JAIST Repository
https://dspace.jaist.ac.jp/

Title	フレームアークの進化に対応可能なWebアプリケーション作成手法の研究
Author(s)	呉，暁雷
Citation	
Issue Date	2013-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/11327
Rights	
Description	Supervisor:鈴木　正人，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

フレームワークの進化に対応可能な Web アプリケーション作成手法の研究

呉 暁雷(1010207)

北陸先端科学技術大学院大学 情報科学研究科

2013 年 2 月

キーワード: フレームワーク, Web アプリケーション, 開発手法, 要求グラフ, 実現グラフ.

1 研究の背景と目的

Web アプリケーション開発においては、フレームワークの利用が一般的である。しかしながら、フレームワークを構成する要素技術は変化が激しいため、フレームワークは、機能拡張及び変更を継続的に行っている。フレームワークの進化によって、旧バージョンのフレームワークで作成されたアプリケーションはそのままで新バージョンのフレームワークでは動作しないという問題が発生している。このような場合には、アプリケーションの変更に正しく反映する必要がある。しかし、変更を行うには開発者は進化前と進化後の両方のフレームワークに精通していなければならず、変更作業のコストが増大している。この問題の解決策として、フレームワークの進化に対応するようにアプリケーションを変更するための体系的な手法が必要である。

本研究の目的は、フレームワークの進化に対応するアプリケーションの変更を支援し、変更コストを削減するためにアプリケーションの変更手法の提案を行う。変更手法として、アプリケーションの要求を構成する要素と実現を構成する要素及び要素の対応関係の 3 種類の構造に注目し、要求の構造をグラフで表現したものを要求グラフ、実装の構造をグラフで表現したものを実装グラフ、要求の要素と実装の要素の関連を表現したものを要求－実現－関連グラフとして定義する。

Copyright © 2013 by Wu Xiaolei

2　フレームワークを使用した開発とその問題点

Web アプリケーションフレームワークを使用してアプリケーションを開発するとき、すべてのアプリケーションで共通となる画面遷移や入出力の機構をフレームワークが提供してくれる。このため、開発者は業務固有の部分を作成に集中できる。フレームワークが提供する機能を利用して画面などを作成することでプログラマにより実装の方針を統一しやすくなる。アプリケーションの開発効率と保守性が向上になる。

フレームワークの進化によって、フレームワークの構造や仕組みが変更された。たとえば、フレームワークの決めている設定ファイルの記法や意味が変更された。また、業務固有処理や画面の実装方法のルールが変更された。それが原因で、旧バージョンのフレームワークで作成されたアプリケーションはそのままで新バージョンのフレームワークでは動作しないという問題が発生している。これを解決するには、フレームワークの進化により修正が必要となる箇所をアプリケーションに正しく反映する必要がある。ただし、変更を行うには開発者はフレームワークの進化前後の両方の知識に精通していなければならず、変更作業が困難になっている。また、フレームワークが変更されることを継続的に行っている。このために、フレームワークが変更されるたびにアドホックな修正を繰り返すとアプリケーションの継続的に変更が行われ、その結果としてアプリケーションの構造が複雑化し、保守のコストが増大する。そのため、フレームワークの進化に対応する体系的な変更手法が必要である。

3　フレームワークの進化を考慮した開発手法の提案

本研究では、問題を解決するためにフレームワークの進化を考慮した開発手法を提案する。従来は仕様から要求分析を行った結果をユースケースなどで記述していたが、提案手法では、(1)要求を構成する要素、(2) 実現を構成する要素、(3)要素の対応関係の 3 種類の構造に注目し、要求－実現－関連グラフを作成する。要求－実現－関連グラフとは、要求構成要素と実装構成要素の間に存在する関連を辺として持つグラフである。要求構成要素は最初に要求分析を行い、主に機能要求の詳細を定義することで得られる。要求分析とは、システム開発の対象となる業務を調査し、開発するシステムの利用方法について分析することである。実装構成要素とは、アプリケーションを構成する部品であり、個々の部品は要求を満たすために必要な機能あるいはデータ実体を表すものである。

あるフレームワークを使用してアプリケーションを開発する場合を考える。要求に変更がない場合でも、フレームワークが進化すれば、アプリケーション実装時に必要な部品の変更により、各フレームワークに対応した実装の構成要素が異なり、それは要求－実装－関連グラフの構成に反映される。まず、要求を構成する要素を確定し、要求

構成グラフを作成する。そして、フレームワークの進化前後に対して、それぞれの実装を構成する要素を確定し、実装構成グラフを作成する。ここで、実装を構成する要素の具体的な実現内容を記述する必要がある。そして、異なる実装構成グラフを比較し、変更が必要な箇所を抽出し、要求－実現－関連グラフ上での変更ルールとして定義する。このように、グラフの変更を実装の変更に変化することにより、フレームワークの変更をアプリケーションの変更に正しく反映させることができるようになる。

4　適用事例

提案手法の有効性を確認するために適用事例としてホテル予約管理管理システムを対象とする。使用するフレームワークは Struts を例とし、バージョン 1 からバージョン 2 に変更された場合のアプリケーションの変更を提案手法に基づいて行い、評価する。対象とするシステムでは、2 つのデータベースがあり、予約記録 DB と客室 DB である。また、5 つの機能があり、「ユーザーを登録する」、「ユーザーを認証する」、「予約を作成する」、「予約を照会する」と「解約する」である。1 つの予約記録には、1 つのセットルームのインフォメーションがある。予約作成と解約するたびに、2 つのデータベースを更新することが必要である。

要求構成グラフを作成することにより、Struts1 における実装構成グラフと Struts2 における実装構成グラフを比較し、変更が必要な部分を抽出し、要求－実装－関連グラフ上での変更ルールとして定義する。その結果、(1)モデルのアクション内部への移動と(2)ActionForm Bean のアクションへの統合 2 つの変更規則を定義した。

次に、対象とするホテル予約管理システムの機能を追加した場合に、グラフを利用しない場合と利用した場合のアプリケーション変更コストを比較することで、変更規則を評価する。まず、対象とするホテル予約管理システム機能追加の仕様から、要求分析した結果より、要求構成グラフを作成した。そして、Struts1 を使用する場合と Struts2 を使用する場合それぞれの実現構成グラフを作成した。変更規則に従って、Struts1 で追加必要な実現構成要素を変更したら、Struts2 で追加必要な実現構成要素が得られた。このことより、変更規則とそれを利用した提案手法の有効性を示した。

5　まとめと今後の課題

本研究では、フレームワークの進化に対応可能な Web アプリケーション作成手法について、グラフを利用した作成及び変更手法の提案を行った。アプリケーションの要求を構成する要素と実装を構成する要素及び要素の対応関係の 3 種類の構造に注目し、要求－実装－関連グラフを提案した。進化前のフレームワークで作成されたアプリケーションと、進化後のアプリケーションにおいて、要求－実現－関連グラフを比

較し、変更箇所を整理し、変更ルールとして抽出する。提案手法に利用して、具体的な事例に対して適用することにより、提案手法の有効性を確認した。

ルールを改善するために、Struts の他の機能（セキュリティなど）を使用するアプリケーションに対する事例を収集する。また、他のフレームワーク（Spring など）に対する事例の収集及び変更ルールの発見と抽出することは今後の課題である。