特集論文

汎用双方向型Web画面自動生成技術

河村美嗣*　小笠原淳子***
田村孝之**
宮嵜弘治***

General-purpose Bidirectional Web Screen Automatic Generation Technology

Yoshitsugu Kawamura, Takayuki Tamura, Kouji Miyazaki, Atsuko Ogasawara

要　旨

近年，情報システムの形態はWebブラウザをクライアントとしてサーバなどを利用するWebコンピューティングが主流となっている。また，システム開発では，社会の急速な変化に対応することが求められており，短期間／低コスト／高品質にシステムを構築するために，より一層の開発生産性と品質の向上が求められている。

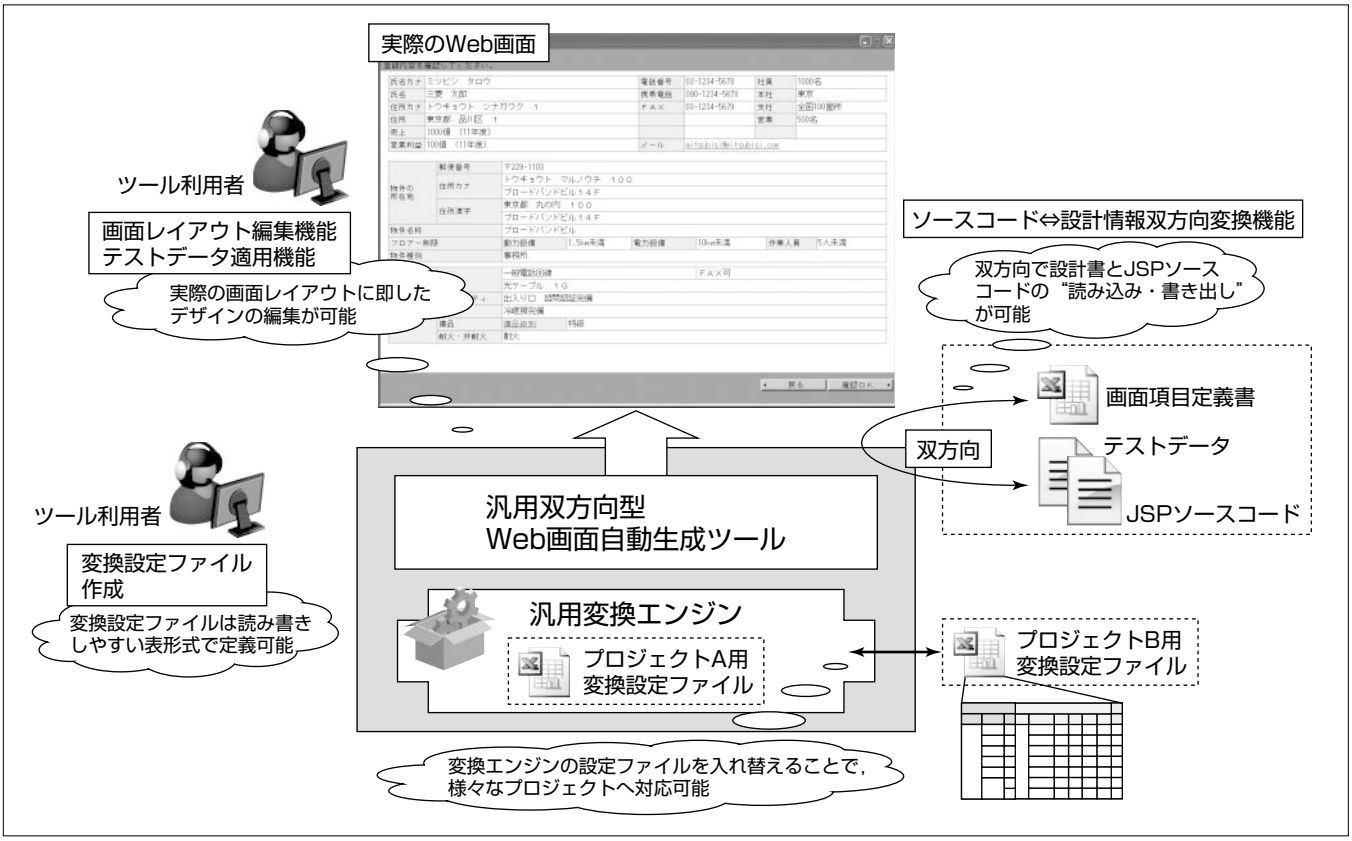
このような背景から，三菱電機インフォメーションシステムズ㈱(MDIS)及び三菱電機㈱では，Web画面開発の効率化を目的に，双方向型Web画面自動生成技術とこの技術を実装した双方向型Web画面自動生成ツールの開発を行ってきた[(1)(2)(3)(4)(5)]。このツールを使用することで，ユーザーの要求を取り込んだWeb画面のレイアウトを容易に作成でき，また，作成した画面レイアウトから設計書やJSP(JavaServer[注1] Pages)ソースコードを自動生成することで，システム開発における生産性と品質の向上を実現している。

しかし，これまでの双方向型Web画面自動生成技術は，プロジェクト固有の仕様書やソースコードのフォーマットに依存する部分が多く，他プロジェクトには容易に適用できないという課題があった。

その課題に対して，今回，このツールを容易に他プロジェクトへ適用可能にするため，双方向型Web画面自動生成技術のコア部分である変換エンジンを汎用化した“汎用双方向型Web画面自動生成ツール”を開発した。

この汎用化によって，他プロジェクトへ適用可能にする際の工数を汎用化前と比較し三分の一以下である5人日以内へ短縮可能という効果が得られている。

(注1)　JavaServerは，Oracle Corp. の登録商標である。



汎用双方向型Web画面自動生成技術

この技術は，画面レイアウトから設計書やJSPソースコードを自動生成して開発工数の削減を可能にする双方向型Web画面自動生成技術のコア部分である変換エンジンを汎用化することで，他プロジェクトへ容易に適用可能にする。この技術を実装した汎用双方向型Web画面自動生成ツールの利用者は，プロジェクト向けの変換エンジンを作成することなく，変換設定ファイルを作成することで，ツールを新プロジェクトへ適用可能にすることができる。

*三菱電機㈱ 情報技術総合研究所　**同研究所(博士(工学))　***三菱電機インフォメーションシステムズ㈱

1. まえがき

近年，情報システムの形態はWebブラウザをクライアントとしてサーバなどを利用するWebコンピューティングが主流となっている。また，社会の急速な変化に即座に対応したシステムの開発が求められ，短期間／低コスト／高品質にシステムを構築する必要性が高まり，顧客からはより一層の開発生産性と品質の向上が求められている。

このような背景から，三菱電機インフォメーションシステムズ㈱及び三菱電機㈱では，Web画面開発の効率化を目的に，双方向型Web画面自動生成技術とこの技術を実装した双方向型Web画面自動生成ツールの開発を行ってきた。このツールを使用することで，ユーザーの要求を取り込んだWeb画面のレイアウトを容易に作成でき，また，作成した画面レイアウトから設計書やJSPソースコードを自動生成することで，システム開発における生産性と品質の向上を実現している。

しかし，これまでの双方向型Web画面自動生成技術はプロジェクト固有の仕様書やソースコードのフォーマットに依存している部分が多く，他プロジェクトへ容易に適用できなかった。他プロジェクトへの適用に時間がかかると，要件定義の初期段階に，このツールの特長である画面レイアウトや画面遷移イメージの共有を容易に行う機能を利用できなくなる。この機能の利用によって短期間かつ高品質に要件を確定し，プロジェクト全体の工数削減効果を大きくするには，他プロジェクトへの適用期間を，プロジェクトへの参画決定から要件定義開始までに，短縮する必要があった。

そこで，双方向型Web画面自動生成技術のコア部分である変換エンジンを汎用化することで，このツールを容易に他プロジェクトへ適用可能にした。本稿では汎用化方式の詳細について述べる。

2. 双方向型Web画面自動生成技術とは

双方向型Web画面自動生成ツールは，ソースコード及び画面仕様書とツール内部の設計情報との双方向変換機能，画面レイアウト編集機能と，テストデータを利用したソースコード(JSP)とHTML(HyperText Markup Language)の双方向変換機能から構成されている。

2.1 ソースコード／設計情報の双方向変換機能

ソースコードであるJSPファイルと，設計情報であるExcel(注2)ファイルを双方向に変換する機能である。このツールはJSPファイルの読込機能，書出機能と，Excelファイルの読込機能，書出機能を全て備えている。ソースコードから設計情報に変換したい場合は，JSPファイルを読み込み，その後Excelファイルを書き出す。逆に変換したい場合は，まずExcelファイルを読み込み，その後JSPファイルを書き出す。

(注２) Excelは，Microsoft Corp. の登録商標である。

2.2 画面レイアウト編集機能

Web画面をグラフィカルな編集画面で作成／編集できる機能である。このツールには，Web画面の作成に必要な部品を配置するためのボタンが用意されており，マウス操作で部品を選択し画面レイアウト編集エリアへ配置することで，Web画面を容易に作成することができる。作成した画面レイアウトは，Webサーバを必要とせずにプレビューで確認することができる。また，プレビュー時に画面遷移を確認することもでき，実際のアプリケーションの動きに即した画面設計が行える。さらに，業務要件に合わせて画面項目の表示・非表示制御を行う場合，このツール上で様々なパターンのデータを適用した場合の画面項目の表示・非表示結果を確認することができる。

2.3 テストデータの適用

一般的なWebアプリケーションでは，画面上に動的にデータ(例えば顧客氏名や住所等)を表示する。そのため，Web画面デザインの要件を詰めていく際には，様々なテストデータが表示された状態のサンプル画面を作成し利用する。

このツールは，テストデータとJSPソースコードを合成／分離する機能を備えており，テストデータを表示したまま画面レイアウトの編集が可能である。また，テストデータを差し替えることで，様々なテストデータを適用した場合の画面デザインを簡単に確認することができる。この機能によって，要件定義段階で作成したWeb画面デザインからそのままJSPソースコードを生成することができ，効率的な開発が行える。

3. 他プロジェクトへ適用する際の課題と対応策

2.1節で述べた双方向変換機能を実現するための変換エンジンは，プロジェクトで用いる仕様書やソースコードのフォーマットに合致するよう，開発する必要がある。そのため，このツールを様々なプロジェクトへ適用するにあたっては，**図1**に示すような変換エンジン個別開発方式が考えられる。これは，プロジェクトごとの仕様書やソースコードのフォーマットに対応する変換エンジンを開発するという方式である。

しかし，変換エンジン個別開発方式では，変換エンジンをプロジェクトごとに新規に開発する必要があるため，開発工数がかかるという問題があり，そのため，このツールでは，変換エンジンを個別に開発するのではなく，変換エンジン自体を汎用化するというアプローチを採用することとした。

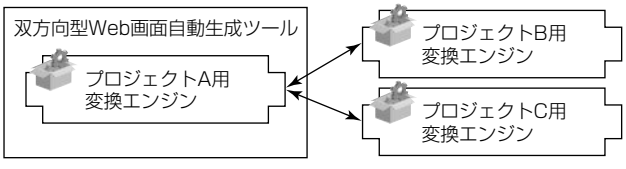


図１．変換エンジン個別開発方式

4. 汎用化実現方式

このツールで採用した汎用化実現方式を**図2**に示す。これは，汎用変換エンジンを開発して，プロジェクトごとに外部で定義したプロジェクト固有の変換設定ファイルを入れ替えるという方式である。

変換設定ファイルには，表形式の変換規則設定ファイル，スクリプト形式の変換スクリプトファイル，変換設定ファイルの自動単体テスト用のテストデータ定義ファイルの三種類のファイルが含まれている。次節より，それぞれの変換設定ファイルの詳細について述べる。

4.1 変換設定ファイル形式の選択

汎用変換エンジンは，変換設定ファイルを読み込み，ソースコード／設計情報の双方向変換を実現する。変換設定ファイルには，ソースコードと設計情報の変換関係の情報を定義する。変換関係を定義するためには，構造化されたデータが表現可能であり，プログラムにとって読み書きが容易であることが望ましい。一方，変換設定ファイルを記述し作成するのは，このツールの利用者であるため，人にとっても読み書きが容易であることが望ましい。

これらの条件を満たすファイル形式を選定するにあたって，**表1**に示すファイル形式について比較検討を行った。ファイル形式としてはCSV(Comma Separated Values)，XML(eXtensible Markup Language)，JSON(JavaScript(注3) Object Notation)，Excelの4形式を列挙し，それぞれ構造化データの表現力と，人，プログラムからの読み書きのしやすさを比較した。比較の結果，データ構造化とプログラムからの読み書きのしやすさでは，XML，JSON形式が優れていることが分かる。しかし，今回はこのツールの利用者にとって読み書きがしやすいことを重視し，Excel形式を選択した。

(注3) JavaScriptは，Oracle Corp. の登録商標である。

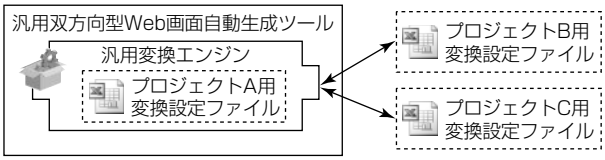


図2．汎用変換エンジン＋変換設定ファイル方式

表1．変換設定ファイル形式の比較

形式	書式	データの構造化	読み書きの容易性	
			人	プログラム
CSV	カンマ区切り	×	△	○
XML	マークアップ	○	△	○
JSON	JavaScript構文	○	△	○
Excel	Excelブック	△ セル結合で表現	◎ 操作性，罫線，色付けに優れる	△ Excelのインストールが必要 入出力処理がやや重い

4.2 1：1又は1：多対応関係の定義

ソースコード／設計情報の双方向変換で，単純な変換規則は，1：1又は1：多の対応関係で表現できる。

1：1又は1：多の対応関係を記述した表形式の変換規則設定ファイルの書式を**図3**に示す。変換規則設定ファイルは，左部分がソースコードに記載のJSPタグ名や属性名，中央部分が画面レイアウト編集に利用するHTMLタグ名や属性名，右部分が設計書の列番号の定義となっている。また，セルを結合することで，構造化されたデータを表現している。

このツールは，このファイル内に記述された対応関係を読み込み，双方向変換時にソースコードに記述されているJSPタグ名や属性名と一致する設定を左部分から探し，同一の行に定義されている対応関係を利用してHTMLや設計書へ変換する。また，逆変換も同様の処理を行うことで実現可能である。

4.3 その他の対応関係の定義

1：1又は1：多の対応関係では定義不可能な変換処理が必要な場合，表形式の変換規則設定ファイルに加えスクリプト形式の変換スクリプトファイルで定義する。

変換スクリプトファイルの書式を**図4**に示す。**図4**では，JSPタグを受け取りHTMLタグを返す処理を持つVisual C#(注4)のメソッドで，JSPからHTMLへの変換処理を定義している。

このツールは，変換スクリプトファイルを読み込み，変換時にコンパイルして実行することで，複雑な変換に対応する。また，逆変換については，逆変換用のスクリプトを読み込み，同様の処理を行うことで実現可能としている。

(注4) Visual C#は，Microsoft Corp. の登録商標である。

4.4 変換設定ファイルの単体テストの自動化

これまで述べてきた変換設定ファイルは，実際にプロジェクトへ適用する前に，想定通りの変換動作をするかテストする必要がある。このテスト工程は，ツールのサポー

図3．変換規則設定ファイル書式

```
public class JspElementConverter{
  public HtmlNode convertElement(HtmlNode node){
    // ここに変換ロジックを記述
  }
}
```

図4．変換スクリプトファイル書式

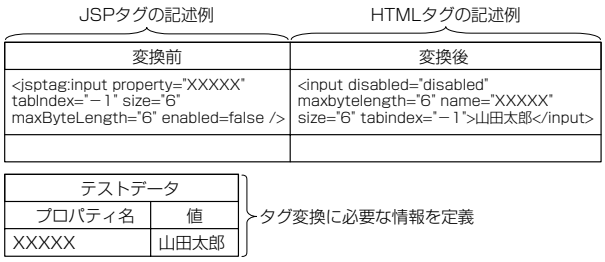


図5．テストデータ定義ファイル書式

トがない場合，無視できない工数がかかると考えられる。そのため，変換設定ファイルの単体テストをツールで自動化し，工数削減を図っている。

図5に自動単体テスト用のテストデータ定義ファイルの書式を示す。単体テストでは，JSPタグとHTMLタグの双方向変換が想定通りに動作するかどうかをテストする。そのためのデータとして，テストデータ定義ファイル内に変換前JSPタグの記述例と，変換後HTMLタグの記述例を定義する。また，JSPに外部から与えるパラメータを，テストデータとして，プロパティ名と値のペアで定義する。

テストツールは，変換規則設定ファイルと変換スクリプトファイルとテストデータ定義ファイルに記述された情報を利用して，実際に双方向変換を実施し，テストデータ定義ファイルに記述された通りの変換が実施されたかどうかの結果を画面に表示する。この結果を確認することで，変換規則設定ファイルと変換スクリプトファイルに記述された内容が正しいかどうかを判断できる。

5. 評　価

このツールをプロジェクトの初期段階である要件定義段階から使用するためには，プロジェクトへの参画決定から要件定義開始までのおよそ1週間で，このツールの準備を完了する必要がある。そのため，プロジェクトへ適用する際の工数の目標値として5人日以内を設定した。

汎用変換エンジンの効果を測定するため，双方向型Web画面自動生成ツールをあるプロジェクトへ適用する際の工数を，汎用化前後で比較した結果を**表2**に示す。

汎用化前は，プロジェクト向けの変換エンジンとして約2KLのプログラムを作成する必要があり，適用完了までにかかった合計工数は約14人日であった。

汎用化後は，変換エンジンの開発は不要であり，プロジェクト向けの変換設定ファイルだけを定義すればよい。変換設定ファイルの書式は用意されているため，作成時に外部設計／内部設計の工程は不要となる。また，単体テストの自動化によって，テスト工数の削減も見込める。

評価に利用したプロジェクトでは，変換設定ファイルの記述量は約20行×15ファイルであり，適用完了までにかかった合計工数は約4.5人日であった。

この結果から，汎用化によって工数を三分の一以下である4.5人日へ短縮することができ，目標値を達成できた。これによって，このツールをプロジェクトの初期段階から使用できることが見込める。

表2．汎用化前後の工数比較

作業内容	汎用化前	汎用化後
仕様確認	1.0人日	1.0人日
外部設計／内部設計	4.0人日	0.0人日
変換エンジン開発	4.0人日	0.0人日
変換設定ファイル作成	0.0人日	1.5人日
単体テスト	4.0人日	1.5人日
結合テスト	1.0人日	0.5人日
合計工数	14.0人日	4.5人日

6. むすび

Web画面の開発工数を大幅に削減可能とする双方向型Web画面自動生成技術と，この技術を実装した双方向型Web画面自動生成ツールの，汎用化における実現方式とその効果に関して述べた。

これまでの双方向型Web画面自動生成技術は，プロジェクト固有の仕様書やソースコードのフォーマットに依存している部分が多く，プロジェクトごとに変換エンジンを開発する必要があり，他プロジェクトへ容易に適用できないという課題があった。そこで変換エンジンを汎用化することで，他プロジェクトへ容易に適用可能とした。評価の結果，他プロジェクトへ適用する際の工数を三分の一以下に短縮可能という効果が得られた。

現在，このツールが入出力可能なソースコードの言語はJSPだけであるが，今後，PHP(PHP：Hypertext Preprocessor)やASP(Active Server Pages)等その他の言語のソースコードを入出力可能とするように拡張開発し，さらに，適用範囲を拡大していく予定である。

参考文献

(1) 河村美嗣，ほか：双方向型Web画面自動生成ツールの開発，情報処理学会，第73回全国大会講演論文集，No.1，221～223（2011）
(2) 杉浦啓介，ほか：双方向型Web画面自動生成ツールの開発とその効果～設計書とソースコードの双方向変換～，情報処理学会，第74回全国大会講演論文集，5A-6（2012）
(3) 河村美嗣，ほか：双方向型Web画面自動生成ツールの開発とその効果～汎用化による適用範囲の拡大～，情報処理学会，第74回全国大会講演論文集，5A-7（2012）
(4) 大島正晴：双方向型Web画面自動生成技術，三菱電機技報，**86**，No.1，68（2012）
(5) 大島正晴，ほか：双方向型Web画面自動生成技術，三菱電機技報，**86**，No.6，349～352（2012）