性の悪いシステムになりやすい、ブラウザの「戻る」ボタンなどでエンドユーザーが画面遷移を勝手に変えてしまうような操作に対応しなければならない、といったWebシステムに特有の課題もある。この課題もこのようなフレーム

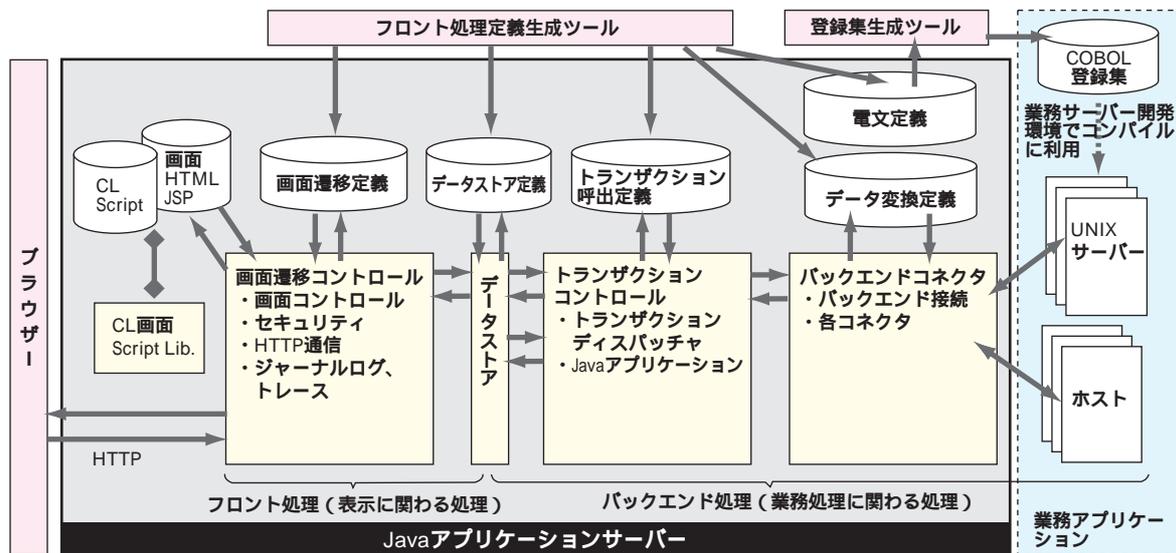


図1 フレームワーク全体図

ワークで解決されていることが望ましい。

Web開発を容易にするフレームワーク

これらの課題を解決するために、NRIは、Javaアプリケーションサーバー用のフレームワークを開発した(図1参照)。このフレームワークは、プログラムを書くことなく、既存システムへ接続できる機能も持っている。また、「オブジェクトワークス」の一部として製品化もされている。このフレームワークの特長は以下のようなものである。

Webシステムのフロント処理を開発するには、HTML(ハイパーテキストマークアップ言語)形式のデータで画面を作成するとともに、画面遷移や、バックエンド処理に引き渡すデータ(データストア)を表形式で定義するだけでよい。

Webシステムのバックエンド処理を開発するには、純粋に業務処理のみを記述すればよい。

障害処理はフレームワークで吸収するが、詳細はプロジェクトごとにカスタマイズ可能である。

このフレームワークは開発のスピードアップに貢献しているほか、開発手順・プログラム構造が標準化されるため、修正が必要な部分がわかりやすく、外部の開発者に作業が依頼しやすいこと、障害処理がフレームワークに組み込まれているため耐障害性が高いこと、などが実証されている。

以上のようなフレームワーク・パッケージの利用は、今後ますます広がっていくに違いない。

(野村総合研究所 並河英二)