

アプリケーション構築サービスを支える Webアプリケーション自動生成技術

天沼敏幸*
浅見可津志*
大野次彦*

Web Application Generator for Application Building Service

Toshiyuki Amanuma, Katsushi Asami, Tsugihiko Ohno

要旨

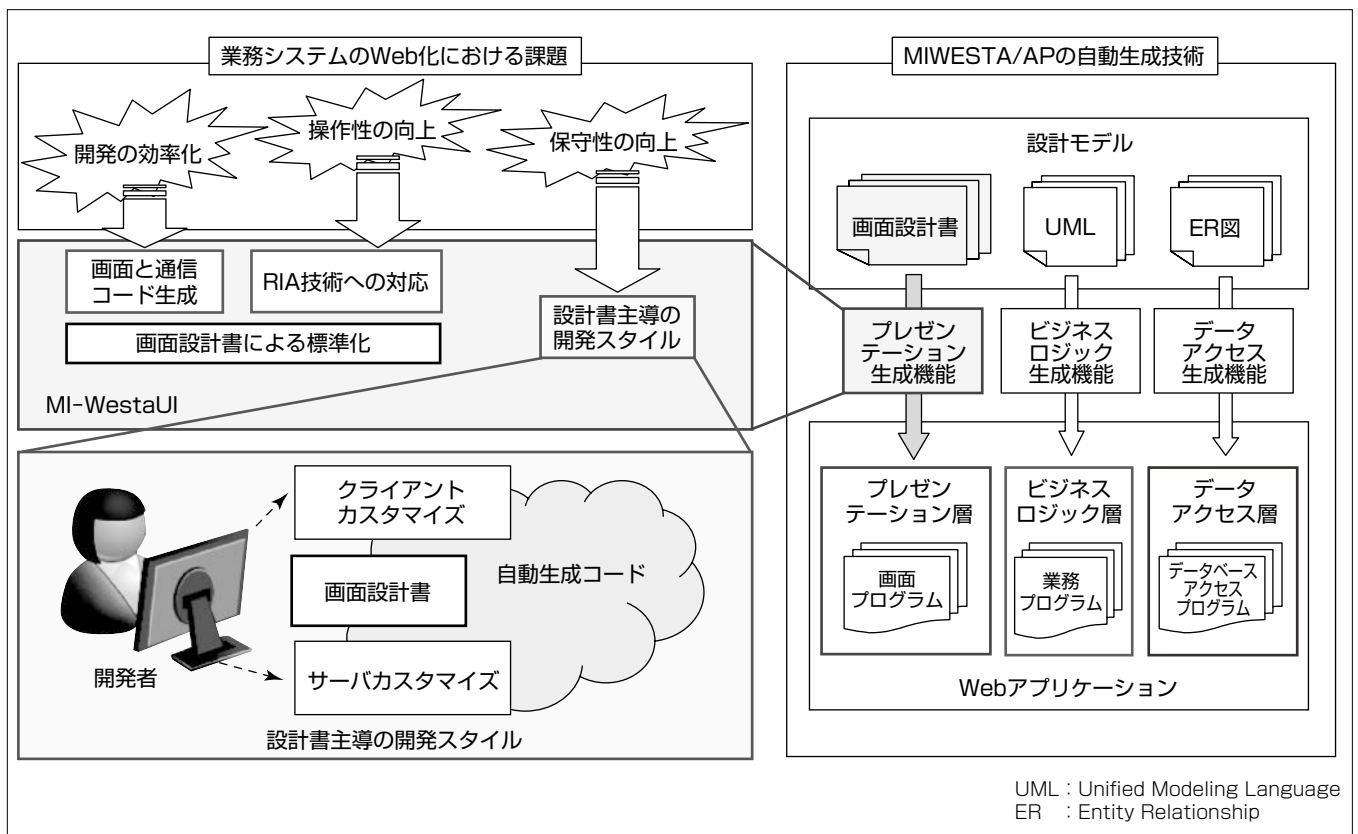
クラウドコンピューティング技術の台頭によって、業務システムをプライベートクラウドなどのWeb環境に対応させる動きが活発化している。三菱電機インフォメーションシステムズ(株)(MDIS)では、早くからWebアプリケーション開発標準を体系化し、MIWESTA/AP(MDIS Web Development Standard for Application)としてWebアプリケーション構築技術の蓄積、整備を図ってきた。近年では、MDISがITサービスインテグレーション“BizFlex”を展開するに伴い、更なる生産性向上を目指し、自動生成技術を強化している。

業務システムのWeb化における一般課題として開発の効率化、操作性の向上、保守性の向上が挙げられる。MIWESTA/APでは論理3階層アーキテクチャの各階層

に対応した自動生成機能があり、特にユーザーインタフェースについては、プレゼンテーション層の自動生成機能(MI-WestaUI)でその課題に対応し、画面設計書による標準化、画面と通信コード生成、RIA(Rich Internet Applications)技術への対応、設計書主導の開発スタイルの諸機能を実現している。実プロジェクトへの適用結果ではMI-WestaUIを含むMIWESTA/APの自動生成技術によって、業務システム全体の開発量の約6割のコード自動生成を実現している。

今後は機能強化を図るとともに、ユーザーインタフェース実装技術の分野で利用が進みはじめたJavaEE(Java^(注1) Enterprise Edition)のJSF(Java Server Faces)について、RIA生成機能への対応を予定している。

(注1) Javaは、Oracle corp.の登録商標である。



業務システムのWeb化における課題とMIWESTA/APの自動生成技術の対応

業務システムのWeb化における課題として開発の効率化、操作性の向上、保守性の向上がある。MIWESTA/APは論理3階層アーキテクチャに対応しており、特にユーザーインタフェースについてはプレゼンテーション層の自動生成機能(画面設計書による標準化、画面と通信コードの生成、RIA技術への対応、設計書主導の開発スタイル)によって、業務システムのWeb化における課題に対応している。

1. ま え が き

MDISではWebアプリケーションの開発標準を体系化し、MIWESTA/APとして、Webアプリケーション構築技術の蓄積・整備を図ってきた⁽¹⁾。近年ではITサービスインテグレーションBizFlexを展開するに伴い、更なる生産性の向上を目指し、自動生成技術を強化している。

MIWESTA/APが対象とする業務システムは数百画面をもつユーザーインタフェース中心のシステムであることが多い。ユーザーインタフェースについては、操作性への要求が高く、ユーザーからの改善要望が集中しやすい。MIWESTA/APの自動生成技術は、そのような業務システムをWebアプリケーションで効率的に開発するための機能提供を目標としている。

本稿では、MIWESTA/APの自動生成技術の全体像を紹介し、その中でプレゼンテーション層の自動生成機能(MI-WestaUI)⁽²⁾を中心に、業務システムのWeb化における課題とそれを解決するMI-WestaUIの実現方式及び実システム適用による改善点について述べる。

2. MIWESTA/APの自動生成技術

MIWESTA/APはWebアプリケーションをプレゼンテーション層、ビジネスロジック層、データアクセス層の論理3階層で構成するアーキテクチャに対応している。

MIWESTA/APの自動生成方式の特徴は、上流側からの設計成果である設計モデルから、Webアプリケーションの論理3階層の各層に対応したプログラムを生成する点にある(図1)。

プレゼンテーション層では、画面設計書からブラウザに表示される画面及び通信に使用するデータ構造とサーバ側

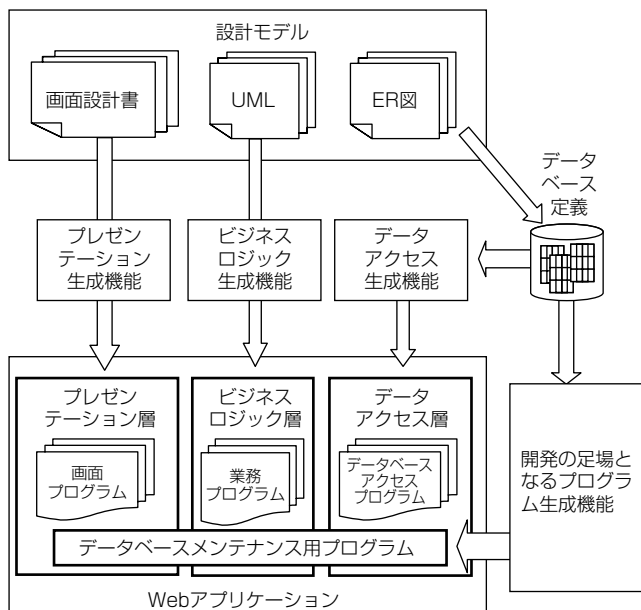


図1. MIWESTA/APの自動生成技術

で送受信するコードを生成する。ビジネスロジック層では、UMLで記述された設計書からJavaコードを生成する。データアクセス層では、ER図から作成したデータベース定義に基づきデータベースアクセス用Javaコードを生成する。これらの機能を利用し、データベース定義からWebアプリケーション開発の足場となるデータベースメンテナンス用プログラムを一括生成する機能も提供している。

3. 業務システムのWeb化における課題

3.1 開発の効率化

大規模なWebアプリケーションを開発する際に問題となるのは、開発環境での実装の自由度が高く、品質や開発スピードが開発者個人のスキルに大きく依存する点である。実際の開発では、開発規約によって標準化を図るようにしているが、徹底は困難で開発の効率化を妨げる要因となる。対策としては、画面の設計モデルを作成する時点で標準化を実施し、それをそのまま実装コードに反映する方式が考えられる。

3.2 操作性の向上

一般に業務システムを標準的なHTML(HyperText Markup Language)の機能だけで実装した場合、操作性が低下することが多い。画面のデザインや操作性への要求は高く、ユーザーからの改善要望が集中しやすいところである。そのため、操作性の向上を目的として各種のRIA(Rich Internet Applications)技術の活用は不可欠であるが、導入にあたっては、個別のRIA製品に精通したプログラミング技術者を確保しなければならないという課題がある。

MI-WestaUIの特長の一つは、実装に依存しない画面設計モデルから、特定のRIA技術に対応した実装コードを生成する点にあり、開発者のスキルに依存しない。

3.3 保守性の向上

ユーザーインタフェースの開発では、試験工程での不具合修正や仕様変更によって、数百の画面の一括変更が必要になるケースが多い。これは、保守フェーズでの改良開発でも同様である。そのため、大量の画面を繰り返し変更できる方式や、開発者の負担を伴わない方式によって画面の設計モデルと実装コードを一致させる開発スタイルをサポートする必要がある。

4. MI-WestaUIの実現方式

この章では、業務システムのWeb化における課題に対応するためにMI-WestaUIが提供している諸機能(図2)について、その実現方式を交えて述べる。

4.1 画面設計書による標準化

通常、画面の開発では次に示す①から④の設計項目について、設計時と実装時の2段階で、設計レベルの詳細化を行う。この過程では担当者間の解釈の齟齬(そご)などによ

って誤りが混入することが多い。

MI-WestaUIでは、標準化された画面設計書によって、設計の時点で、実装まで含めた開発手順の標準化を実施している。設計者に対しては、アプリケーションフレームワークの機能を前提にすることで、実装に関する知識がなくても、詳細レベルの設計ができるようにしている。

MI-WestaUIの画面設計書では、Excel^(注2)の形式で、次に示す画面設計項目を指定する。

- ①画面レイアウト-画面上の部品配置及び表示ラベル
- ②入出力項目-部品に対応するデータ名、初期値
- ③入力制約-入力データのエラー検出のための規則
- ④アクション明細-サーバ通信時の呼出し方法

MI-WestaUIは画面設計項目が入力されたExcelシートを基に、実装コードを自動生成する。

生成されたコードは、MI-WestaUIが提供するEclipse^(注3)上の開発環境で自動的にコンパイルされる仕組みになっており、そのままアプリケーションサーバを利用した試験作業に移行できる。

また、画面設計書の設計情報は、Eclipse上の“UIモデルエディター”(図3)でグラフィカルに表示することができ、プログラマが設計情報を確認しながら実装作業を実施できる。

(注2) Excelは、Microsoft Corp. の登録商標である。

(注3) Eclipseは、Eclipse Foundation, Inc. の登録商標である。

4.2 画面と通信コード生成

MI-WestaUIでは画面設計書から、ブラウザの画面コード、サーバとの送受信メッセージ及び通信コードを自動生

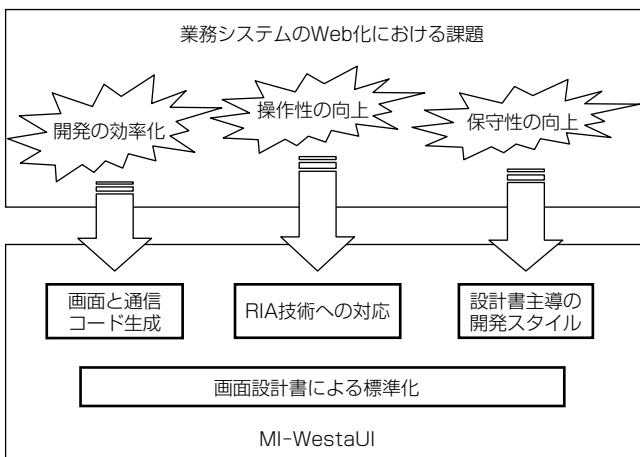


図2. 業務システムのWeb化における課題と対策

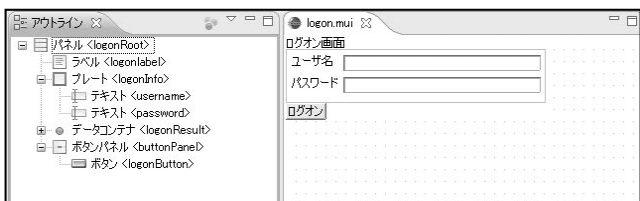


図3. UIモデルエディター

成することによって、プレゼンテーション層の開発を効率化している。また、MI-WestaUIによって生成される通信機能は、図4に示すようにWebサービス(HTTP(Hyper Text Transfer Protocol) SOAP(Simple Object Access Protocol))として動作するため、通常のWebアプリケーションのように通信に伴う画面全体の再表示が発生しない。

4.3 RIA技術への対応

MI-WestaUIの自動生成機能は、様々なRIA実装技術に対応するため交換可能な方式になっており、現在は、RIA技術で最も普及しているFlex^(注4)に対応している。Flex部品群の実装コードを自動生成することによって、Flexに精通していない開発者でも高度な操作性を持つ画面を容易に構築することができる。

操作性の向上の一例として、入力制約違反時のメッセージ表示で該当項目が指摘されるユーザーインターフェースを示す。入力制約は図5に示すように、必須チェック、テキスト属性(半角、全角ほか)、最大・最小桁等主な項目を画面設計書に指定することによって記述する。入力制約チェックの実装コードはMI-WestaUIによって生成されるが、制約違反時のメッセージは、部品のラベル設定から自動で合成されて表示される。この例ではメッセージの“ユーザー名”の部分が部品のラベル設定に当たる。

(注4) Flexは、Adobe Systems, Inc. の登録商標である。

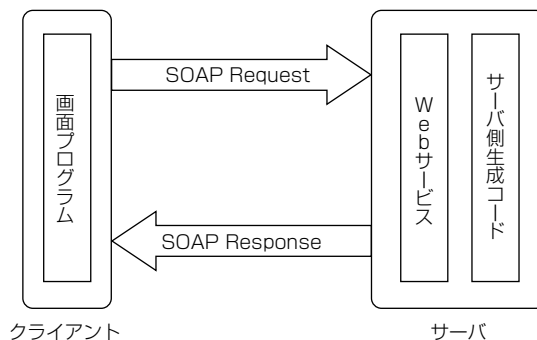


図4. MI-WestaUIによって生成される通信機能



		入力制約							
タグ名	必須	テキスト属性	ゼロ不可	最小桁	最大桁	最小値	最大値	ステップ数	正規表現
name	必須								
pass	必須								

図5. 必須チェックの表示と設定例

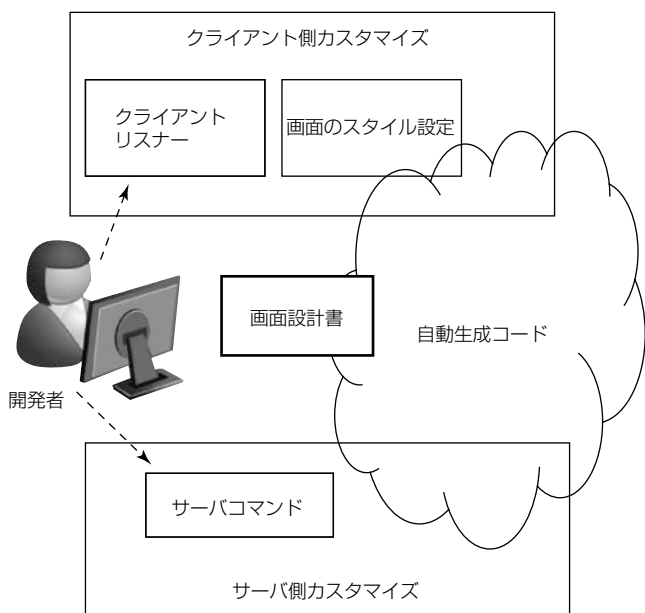


図 6. MI-WestaUIの開発スタイル

4.4 設計書主導の開発スタイル

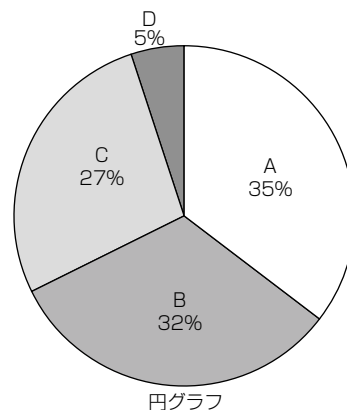
MI-WestaUIでは、頻繁な画面の仕様変更に対応するため、修正が発生した場合に、画面設計書を修正し、画面を自動生成する手順を繰り返す開発スタイルを想定している(図6)。そのためには開発者が画面のカスタマイズの目的で独自に記述したコードを、自動生成の再実行で書きさしなことが前提となる。したがって、開発者がカスタマイズのために記述する以下のコードを自動生成部分から分離する方式にしている。

- ①クライアントリスナー(クライアント側における標準外の操作性を向上する画面ごとのコード)
- ②サーバコマンド(サーバ側のメッセージ送受信, ビジネスロジックの呼出しの画面毎コード)
- ③画面のスタイル設定

このように、標準化された部分は自動生成によって開発効率を下げずに変更を繰り返すことができ、一方、カスタマイズ部分は局所化され、変更が把握しやすいという利点がある。また、結果として画面設計書とコードは一致した状態が保たれる。

5. 実システム適用における改善

MI-WestaUIを含むMIWESTA/APのフレームワークを利用したシステム開発では、全体開発量の約6割のコードを自動生成することができた。また、プロジェクト適用の結果に基づき、機能追加、性能向上、操作性の改善を行った。実システム適用における機能追加では、Flexで利用可能なカスタム部品を利用したいという要望があった。



部品の種別	画面レイアウト		入出力項目		拡張部品	
	表示用ラベル	深さ	タグ名	部品名	属性名	属性値
パネル		2	plate			
ラベル	円グラフ	3				
拡張部品		3	medalsChart	mx:PieChart	dataProvider	{dataInfo.plate.medalsChart.data}

図 7. 拡張部品及び拡張属性の記述例

MI-WestaUIの画面設計書では、業務システムで利用頻度の高い約30種類の標準部品を選択できるが、カスタム部品の情報を記述することはできなかった。そこで、プロジェクト適用時の柔軟性に配慮し、拡張部品及び拡張属性を記述できるようにした。これによってサードパーティから提供されるカスタム部品の利用も可能となっている。Flexのチャートコンポーネントの一つである円グラフを利用する例を図7に示す。設計仕様書の部品名に拡張部品であるmx:PieChartを直接記述することによって、標準部品と組み合わせた画面を構成している。

6. むすび

Webアプリケーションの実装技術の進展にはめまぐるしいものがあり、MIWESTA/APでは、新技術を体系化するために継続的な取組みを行っている。ここで述べたWebアプリケーションのユーザーインターフェース実装技術の分野では、JavaEEのJSFが利用されるようになってきており、今後のMI-WestaUIの実装技術対応として、JSF実装のRIA生成機能開発を予定している。

参考文献

- (1) 川口正高, ほか: 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”, 三菱電機技報, 81, No. 7, 489~492 (2007)
- (2) 渡邊圭輔, ほか: Webアプリケーションユーザーインターフェース構築技術, 三菱電機技報, 82, No. 12, 783~786 (2008)