

オープン環境のシステム構築を高品質・短納期で実現するWebシステム開発標準 “MIWESTA”

川口正高* 塩尻綾子*
浅見可津志* 原田雅史**
佐藤啓紀*

要 旨

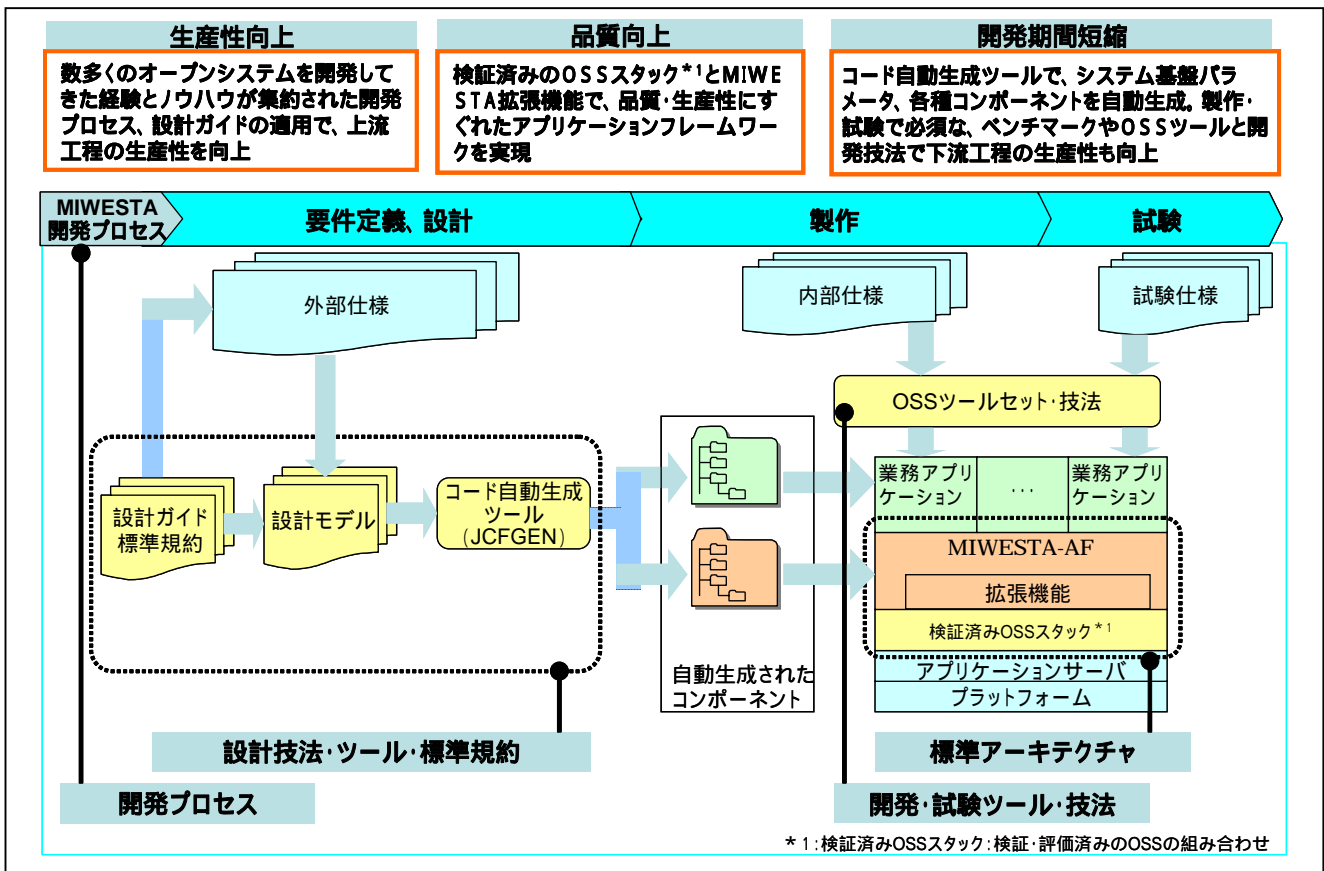
企業のTCO削減を目的としたオープンソースソフトウェア(OSS)を活用したミッションクリティカルな企業情報システム開発の事例が増えるにつれ、システムを高品質、低コスト、短納期で開発することを可能とする開発標準の整備が必須となっている。三菱電機インフォメーションシステムズ(株)(MDIS)では、OSSの黎明期よりMDIS Webシステム開発標準(MIWESTA^(注1):MDIS Web Development Standard)の整備を進め、顧客システムの品質・生産性向上を図ってきた。

オープンシステム対応の開発標準において、システムに最適なOSSを選定する技術力は非常に重要であるが、単にOSSを適用するだけでは最新技術を導入する際のオーバーヘッドによって、期待する生産性を得ることはできず、この

オーバーヘッドを削減する開発プロセスや設計技法を整備することが重要である。

MIWESTAはこれらの課題を解決する開発標準で、既存の様々なOSSアプリケーションフレームワークの長所を組合せて実現したMIWESTAアプリケーションフレームワーク“MIWESTA-AF”をはじめ、このフレームワークを活用するための開発プロセスや設計技法及びコード自動生成ツール“JCFGEN”などをオールインワンで提供している。これまでに、様々な規模の実システムに適用され、その効果も実証されつつある。

(注1)MIWESTAは、三菱電機インフォメーションシステムズ(株)が商標登録出願中である。



MIWESTA技術体系図

MIWESTAは、検証済みOSSスタックとWebシステム構築に必要なMIWESTA拡張機能から成るアプリケーションフレームワークMIWESTA-AFをはじめ、MIWESTA開発プロセス、設計ガイド、開発ツール・技法、コード自動生成ツールなどオープンシステムの開発プロジェクトに必要な技術を体系化したWebシステムの開発標準である。

*三菱電機インフォメーションシステムズ(株) **三菱電機(株)

1. まえがき

企業情報システムをオープンシステムとして構築する事例が増えるにつれ、システムインテグレータ業界ではアーキテクチャ設計に起因するシステム品質・生産性の課題が頻繁に取り上げられるようになった。オープンシステムでは、OSS活用とマルチベンダー製品による開発が中心となるため、従来のシングルベンダー環境の開発に比べてソフトウェア・アーキテクチャ（以下、アーキテクチャと略称）設計がシステム品質に大きく影響していると考えられる。MDISでは、上記の課題を解決するため、Webシステム開発標準“MIWESTA”を開発した。

本稿ではMIWESTAの提供する主な開発技術と適用事例について述べる。

2. オープンシステムの品質・生産性の課題

OSSを活用した企業情報システムの品質・生産性を確保するためには、オープンシステムに有効な開発標準の整備が必須であるが、次の課題がある。

- (1) 多様な業務ニーズに適応可能な品質・生産性の高いアーキテクチャを選定することが必要である。その実現手段として複数のOSSを活用することが一般的であるが、OSS選定では、OSSを組み合わせたアーキテクチャの機能・性能・品質を事前に評価することが重要である。
- (2) オープンシステム開発では、OSSの特質を把握したうえで、性能や拡張性・使用性などの品質特性を引き出すためのアーキテクチャ設計がシステム全体の品質に大きく影響する。そのため、基本的なソフトウェア構造を規定するアーキテクチャを決定し、それに基づいて開発を進めるアーキテクチャ駆動の開発プロセス⁽¹⁾が重要である。
- (3) Javaを基本としたオープンプラットフォームでは、フレームワークを構成するOSSの技術習得に多くの時間がかかるため、そのオーバーヘッドを軽減できる定型化された開発方式への転換が重要である。

3. MIWESTAの全体構成

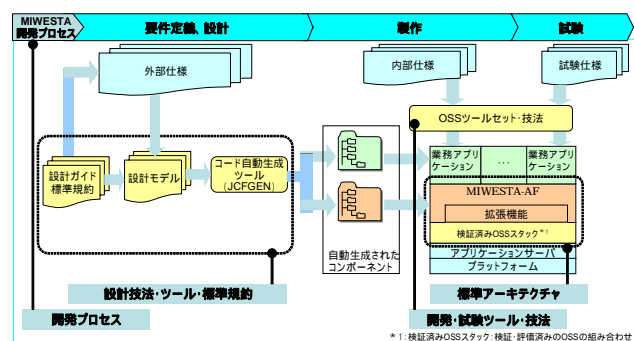


図1. MIWESTAの全体構成

MIWESTAは、検証済みOSSスタックとWebシステム構築に必要なMIWESTA拡張機能からなる3階層アーキテクチャのアプリケーションフレームワーク“MIWESTA-AF”を中心とし、このフレームワークを活用するための開発プロセスや設計技法及びコード自動生成ツール“JCFGEN”など、オープンシステムの開発プロジェクトに必要な技術を体系化したWebシステムの開発標準である（図1）。

4. MIWESTAのアーキテクチャ

4.1 基本的な考え方

MIWESTAは、図2に示すとおりプレゼンテーション層、ビジネスロジック層、データアクセス層からなる論理3階層モデルを標準のアーキテクチャとして採用しており、各層間は疎結合なアーキテクチャとしている。また、OSSで不足しているシステム構築に必要な共通機能は、“拡張機能”として、MIWESTA-AFが提供している。

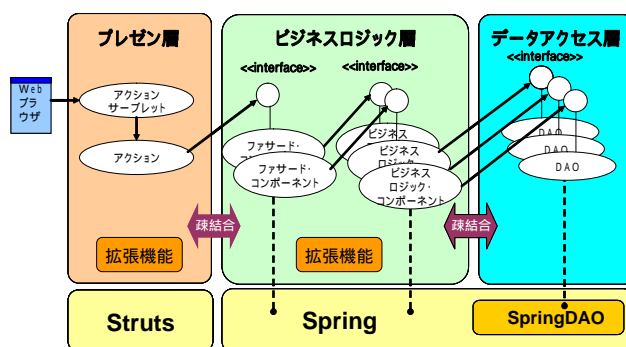


図2. MIWESTAの3階層構造

4.2 MIWESTAのアーキテクチャ概要

MIWESTA-AFを構成する各層は、次のようなアーキテクチャである。

- (1) プレゼンテーション層では、Java EE^(注2)(Java2 Enterprise Edition)が提供するサーブレット、JSP^(注2)(Java Server Pages)の基本技術をベースにブラウザからの処理要求をサーバ側のアプリケーションプログラムに伝えるためのフレームワークとして“Struts”を採用している。
- (2) ビジネスロジック層では、POJO(Plain Old Java Object)でビジネスロジックが作れるようにDI (Dependency Injection) コンテナが使われる。DIコンテナは、複数のオブジェクトを組合せて利用する場合に、オブジェクト管理を簡単にする仕組みである。DIコンテナには、“Spring”を採用している。
- (3) データアクセス層は、Springが提供するDAO (Data Access Object) デザインパターンを利用して実装する。ここでは、従来の“JDBC”^(注2)(Java Database

(注2) JavaEE, JSP, JDBCはSun Microsystems社の登録商標である。

Connectivity)のフレームワークの他に、オブジェクトとデータベースのテーブルとの間で形式の変換を行うO/R(Object Relational)マッピング技術も利用することができる。

4.3 MIWESTAアーキテクチャの特長

- (1)各層間の結合が“疎結合”であり、置換性、独立性がある。例えばプレゼンテーション層の実装技術の置換が容易であり、後述する事例ではWebサービス^(注3)による実装をプレゼンテーション層に追加することで、“.net”^(注3)のリッチクライアントをブラウザベースのクライアントとプレゼンテーション層に共存させるアクセス方式を実現している。
- (2)業務機能以外の要件に基づく共通ポリシー的な非機能要求機能は、MIWESTA - AFの拡張機能として持たせることで品質・生産性の高いアーキテクチャを目指している。例えば、トランザクション、例外処理、ロギング、ユーザ認証済みチェック、実行認証チェック、データベースコネクションなど性能やセキュリティを確保するための拡張機能がある。
- (3)MIWESTAでは、“DBT-1”(Open Source Development Labs Database Test1)をMIWESTA-AF上に実現したベンチマーク⁽²⁾を利用して、一連のOSS技術を組合せた機能・性能を評価することで可能であり、目標レベルの品質・生産性を確保できるかどうか事前に評価することができる。

5 . MIWESTAの開発プロセス

5.1 MIWESTA開発プロセスの概要

MIWESTA開発プロセスは、業界標準の統一プロセスUP(Unified Process)に基づくオブジェクト指向の手法をMDIS標準のウォーターフォール開発プロセスに取り入れたアーキテクチャ駆動の開発プロセスである。

このプロセスでは、MIWESTA-AFが規定するソフトウェア構造(図3)を構成する要素を設計するための具体的な設

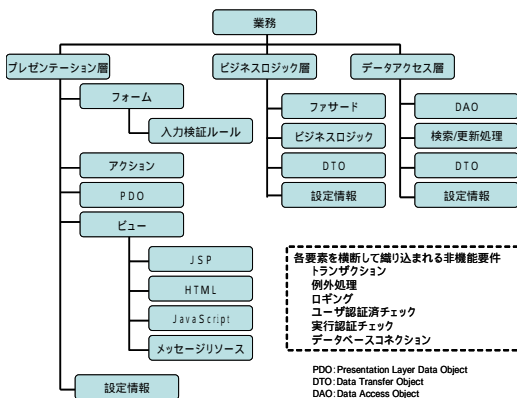


図3 . MIWESTA-AFが規定するソフトウェア階層構造

(注3)Microsoft .netは、Microsoft社の登録商標である。

計手順、標準ドキュメントを定義している。従って、実際のプロジェクトではこのMIWESTA開発プロセスをプロジェクトの特性に合わせて修正することにより、短時間で作業定義の漏れがないプロジェクト固有のWBS (Work Breakdown Structure)を定義することが可能となる。更に、MIWESTA開発プロセスでは、コード自動生成ツール“JCFGEN”の使用を前提とした開発プロセスも提供している。

5.2 設計モデルと自動生成ツールによる開発手法

MIWESTAは、図1に示すとおり、設計モデルとJCFGEN及び設計モデルを作成するための設計ガイドを提供している。設計モデルは、MIWESTA-AFのソフトウェア構造を構成する要素を形式化したもので、13種類の設計モデルがあり、外部仕様の設計情報を各設計モデルへ記述する方法は設計ガイドで示している。JCFGENは、この設計モデルの情報を入力情報として、MIWESTA-AFが規定するソフトウェア構造のソースコードを自動生成する。

5.3 自動生成のメカニズム

MIWESTA-AFに沿ったJavaのソースコードの自動生成効率を最大限にするために、過去のソフトウェア設計のドキュメントと、作成されたWebアプリケーションのソースコードの関係及び設計情報間の関係を調査・分析し、13種類の設計モデルに類型化した(図4)。関連する各設計モデルは、システム内でユニークな完全修飾されたクラス名をキーとして、画面遷移情報用、クラス情報用、メッセージ情報用など、合計13種類の表形式のテンプレートとして作成されている。

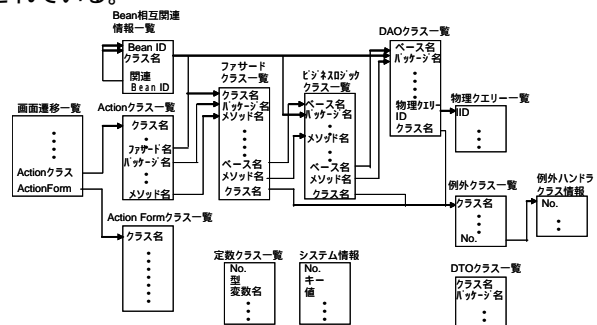


図4. MIWESTA設計モデル 関係図

JCFGENは、この設計モデルを元に、MIWESTA-AFに準拠するソフトウェア構造のファイル体系に沿って、各種コンポーネントのクラスを生成する。クラスには、次の項目が生成される。パッケージ文、インスタンス変数コメント、クラス変数、インポート文、インスタンス変数、メソッド・コメント、クラス・コメント、クラス変数コメント、メソッドなどである。単一の設計モデルからだけでは生成できない項目については、クラス名をキーとして、関連する設計モデルテンプレートを検索し、それらの情報を組み合わせることで生成する。また、メソッド内のコードを出力す

際には、予め設計して登録してあるコード（入力パラメータやリターン値のチェック、トレース文、例外処理など）を利用することで、最適なコードを生成することが可能になる。

5.4 JCFGENの特長

JCFGENの主な特長は、以下のとおりである。

- (1) クラスやメソッドの枠組みを自動生成するので、保守性や移植性などの向上につながる。
- (2) コメント文、ログ出力機能、トレース機能、パラメータ・チェック機能などの定型的な処理コードを自動生成することで、コーディング段階の生産性が向上する
- (3) MIWESTA-AFに準拠の業務ロジックのひな型だけでなく、それに関連する設定ファイルやプロパティファイルも生成され、生産性が向上する。
- (4) 開発者がこれまで蓄積してきた実装ノウハウなどを反映したベストプラクティスに基づいてソースコードを生成しており、性能や拡張性を含む幅広い品質特性の向上につながる。

6. MIWESTAの適用事例と効果

MIWESTAは、業種を問わず情報系システムから基幹系システムまで幅広く適用されている⁽⁴⁾。ここでは、MIWESTAを適用した事例システムとその適用効果について述べる。

6.1 事例システム概要

本事例システムは、製造業向け商品発注システムでオフコンを中心とした現行システムを、インターネットを活用したWebシステムに再構築する事例である(図5)。

システム化の課題としては、多様な販売チャネルへ対応するため、Webクライアントとリッチクライアントに対応したシステム構築が必要であった。

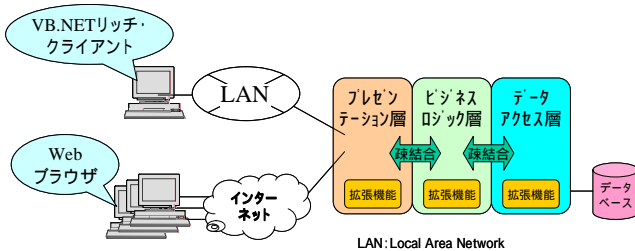


図5. 製造業向け商品発注システム

6.2 MIWESTA利用の効果

生産性向上、品質向上、開発期間短縮の観点から、上述の事例システムにおける効果について述べる。

(1) 生産性向上

本事例では全体開発量（コメント含む）の56%をJCFGENで自動生成でき、生産性向上に寄与していることが確認できた(表1)。今後は更に、データを蓄積し、アプリケーションの特性との関係も分析することが課題である。

また、疎結合な論理3階層アーキテクチャにより、本事例のような多様なプレゼンテーション層への対応が容易で

表1. 自動生成率

内訳	単位[KL]		単位[%]		
	全体開発量	自動生成量	追加・修正量	生成率	
本事例	全体行数	15.2	8.5	6.7	56

注) 画面系：JSP、HTMLを含まず

あり、リッチクライアントをブラウザベースのクライアントとプレゼンテーション層に共存させるアクセス方式をとっても、ビジネスロジック層、データアクセス層のクラスの共通化ができた。

(2) 品質向上

OSSを組合せて、性能も検証済みのOSSスタックを利用することで、システムの性能も確保できた。

(3) 開発期間短縮

開発期間短縮については、次のような観点から、効果を確認することができた。

- ・開発プロセス、MIWESTA-AFなどの適用によりアーキテクチャ設計の期間を短縮することができた。
- ・MIWESTA-AFの拡張機能の提供により、プロジェクトで機能を開発することなく活用でき生産性に効果があった。
- ・コード自動生成により視認性が高く、エラー処理も生成されるので抜けがなく、品質・生産性に効果があった。

6. むすび

MIWESTAの利用状況は、06年度実績として当社のオープンシステム開発プロジェクトの約4割に達している。今後は、MIWESTA適用を通じての課題をフィードバックすると共に、リッチクライアントやSOA(Service Oriented Architecture)など新しいアーキテクチャパターンの拡充などの“システム適応力の拡大”と、アーキテクチャ設計技術の拡充や自動化の強化などの“ソフトウェア生産性向上”を2軸の柱として強化拡充を図っていく。

参考文献

- (1) 松岡 恭正, 他: お客様との共創を目指したエンタープライズアーキテクチャ (EA), 三菱電機技報, 79, No. 4, 241~246 (2005)
- (2) 金木 佑介, ほか: JavaベースのWebシステムを想定したPostgreSQLベンチマークスイート, 情報処理学会, 第68回全国大会, 第3分冊 (2006)
- (3) 土屋 隆, ほか: Webサービスを利用したJava/.NET連携フレームワーク, 情報処理学会, 第69回全国大会, 3E-1(2007)
- (4) 福島 修, ほか: テレビ会議を活用した新金融リテールソリューション 三菱東京UFJ銀行向け “MUFJテレビバンクシステム”, 三菱電機技報, 80, No. 4, 241~244 (2006)