2008－DPS－135 (6)
2008／6／19

画面遷移設計を基盤とした小規模Webシステム開発用 Web APIフレームワークとNST・褥瘡システム開発への応用

北野 優[†1,†2] 本間 圭[†1,†3] 高橋 佳嗣[†1] 上崎 達也[†1] 宮内 一平[†1]
齋藤 敬[†1] 鈴木 博勝[†1] 松田 豊臣[†1] 富樫 敦[†4]

Webアプリケーションを作成するためのフレームワークに類するものは数多く存在するが，それらの多くは大規模向けWebアプリケーション構築用であり，必ずしも中小規模のWebアプリケーション開発には適さない．本論文では，20ページ程度の中小規模Webアプリケーション開発に適したWebAPIフレームワークを紹介する．本フレームを使用し実際のWebアプリケーションを作成して実際の生産性の向上を示すと共に，他のWebAPIフレームワークとの比較・検討および評価を行った．

A Small-scale Web API Framework based on Page Transition Design and its Application to Development of NST-Bedsore System

You KITANO,[†1,†2] Kei HONMA,[†1,†3] Yoshitsugu TAKAHASHI,[†1]
Tatsuya UESAKI,[†1] Ippei MIYAUCHI,[†1] Takashi SAITO,[†1]
Hirokatsu SUZUKI,[†1] Toyoomi MATSUDA,[†1] and Atsushi TOGASHI[†4]

A variety of web application frameworks have been proposed for easy development of web-based systems. However, almost of them are designed for large-scale web applications. They are not always suitable to build small-size systems. In this paper, a small-scale web application framework is designed based on web-page transition, which is more suitable for small-size web systems. The paper describes the practical application to an NST-bedfore system and show practical improvement in development. The comparative studies of frameworks are given as well with respect to software design.

1 はじめに

アプリケーションフレームワークという言葉を利用する際に，フレームワークの意味を定義することなく誰もがほぼ共通の認識を持つことができるほど近年アプリケーションフレームワークの認知度は高まったといえる．アプリケーション開発を数回行った経験を有する技術者であれば，何らかのフレームワークと遭遇している可能性が高いだろう．フレームワークの有用性は認識され，現在，多種のフレームワークが開発されている．

フレームワークは，特定のソフトウェア問題を解決するクラスやインタフェースの集まりであり，フレームワークを利用することにより，問題を解決するために必要となる問題領域の分析やアプリケーション実装のための設計を再利用することが可能となることがフレームワークを利用する上で最大の利点と考えられる．フレームワーク利用のもう一つの利点として，フレームワークが提供しているユーティリティやツール類の利用ということも挙げられる．

一方，アプリケーションフレームワークにおいてはいくつかの問題点も指摘されている．1つ目に，Strutsに代表されるような多機能なフレームワークにおいては大規模な故のフレームワークの習得期間の長さが挙げられる．機能を十分に利用可能になるための期間だけでなく，設計を行えるようになるまでにはかなりの習熟期間を有する．

2つ目に，解決すべき問題の対象領域が異なっている場合，フレームワークに合わせることにより開発の要件を満たせない等の可能性が指摘される．

既存のフレームワーク (**表 1**) を調べてみると，Javaベースのフレームワークは最も実績があると思われるApache の Struts をはじめとして，JSF，Tapestry，Wicket 等多数ある．Java 以外のフレームワークとな

†1 宮城大学大学院事業構想学研究科
Graduate School of Project Design, Miyagi University
〒981-3298 宮城県黒川郡大和町学苑 1 番
E-mail: p0852008@myu.ac.jp（北野）
†2 有限会社アイ・シー・ティ
ICT Corporation
†3 日本アイ・ビー・エム株式会社
IBM Japan
†4 宮城大学事業構想学部
School of Project Design, Miyagi University

ると，それほど数が多いわけではないが，今回開発言語として利用したPerlのフレームワークであるCatalystやRubyを利用したRuby on Rails等それぞれ特徴を持ったものが存在する．

複数ある既存のフレームワークの再利用でなく，本研究でフレームワークを構築するに至った経緯として，小規模向けフレームワークであり，かつ，画面遷移設計を基盤とした開発手法に合致するような既存フレームワークが存在せず，他を適用しようとすると上記問題にあるような開発期間や要件を満たせない事象が発生すると考えたからである．

本論文では，今回構築したフレームワークの言語仕様を明らかにした後，本フレームワークを適応した事例の紹介と他のフレームワークとの比較・検討および評価を行う．

2 Webアプリケーション作成技法

Webアプリケーションの作成方法は大まかに考えて次の順番で行われる**図 1**.

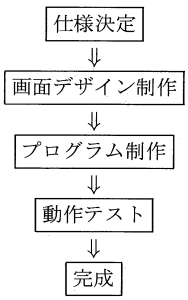

図 1: Webアプリケーションの作成手順

実際のWebアプリケーションの作成では，仕様が当初から厳密に確定できることは希である．作成途中でWebアプリケーションが完成してくると変更要望が出てくる．これは制作側が依頼側の要望を把握できなかったり，依頼側が要望を正確にまとめることができなかったり，結果が仕様設計時の予想と異なっている等原因は様々だが，再度仕様から作り直すことになる事例も珍しくない．最終的な仕様を早期に確定するには，依頼者に画面遷移を含めた画面イメージや出力サンプルで判断してもらう必要がある．

また，画面デザインをプログラムと完全に分離すれば，**図 2**のように完全にデザイナとプログラマの作業を分業することができる．これにより開発効率の向上が期待でき，完成してからのデザイン変更も容易である．従来のWebAPIフレームワークはhtmlデザイン部分にも繰り返し構文などを記載する必要があるものが多く，完成してからのデザイン変更はデザイナーの他にプログラマの作業も必要になっていた．

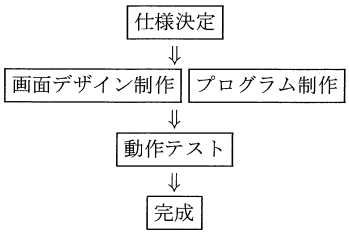

図 2: Webアプリケーションの作成手順 2

3 Web API フレームワークの言語仕様

3.1 環境

WebAPIフレームワークが動作する環境を**表 2**に示す．

表 2: サーバー環境

os	Linux
Web サーバー	Apache
データベース	PostgreSQL
言語	Perl

いずれのサーバーソフトウェアも，インターネットサーバーとして現在最も一般的に利用されているサーバーソフトウェアである．これらのサーバーソフトウェアは，Web2.0で標準となっているutf8と親和性が良く多国語対応がしやすい，商業利用にも制限がないフリーソフトウェアである．

3.2 構造

Web APIフレームワークの構造について**図 3**に示す．

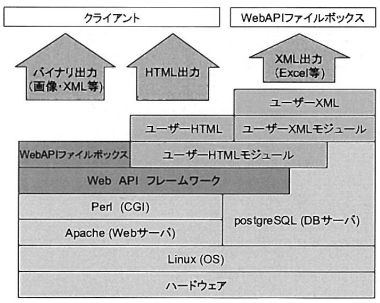

図 3: WebAPIフレームワークモデル

表 1: 主な WebAPI フレームワーク

名称	言語	説明
Struts	Java	Apache プロジェクトの一つ．MVC アーキテクチャを適用．JSP/Servlet を用いた開発をベースにしている．
JSF	Java	リクエスト駆動型の MVC Web フレームワークと異なり，コンポーネントベースのアプローチをとっている．
Tapestry	Java	Apache プロジェクトの一つ．コンポーネントベースであり，コード量が少なくて済む点が特徴．Java やネットワークの知識がないウェブデザイナーでも簡単に Java 製ウェブアプリケーションを作成できるという利点がある．
Wicket	Java	Apache プロジェクトの一つ．コンポーネントベースの軽量 Web アプリケーションフレームワーク．デザインとロジックの分離が明確なうえ，双方をジョイントする設定ファイルが不要という特徴がある．
Velocity	Java	Apache プロジェクトの一つ．フレームワークというよりは汎用テンプレートエンジン． MVC モデルに基づく Web サイト開発で，Java プログラマと Web デザイナが並行して作業できる．
Catalyst	Perl	MVC アーキテクチャが適応されており，コンテンツ，プレゼンテーション，フロー管理といった問題を簡単に切り分けて独立したモジュールにすることができる．既存の Perl モジュールの再利用を奨励している．
Ruby on Rails	Ruby	MVC アーキテクチャに基づいて構築されている．データベースを利用する Web アプリケーション開発を強みとしている．

WebAPI フレームワーク本体は perl で記述されており，またユーザーが作成するユーザーモジュールも perl で記述する．WebAPI フレームワークはユーザーモジュールを適切に呼び出し，Web アプリケーションとして成立させる．WebAPI フレームワークを使用した場合の一般的な応答を**図 4** に示す．http リクエストにより呼び出された WebAPI フレームワークが，適切なユーザーモジュールおよびユーザー html を呼び出し，最終的な CGI 応答である html を作成し応答出力する．

Web アプリケーションでは，クライアントに html の他にも画像やオフィスドキュメントなどの任意のファイルを http プロトコルを利用し応答出力する場合がある．WebAPI フレームワークでは予め出力したいファイルを WebAPI フレームワークファイルボックスに登録することにより，任意のファイルを出力することができる．**図 5**

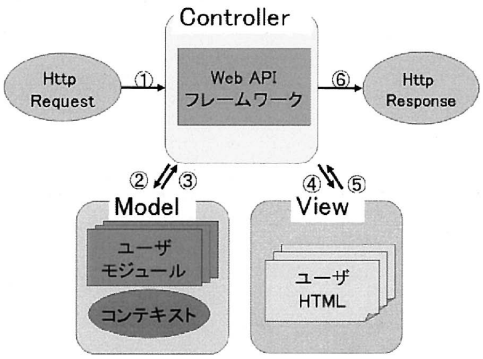

図 4: html 出力

3.3 機能

WebAPI フレームワークはユーザーモジュールに対して，いくつかの機能を供給するが，全機能の一覧を

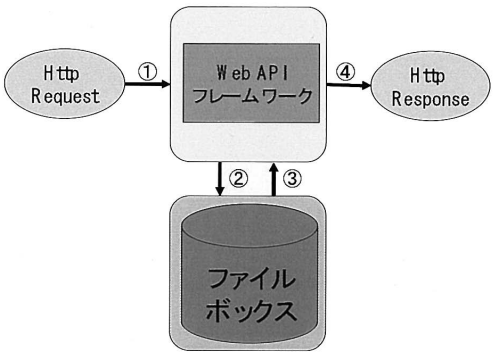

図 5: 任意のファイル出力

表 3 に示す.

表 3: 機能一覧

機能名	概要
html テンプレート	CGI 応答画面生成
ユーザー管理	認証処理
セッション管理	Web ページ間の連続性の維持
ファイルボックス	画像ファイルオフィスドキュメントを扱う
xml テンプレート	オフィスドキュメントなどの生成
その他ライブリ	和暦処理など

3.3.1 html テンプレート

html テンプレート機能は基幹となる機能で，WebAPI フレームワークを用いて作成する Web アプリケーションは，ユーザーモジュールとユーザー html から cgi 応答である html を作成する．ユーザー html は通常の html にユーザーモジュールが作成したデータの置換用タグなどを加えたものである．置換タグのルールを**表 5** に示す．タグには，その他にも perl コードを直接実行する機能も含まれており，データの置換後にコードを実行することができる．これにより，データ値の評価など簡単なプログラムは，ユーザー html 内に記載することができる．ユーザーモジュールおよびユーザー html の選択には，html 内の form 変数である sm および ss の値が直接使われる．ユーザーモジュールは，WebAPI フレームワークから呼び出されるが，呼び出される関数を**表 6** にまとめた．これらの関数内で，置換用ハッシュ変数である%kw::out_param に置換データを設定する．

表 4: ユーザーモジュール・ユーザ html 呼び出しルール

ユーザーモジュール	module/(sm の値).pm
ユーザー html	arc/(sm の値)_(ss の値).html

表 5: タグのルール

タグ	説明
$ ～変数名～ $	置換
@～ループ名～ {～@	ループ開始
@～} ～@	ループ終了
！～コード～！	コードの実行
＜！－－@ comment begin －－＞	コメント開始
＜！－－@ comment end －－＞	コメント終了

表 6: WebAPI フレームワークから呼び出される関数

関数名	説明
makeVal	配列ではない置換データを設定する.
ループ名_loadArray	ループ開始時に呼ばれる. ループ全体に関する置換を記載する. 戻り値は行数.
ループ名_makeVal	ループ時に呼ばれる. ループ時のみの置換を記載する.

3.3.2 ユーザー管理とセッション管理

ユーザー管理はユーザーモジュールの一つとして供給されるが，セッション管理機能と併せて Web アプリケーションの中でセキュリティの確保を行う．WebAPI フレームワークでは cookie にキーとなる乱数文字列を格納することによってユーザーやセッション情報を維持する．この機能によりユーザーモジュールからは特段意識することなくユーザー情報およびセッション情報を利用できる．

3.3.3 ファイルボックス

WebAPI フレームワークはファイルボックスというファイルを管理する機能を持っており，Web アプリ

ケーションが提供するhtml以外の画像ファイルやxml等の任意のファイルを扱うことができる．このファイルボックス内でもセッション管理機能は有効で，セキュリティ面の確保が行われている．ファイルボックス制御関数を関数を**表 7**にまとめた．

表 7: ファイルボックス制御関数

関数名	説明
getFileInfo(fileid)	ファイルの情報を得る．
delFile	ファイルを削除する
uploadFileFromForm	フォームからのファイルを登録する
uploadFileFromValue	変数をファイルの実態として登録する

3.3.4 xml テンプレート

近年xmlは様々なアプリケーションのデータ形式として採用されており，ワードやエクセルなどのオフィスドキュメントもxmlの形式に沿ったものもが使用できる．xml形式のファイルを生成しファイルボックスに登録する機能がWebAPIフレームワークには提供されている．この機能はhtmlテンプレート機能と類似しており，ユーザーxml，ユーザーxmlモジュールと呼んでいる．使用方法も似ているが，htmlテンプレート機能はWebAPIフレームワークから呼び出されるのに対してxmlテンプレートはユーザーモジュールから呼び出される点が異なる．

3.3.5 その他のライブラリ

WebAPIフレームワークはその他にPerlに不足している和暦関連の関数や数字を三桁毎に区切る関数などが用意されている．

4 応用事例

4.1 NST・褥瘡システム

WebAPI フレームワークの利用事例として，NST(Nutrition Support Team) システムの開発をあげる．NSTとは，医師，看護師，薬剤師，管理栄養士，臨床検査技師などの多職種で栄養サポートを実践するチームのことである．NSTシステムの開発は，宮城県立循環器・呼吸器病センターと宮城大学が協力して行っている．

しかしながら，前例の少ないシステムであることもあり，仕様が半ば宙に浮いた状態であり，設計，作成するも時間とともに仕様に微細な変化が起こってしまう．そのたびに設計仕様から変更することを余儀なくされたため，開発着手から1年と半年経った時点でもシステムの基本部分の作成が終わっていなかった．開発メンバーは数名いたが，デザインからプログラムまで実質一人で調整することになってしまっていた．それはデザインとプログラムが半ば融合してしまっていた点やWebアプリケーションの構造を無視したプログラムの共通化が行われていた為であった．

図 6: NST システム

そこで，このNSTシステム開発に本WebAPIフレームワークを利用して，作業を分担してみたところ，システムの基本部分の作成完了までにかかった期間はわずか3ヶ月足らずであった．画面遷移を基盤とする開発手法に合致する本WebAPIフレームワークを使ったことにより，画面のデータ変更などのわずかな仕様変更を素早く反映させることができたのが理由のひとつと考えられる．

NSTシステムの開発は，本WebAPIフレームワークの効果が顕著に現れた例のひとつである．

4.2 ME システム

WebAPIフレームワークを利用した事例として，ME（医療機器管理）システムがある．MEシステムは，昨今の医療危機管理の重要性に対する認識が高まっている現状をにらみ，薬事法や医療法の改正に伴う業務変化に対し，より作業効率を向上できるように開発されたものである．MEシステムの開発もNSTシステムの開発と同様，宮城県立循環器・呼吸器病センターと宮城大学が協力して行ったものである．

医療機器管理の実態を厚生労働省に報告する際に必要となる資料のファイル作成機能の開発において，

WebAPI フレームワークが提供する xml テンプレートにより大幅に工程が縮んだ.

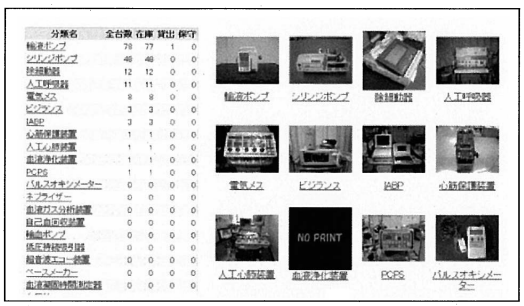

図 7: ME システム

5 評価・検討

前章までに実装した WebAPI フレームワークを評価する為に，既存の WebAPI フレームワークである Structs と比較検討した．以下にその概要を示す.

5.1 開発の手順

一般的な Structs の開発手順を次に示す.

1. html で画面をつくる.
2. 対応する Java クラスを作成する.
3. html を jsp に変更する.
4. 設定ファイルを記載する.

デザイナは一度 html でダミーのホームページを作成する必要がある．プログラマはその後仕様を確定させ，実際のクラスを作成しデザイナが作成した html を jsp に組み込みを行う．また一度作成した jsp のデザインを変更するには，デザイナとプログラマの両方の作業が必要である．設定ファイルも複雑で変更項目が多くあり，ページ構成が変わるたびに修正を要する.

一方，本 WebAPI フレームワークの開発手順は次の通りである.

1. ユーザー html を作成する.
2. 対応するユーザーモジュールを作成する.

対応するユーザーモジュールは機能が無いダミーモジュールで構わないので，画面遷移を含めた Web アプリケーションの動作イメージをデザイナーだけで WebAPI フレームワークに適合する形で開発することができる.

5.2 考察

先に述べた通り，Structs では，デザイナとプログラマの分業が完全にはできない．また，クラスを一定レベルまで作成しなければならないので，最低限作成しなければいけないプログラムや設定など作業が多い．一方，本 WebAPI フレームワークを使ったものはデザイナとプログラマの分業が完全にでき，簡単に行うことが出来る．しかしながら，Structs はページ数が多い比較的大規模な開発では当初から綿密な設計を行うので、効率が良いことが予想される.

6 おわりに

WebAPI フレームワークを実際に作成したが，その生産性向上効果は予想以上に大きかった．一方テストスイートや統合開発環境の問題など改善点も多く，WebAPI フレームワークを改良する意義は今後も高いと言える.

謝辞

本研究は，総務省戦略的情報通信研究開発推進制度 (SCOPE) 地域 ICT「中山間地を対象とした次世代ヘルスケアシステムを基盤とする地域振興に関する研究」(072302006) により一部支援を受けている.

参考文献

[1] 中所 武司, 津久井 浩. "予約業務を例題とした Web アプリケーション用フレームワークの再利用性の評価 (ソフトウェア工学)". 社団法人電子情報通信学会．電子情報通信学会論文誌. Vol. J88-D-I, No. 5(20050501) pp. 930-939

[2] 大澤 幸子, 遠山 元道 . "SuperSQL クエリ作成支援系における情報容量の提示 (セッション 1C : ツール)". 社団法人情報処理学会 . 情報処理学会研究報告．データベース・システム研究会報告. Vol. 2003, No. 71(20030716) pp. 73-80

[3] 独立法人情報処理推進機構　セキュリティーセンター "安全なウェブサイトの作り方"