

官公庁システムのフレームワーク適用事例

Adaptation of Frameworks for Development of Public Office Systems

あらまし

官公庁における情報システムは、従来、メインフレームを中心として開発されてきたが、近年はシステムのライフサイクルコストの問題、国民への公共サービス拡大やユビキタス社会への対応の観点より、Webシステムでの開発を行うことが主流となっている。Webシステムにおけるアプリケーションの開発は低コスト、短納期および高品質が求められており、その解決手段の一つとしてフレームワークの適用があげられる。

本稿では、富士通が官公庁で実施した業務システムのWebアプリケーション開発を行う中で、J2EEに準拠したフレームワーク製品である“Interstage Application Framework Suite”の提供する機能を拡張した「個別フレームワーク」の開発を行い適用することにより、開発コストの低減や納期短縮を実現した効果について紹介する。

Abstract

In the development of information systems for public offices, development based on the Web system is now a mainstream system development activity in terms of addressing the problem of system lifecycle costs to expanding public services to people and the ubiquitous society, although such systems have conventionally been developed from mainframes. Moreover, applications developed on the Web system must meet the requirements of low cost, short-term delivery, and high quality. The adaptation of certain frameworks has been cited given as one approach to a solution. This paper introduces the effects achieved by Fujitsu in reducing development costs and shortening the delivery of Web applications for the service systems of public offices by adapting an “individual framework” which enhances the functions of the “Interstage Application Framework Suite,” a framework product that conforms to J2EE.



井上均史 (いのうえ ひとし)
第一ソリューション統括部 所属
現在、官庁系の業務システムの運用
支援、開発に従事。



中西広樹 (なかにし ひろき)
第一ソリューション統括部 所属
現在、官庁系の業務システムにおけ
る運用支援に従事。

まえがき

官公庁における情報システムの開発では、ホストコンピュータを中心としたシステムを形成してきたが、近年はライフサイクルコストの観点からシステムをダウンサイジングする傾向にある。それに加え、申請・届出の電子化といった国民への公共サービス拡大やユビキタス社会への対応の観点から、Webシステムによる開発が主流となってきている。

Webシステムの開発においては、官公庁で主流となっているホストコンピュータ中心のシステム開発のノウハウでは生産性向上や品質向上に対する効果が得られないのが現状であると同時に、JavaエンジニアやWebアプリケーション開発技術者の慢性的な人材不足という問題点がある。

本稿では、某官公庁の大規模業務システムの開発において、SDASの概念に基づいた生産性向上や品質向上を実現するフレームワークを適用した事例を紹介する。

システム開発要件と設計方針

本システムの構築に当たり、某官公庁より、申請者が書面で提出していた申請・届出を電子化することに加え、職員の利便性を考慮し、ホストコンピュータなどで管理している既存のバックエンドシステムと連携するために、以下の要件が挙げられた。

- (1) 金銭の自動振込を行うため、高品質であること。
- (2) 業務特性から類似した画面が多いこと。
- (3) 業務の特性上、法改正などによるシステムの

変更があることが想定されるため、メンテナンス性に優れていること。

- (4) セキュリティ要件などからシステムの構成はWebサーバ・アプリケーションサーバ（APサーバ）・DBサーバの3階層システムであること。

上記の要件により、複雑なシステムをいかに短納期（18箇月で総合運転試験完了）で効率良く開発するかが課題であった。その効率化を図る手段の一つとして「個別フレームワーク」を開発することとした。個別フレームワークの基本フレームワークはInterstage Application Framework Suite⁽¹⁾（以下、IAFS）を利用することを前提に、以下の設計方針を確立した。

- (1) IAFSが提供しているフレームワーク範囲を拡張する。
- (2) 3階層のうちWeb層のアプリケーション開発は不要とする。
- (3) IAFSで個別に開発する機能をグループ化することにより、簡易なインタフェースを開発者に提供し、メンテナンス性に優れた仕組みとする。
- (4) フレーム分割画面の柔軟な制御を行えるようにする。

個別フレームワークの機能

本章では前述の開発方針に基づいて開発された個別フレームワークの概要を説明する。本フレームワークは以下の機能ごとに分類されている（図-1）。

- (1) スーパーWebハンドラ

3階層システムを開発する場合、アプリケーションサーバに業務アプリケーションを構築し、Web

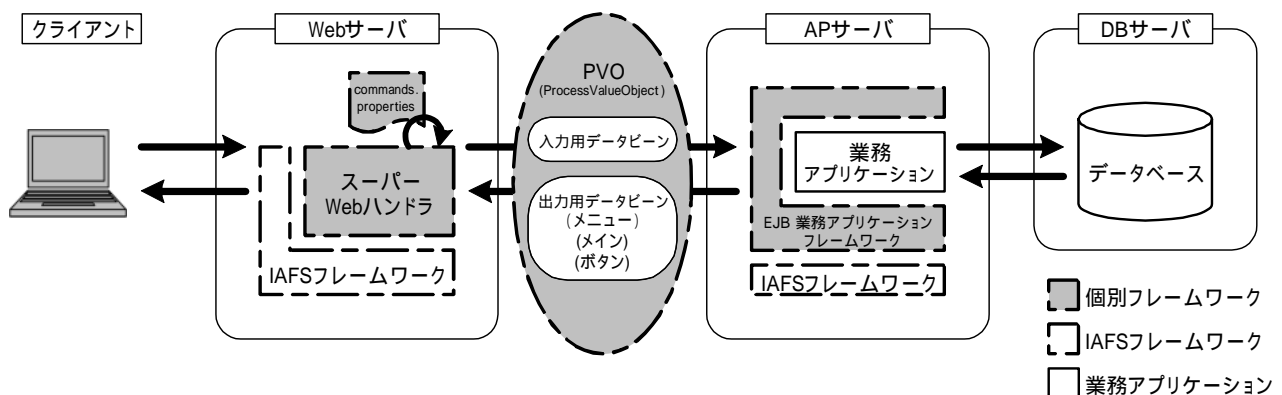


図-1 フレームワークの概要
Fig.1-Outline of framework.

サーバには、入力データの属性チェックや画面制御を行うアプリケーション（Webハンドラ）を実装するのが一般的であるが、類似した画面が多く、高品質が求められる。そのため、画面ごとの開発規模の増加および品質のばらつきを抑止することを目的とし、IAFSのWebハンドラ機能を拡張した。

この機能は、画面ごとのWebハンドラ実装を省略し、属性チェック定義体などを作成するのみで、以下の機能を実現する（図-2）。

- ・属性チェック
- ・データビーンからEJB（Enterprise JavaBeans）業務アプリケーションへの情報移送
- ・EJB業務アプリケーションの呼出し
- ・画面制御

(2) EJB業務アプリケーションフレームワーク

業務システムを作成する際、共通に使用する部品である業務アプリケーションを構築するためのスーパークラスを提供しており、以下の機能を実現する。

- ・データベースのコネクション/クローズ
- ・データベースのコミット/ロールバック
- ・ログの活性化/有効化

(3) PVOインタフェース

Process Value Object（以下、PVO）は、Webサーバとアプリケーションサーバ間で連携するために使用するValue Objectであり、今回の個別フレームワークで、スーパーWebハンドラとEJB業務アプリケーションフレームワークとのインタフェースに

利用するため開発を行ったものである。

実現機能紹介

本章では、実際のシステムで本フレームワークにより実現した機能を紹介する。

(1) 画面遷移の実現

本システムでは、操作性の向上のためにメニューやボタンなどの位置固定により、フレーム分割やフレームの組合せが複数パターン存在する。このような様々な分割画面パターンへの画面遷移を可能とするため、スーパーWebハンドラに実装しているプロパティファイルに処理ごとの遷移パターンコードを記述する。これにより、画面遷移時にフレーム分割パターンを指定する仕組みを提供した（図-3）。

(2) 複数フレームの同時更新機能の実現

複数フレームを使用した画面設計を行った場合、各フレームの内容の同期が必要となり、複数フレームを同時に更新しなければならない。そこで、本フレームワークでは、PVOインタフェースの出力データビーンに更新機能を実装することにより同時更新を可能とした（図-4）。

IAFSはPDFや電子帳票を含んだ複数画面の同時更新をサポートしていないが、機能の拡張を行うことにより、アプリケーション開発側で意識することなく、PVOインタフェースの出力データビーンに更新機能を実装するだけで同時更新できる仕組みとした。

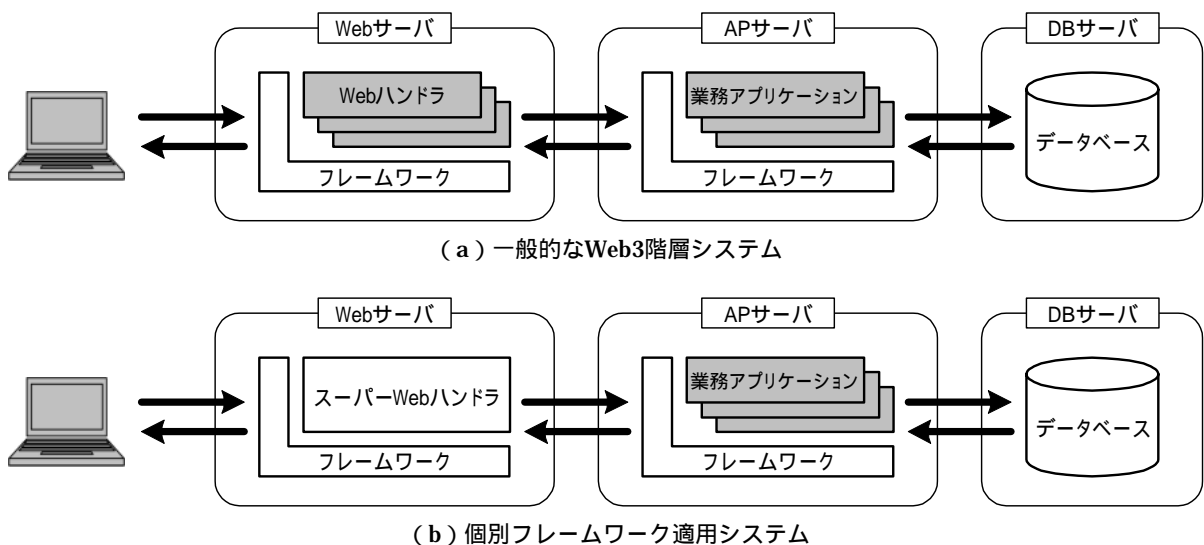


図-2 スーパーWebハンドラ
Fig.2-Super Web Handler.

(3) 画面遷移情報の履歴管理機能の実現

本フレームワークでは、画面遷移時にPVOを保存しておき、必要に応じて保存されたPVOから画面の生成を行うことが可能な画面の履歴管理を実現している。PVOには複数フレーム分のデータビーンが格納されているため、複数フレームで構成された画面においても、単一フレーム画面のときと同様な操作で履歴画面を表示することができる。

(4) 開発/テスト環境の簡略化の実現

Webシステムのアプリケーション開発を行う場合、一つの環境で複数の開発者が同時に開発/テストを行うと相互に与える影響が大きいことから、個別の環境を準備する必要がある。とくに、3階層システムの場合、開発工程ごとにWebサーバ、アプリケーションサーバを準備する必要があるが、サーバ内に

複数の開発環境を準備しても、開発/テストの簡略化には限界がある。

この問題を解決するため、本フレームワークでは、開発工程の簡易な変更を行った。Webサーバ内に業務アプリケーションを配備して(アプリケーションサーバを用意せず)、システム開発環境の準備の効率化を保證する仕組みを実現した。本フレームワーク独自のプロパティファイル(commands.properties)にデータビーンの中に呼び出されるEJBアプリケーション名とアクションクラス名を記述し、スーパーWebハンドラが、このファイルを参照することによりデータビーンに対応する業務アプリケーションを自動的に呼び出している。同プロパティファイル内のEJBアプリケーション名を省略するだけで、Webサーバ内の業務アプリケーションを呼び出す仕組みを提供していることから、Webサーバのみで業務アプリケーションの開発/テストを実施し、開発やテストの簡略化を実現した(図-1)。

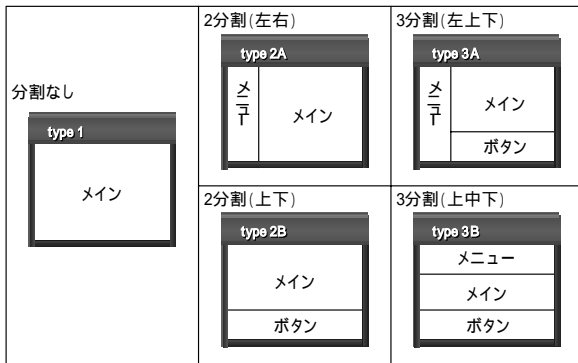


図-3 分割画面パターン
Fig.3-Multi-Frame pattern.

適用効果

上記の機能をフレームワークで提供することにより以下のような効果があった。

(1) 生産性向上の実現

本フレームワークを適用することで、適用前の各業務で個別に開発した結果に比べて、システムの開発規模を約3割削減することができ、生産性を向上した。

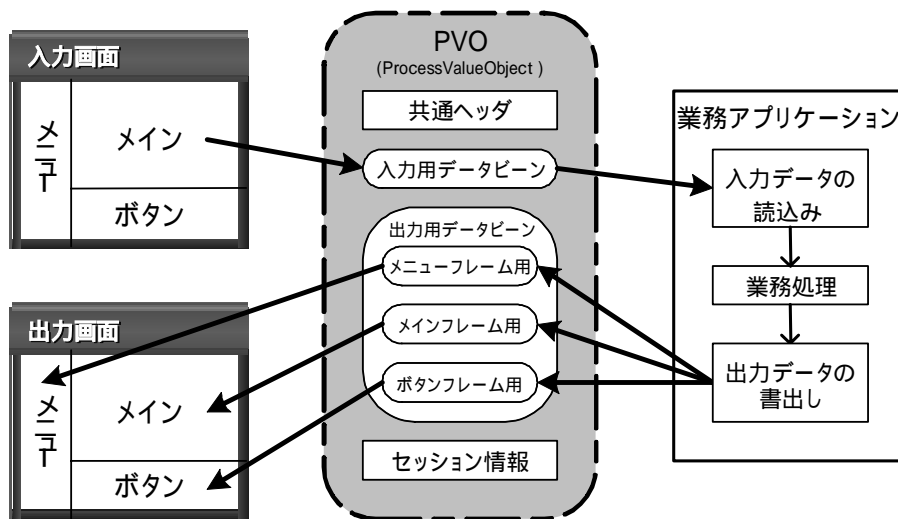


図-4 画面の同時更新イメージ
Fig.4-Image of multi-frame update at once.

(2) 開発/テスト環境の効率化

本フレームワークでは、開発環境をWebサーバとアプリケーションサーバおよびDBサーバの3階層のシステムを連携（同一環境）することにより、開発/テストの効率化を実現することができた。

(3) メンテナンス性の向上

本フレームワークは、業務ごとに開発している部分を1箇所に集約することにより、開発中に発生した制度見直しによる仕様変更などに柔軟に対応し、納期を確保することにつながった。

む す び

本稿では、大規模な業務システムのWebアプリケーションの開発を短納期で効率的に行う手段として、開発効率やメンテナンス性を考慮し、開発技術

者の要員不足を補うことを目的としたフレームワークの適用を述べた。

業務システムの開発においては、開発の背景や求められる成果によって様々な対応が必要になってくる。そこで、本フレームワークは、他システムへも適用できるように、汎用性を持たせている。今後の運用フェーズでは、さらなる改善項目を検討し、IAFSと本フレームワークとの組合せを標準化することで、他システムへ展開していきたいと考えている。

参考文献

- (1) 富士通：J2EEアプリケーションフレームワーク
Interstage Application Framework Suite .
<http://interstage.fujitsu.com/jp/framework/index.html>

