

システムリフレッシュを支える IT 基盤整備

IT Infrastructure to Support System Refreshing

原田 敬太 HARADA Keita
加島 靖蔵 KASHIMA Seizo

JFE スチール IT 改革推進部長
JFE システムズ 東京事業所 販生流システム開発部基盤グループ長

要旨

NKK と川崎製鉄の経営統合による JFE グループ設立に合わせて構築した J-Smile™ (販売・生産・物流) システムでは、独自の Java アプリケーションフレームワークを構築し、開発生産性向上、構築システムの品質向上、コンピュータハードウェアからの独立性確保を図った。その後の製鉄所、グループ会社のシステムリフレッシュにあたり、この J-Smile™ のフレームワークを活用することで、既存資産の再利用を促し、低コスト・短期間でのシステム構築を目指した。具体的には、異なる種類のコンピューター上で稼動するようにフレームワークを改良し、アプリケーションプログラムの移植性・再利用性を向上させた。

Abstract:

In conjunction of the formation of JFE Group, an original Java application framework in the J-Smile™ (Sales, Production, and Distribution) system was constructed. It improved the development productivity, brought a high-quality system, and achieved independence of the computer hardware. Since then, this framework of J-Smile™ was used when the systems of ironworks and group companies were refreshed, aiming at short term and low cost system construction. For smooth applications, the framework was revised to be operate on a different kind of computer, and secured the portability of the application system.

1. はじめに

2003 年 4 月、NKK と川崎製鉄が経営統合し JFE グループが誕生した。新会社の設立に合わせて構築した新統合システム「J-Smile™」¹⁾ は大規模な Web システムとして 1 300 万 step の Java プログラムを保有し、現在も新機能の開発、改善を行っている。J-Smile™ の開発に当たっては、後続する各製鉄所、グループ会社の Java を用いたさまざまなシステム開発に対して、資産の流用性を高める仕組みが求められた。そのためには、アプリケーションシステムの構造を標準化することが必須であり、加えて品質の向上、開発生産性の向上を図るためにも共通のアプリケーションフレームワークを構築する必要があった。

2. 共通フレームワークの構築方針

共通フレームワークの構築に当たっては、プラットフォームとの独立性を追求した。プラットフォームとの独立性として考慮すべき点は下記の 2 点である。

- (1) 異なるオペレーティングシステム (OS) での稼動
- (2) 異なるベンダーのミドルウェア製品上での稼動

2.1. 異なるオペレーティングシステムでの稼動

J-Smile™ 構築時に開発したフレームワークは International Business Machines Corp. (IBM) のミドルウェア製品上で稼動することを前提としていた。具体的には、アプリケーションサーバーとしての WAS (WebSphere Application Server) とデータベースとしての DB2 (Database2) である。そして、その前提の元で、異なるオペレーティングシステムでの稼動はすでに実現していた。

各プログラムは Windows パソコン上で開発を行い、UNIX サーバー上でシステムテストを行う。最後にメインフレーム環境で運用テストを実施し、そのままメインフレーム上で本番稼動させている。アプリケーションプログラムには手を加えることなく、いずれの環境下でも等価に稼動するポータビリティをアプリケーションフレームワークで実現していた。

2.2. 異なるベンダーのミドルウェア製品上での稼動

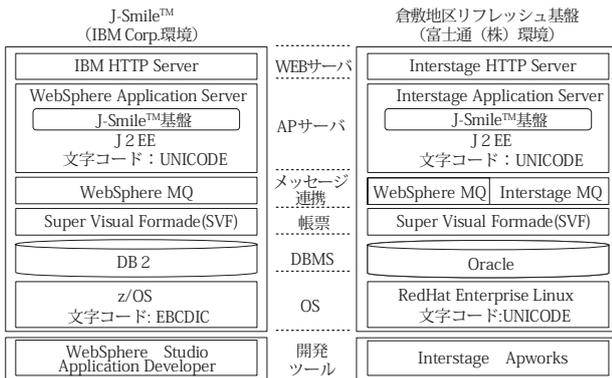
JFE スチールでは、製鉄所ごとに異なるベンダーのハード

ウェア・ソフトウェア製品群を利用して、倉敷地区、千葉地区、知多地区では、主に富士通（株）の製品を採用していた。これらの地区では、Java アプリケーションシステムの開発においても、既存システムとの親和性から、富士通（株）のミドルウェア製品の利用が必須となった。共通フレームワークとしては、富士通（株）の製品上でも同じように稼働し、しかもアプリケーションプログラムは、ミドルウェアの違いを意識しなくても良いようにする必要があった。

最初に富士通（株）の製品上でシステム構築することになった倉敷地区の厚板・薄板リフレッシュで選定したアプリケーションサーバーは Interstage、データベースは Symfaware であった。ただし、データベースとしての Symfaware は、フレームワーク稼働の鍵となる 2 フェーズコミットに当時対応していないという問題があった。データベースへの書き込みと待ち行列（キュー）への書き込みを実施する際に、両方成功した場合に確定（コミット）し、いずれかが失敗した場合には、元の状態に戻す必要があるが、Symfaware は対応できていなかったのである。よって、データベースとしては富士通（株）のハードウェアで稼働実績の豊富な Oracle Database を採用することにした。

図 1 に、共通フレームワークが稼働する、IBM Corp. 環境、富士通（株）環境の製品の対応を示す。

Interstage, WAS ともに J2EE (Java2 Enterprise Edition) の標準に準拠しているものの、キューのコミットのタイミングが異なることが事前検証で判明した。また、データベースとしての Oracle Database も排他制御の考え方などで DB2 との差異が何点かあった。



IBM, DB2, WebSphereは、IBM Corporationの商標である。
Java, J2EE Oracleは、Oracle Corporationの商標である。
Interstage, Symfawareは、富士通株式会社の登録商標である。
SVFは、1stホールディングス株式会社の登録商標である。

図 1 共通フレームワークの稼働環境の差異
Fig. 1 Difference of operation environment

IBM, DB2, WebSphereは、IBM Corporationの商標である。
Java, J2EE, Oracleは、Oracle Corporationの商標である。
Windowsは、Microsoft Corporationの米国およびその他の国における商標である。
UNIXは、The Open Groupの商標である。
Interstage, Symfawareは、富士通株式会社の登録商標である。

これらのミドルウェアの仕様の差異が個々のアプリケーションプログラムに影響を及ぼさないよう、フレームワークでそれらの差異を吸収することとした。これにより、同じプログラムが、OS やミドルウェアが異なってもそのまま稼働することができ、プラットフォームとの独立性を実現することができた。

3. 共通フレームワークの特徴と構築上の工夫点

3.1. フレームワーク構造の見直し

共通フレームワークの構築にあたっては、J-Smile™で開発したフレームワークの構造を見直すことにした。稼働環境の違いをパラメータにより外部から制御できるように組み替え、また大きく異なる主要部品については差し替えることができるように再構成した。また、この機に印刷機能、FAX機能などの周辺機能や他製品に依存する機能も必要に応じて組み合わせるよう部品形式に変更した。

スリムになったフレームワークは拡張の自由度が高まり、新機能の追加にも迅速に対応できることとなった。最近では添付文書を扱うことのできる文書管理機能も追加するなど、より付加価値の高い機能を改修の負荷をかけず取り入れることができるようになった。

これらの対応は図 2 に示すように新統合システムを構築する際のコンセプトであった「分離」と「局所化」、および「モデル重視」に基づく改修である。

3.2. 運用管理体制の構築

個々のシステム要件に合わせて個別に改善していくと異なるフレームワークに発散することが想定されるため、表 1 のようにフレームワーク利用における管理範囲、利用規定、作業分担を設け、フレームワークの構造、利用形態を統制することとした。

また、維持管理に必要な要員のスキルについても整理し、各地区にも必要な要員を育成しているところである。

このような運用体制を構築することで、共用フレームワークの全社一元管理を目指している。

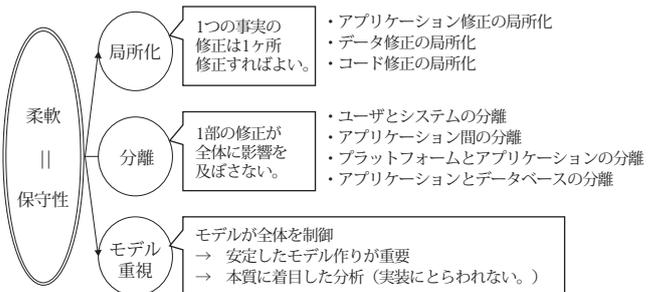


図 2 システム構造の柔軟性
Fig. 2 Flexible system structure

表 1 運用管理業務の役割分担
Table 1 Role of business operation

	障害	変更	調査・問い合わせ	運用
提供部署	フレームワーク調査/ 技術フォロー	主体的に行う ・変更可否 ・全社最適 ・全社広報	技術フォロー	初期導入のみサポート
利用部署	主体的に行う ・発見/一次切分 ・暫定対応 (暫定修正含む) ・障害管理	地区固有部分をフォロー ・地区内調査/広報 ・地区環境テスト	主体的に行う ・環境の保有 ・製造販売会社と連携 ・アプリケーションと連携	主体的に行う
利用部署に 具備すべき案件	↓ フレームワーク調査・ 緊急処置技術	↓ 利用部署環境下 での動作テスト	↓ 利用部署環境 での一次切分	↓ 利用部署環境に 合わせた運用設計

4. 共通フレームワークの効果と今後の課題

4.1. 共通フレームワークの効果

図3に示すように、共通フレームワークは、倉敷地区の厚板・薄板リフレッシュに加え、京浜地区の新酸洗、千葉地区の薄板、知多地区の鋼管など各製鉄所において活用が進んでおり、また、JFE 条鋼, JFE 鋼板, JFE 商事などグループ会社にも展開している。

各システム構築プロジェクトでは、個別にフレームワークを作成する必要はなく、その保守も一元化された体制で実施している。また、アプリケーションプログラム資産の再利用も進んでおり、たとえば、仕様項目のマスタ管理プログラムは本社、複数の製鉄所で同じものが利用されている。

4.2. 今後の課題

共通フレームワークの今後の課題として2点述べる。

(1) J-Smile™フレームワークへのバージョン統一

図4に示すようにフレームワークは開発当初から進化を遂げ、現在のJ-Smile™で利用されているバージョンが最新版である。J-Smile™は常にフレームワークの最新版を取り込んでいるが、資材システム、経営管理I期、経営管理II期の各システムは当時のフレームワークのバージョンを凍結し利用している。

真の意味でのフレームワーク運用管理体制の一元化

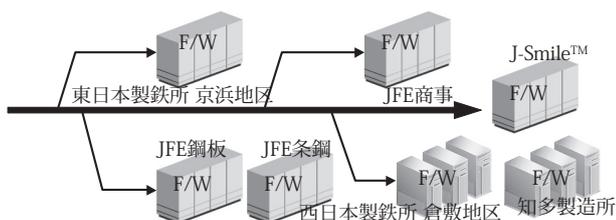


図3 フレームワークの横展開
Fig. 3 Spread of common framework

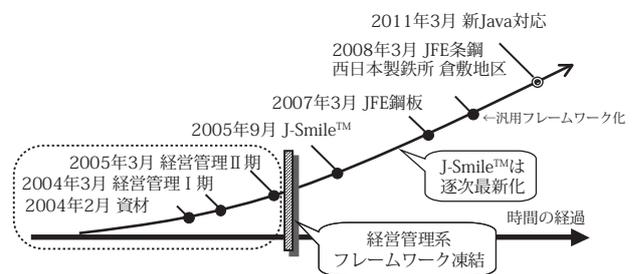


図4 フレームワークの進化
Fig. 4 Evolution of common framework

のために、これら経営管理系の古いフレームワークをJ-Smile™フレームワークのバージョンに追いつかせる予定である。

(2) Java のバージョンアップ対応

もう一つの大きな課題はJavaのバージョンに起因するものである。J-Smile™はJava1.4のバージョンで作成されており、アプリケーションサーバーはWAS5.1のバージョンである。このWAS5.1はIBM Corp.によるサポート切れを迎えるタイミングにある。また、遠隔地にバックアップシステムを構成するBCP (Business Continuity Plan: 事業継続計画)を進めているが、Javaのバージョンアップを実施しなければ、最新のハードウェア、ソフトウェア上で稼動しない状況にある。

現在、WAS7.0へのバージョンアップを計画している。WAS7.0のJavaはJava6である。プログラム言語であるJavaは進化を続けており、多くの新機能を持つが、下位互換が保証されておらず、多くの非互換を持つ。JavaアプリケーションのみならずJ-Smile™のフレームワークもJavaプログラムで構成されていることからJavaのバージョンアップにともないフレームワークを含めてプログラムを改修する予定である。

5. 終わりに

現在、あるシステムを富士通(株)環境からIBM Corp.

環境への移植を行っているが、これは共通フレームワークにより、アプリケーションプログラム資産とプラットフォームが独立しているから可能となったものである。プログラムの修正を行うことなく、劣化更新やコストダウンのために、プラットフォームを自由に選択・変更できるようになった。

アプリケーションシステムは、ビジネスの変化に柔軟に対応し、継続的に成長させていく必要がある。共通フレームワークにより、このアプリケーション戦略とプラットフォーム戦略を独立して推進できるようになった意義は大きいと考える。

参考文献

- 1) 原田敬太, 森本哲也, 大川浩志. J-Smile を支える IT イノベーション (テクノロジー) —大規模基幹システムへの Web 技術適用—. JFE 技報. 2006-11, no. 14, p. 21.



原田 敬太



加島 靖蔵