5H-05

Ruby on Railsを対象としたモデル検査による画面遷移検証

大嶋　乃斗[†1]　　岸　知二[†2]

[†1] [†2] 早稲田大学大学院 創造理工学研究科 経営システム工学専攻

1. はじめに

近年、インターネットの発展に伴い、重要な処理を扱うWebアプリケーションが増加しており、その開発において早期の段階で設計誤りを発見することの重要性も増している。そうした設計検証の手法の1つとして、モデル検査技術が注目されている。モデル検査を行うためには利用するモデル検査器に応じたアプリケーションのモデル化が必要であるが、その作業は通常のソフトウェア開発者には難しいという課題がある。本稿では、Webアプリケーションの基本構造を規定する画面遷移に注目し、フレームワークRuby on Railsにより作成したWebアプリケーションの定義から、画面遷移の検証を行うモデルを自動生成してモデル検査を行う手法を提案する。

2. 関連研究

Web アプリケーションにおける画面遷移について、モデル検査によって検証している研究の例として、以下を挙げる。本間ら[1]の研究では、Web アプリケーションの設計で用いられる画面遷移図に着目し、画面遷移とシステム環境(アプリケーション内部の状態)を 2 つの有限状態オートマトンとして捉え、その直積オートマトンによってWebアプリケーションをモデル化する手法を提案している。この手法では、実装部分ではなく、画面遷移図から検証モデルを導出する方法を提案している。藤原ら[2]の研究では、Webアプリケーションフレームワーク Strutsに着目し、それを使用したWebアプリケーションにおいて、ユーザの誤使用によって異常が発生した状況でも正常動作することを検証するために、不適切な振る舞いを含めたモデル化手法を提案している。Strutsでは、画面遷移情報を示すJSPや設定ファイル(struts-config.xml)のような記述形式が統一されている定義ファイルが存在し、この研究ではその定義ファイルに着目して、検証モデルへの変換を行う手法を提案している。

3. 研究対象

本研究で扱う研究範囲は、フレームワーク Ruby on Railsを用いたWebアプリケーションを対象とする。Ruby on Railsは MVC 設計に基づき、Model、View、Controller のそれぞれの要素と、それを定義するファイルの位置や全体のディレクトリ構造が定まっている。そのうち、中核となるディレクトリは app である。app 内部のディレクトリ構造を以下の図1に示す。また、Ruby on Railsは「設定より規約」という特徴があり、規約に基づき開発していくことで記述量を減らし、効率的に開発していくことができるフレームワークである。

図1. appのディレクトリ

Ruby on Railsは、Strutsのように設定ファイルなど記述形式が統一されている定義ファイルが存在しないため、[2]の XML 形式の設定ファイルから自動で画面情報を抽出するといった方法は適用できない。そこで、本研究では、このあらかじめ決まっているディレクトリ構造や規約に着目して、検証モデルへの変換を行う手法を提案する。

本研究で着目する規約の一部について説明する。View 内の各ファイル名は Controller クラスに記述されるアクション名と同じ名前にするという規約がある。また、View 内の各ファイルは「app/views/Controller 名」のディレクトリ内に配置することが決められている。この規約に着目し、画面情報の抽出を行う。また、View 内の各ファイルでは、ソースコードの link_to メソッド内に、遷移先のアクション名を記述することでその画面における遷移情報を定義するため、そこからも抽出を行う。

本研究では、対象をモデル化しやすいように規約のみでなく、特定の記述方法に従い定義されたものを対象とする。また、今回、Model クラスのデータに依存した遷移や、複数の Controller、外部へのリンクについては考慮しない。

4. 提案手法

4.1. 提案手法概要

Ruby on Railsに基づくWebアプリケーションにおいて、MVC 設計に着目し、画面遷移情報をControllerと Viewから抽出し、モデル化を行う。提案手法の流れを以下の(1)～(3)、モデル化までの提案手法の全体像を図3に示す。

(1) Controllerクラスと Viewクラスから画面情報と遷移情報を抽出
(2) 抽出したそれぞれの情報から検証モデルを作成
(3) モデル検査により性質を検証

「A Model-checking Method for Ruby on Rails Applications focusing on Screen Transitions」

[†1] Naito Oshima・Waseda University

[†2] Tomoji Kishi・Waseda University

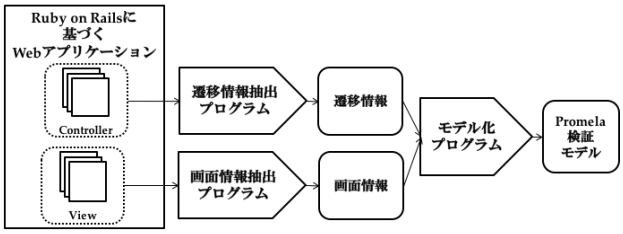

図 3. 提案手法の全体像

4.2. 画面遷移情報の抽出

画面遷移情報は画面情報と遷移情報から構成され、画面情報には、画面とその画面から実行可能なアクション、遷移情報には、アクションとそのアクションで遷移する画面の情報が定義される。app 内部の MVC 設計の V(View)と C(Controller)に該当する部分から画面情報と遷移情報をそれぞれ抽出する。

具体的には、View のファイルは、表示される画面のテンプレートファイルで構成されており、そのソースコード内の link_to メソッド内から、どの画面からどのアクションが実行可能かを抽出する。次に、controllers 内の Controller ファイルのソースコードから、アクション名と遷移先の画面情報を抽出することで、どのアクションが存在するかを示す。

4.3. 検証モデルの作成

4.2 の方法でそれぞれ抽出された画面遷移情報を入力として、モデル検査を行うためのモデルに変換する。本研究では、モデル検査に SPIN を用いて行うため、その仕様記述言語である Promela によって、検証モデルを記述する。以下の図 4 に、画面遷移情報から Promela で記述した検証モデルへの変換ルールを示す。

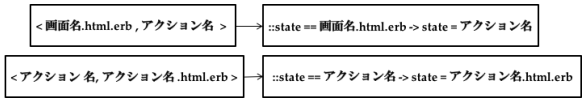

図 4. 画面遷移情報の変換ルール

4.4. SPIN による検証

4.3 で作成した Promela で記述した検証モデルを入力とし、SPIN により、到達可能性の検証を行う。

5. 評価実験

5.1. 実験概要、目的

4 で提案した検証手法を用いて、サンプルプログラムの画面遷移部分を対象とした検証を行った。サンプルプログラムを用いて実験を行うことで、画面遷移の定義から検証までの提案手法が妥当であるかどうか、実際に検証ができるかどうかの実現性を確かめる。

5.2. モデル化支援ツールの実装

評価実験を行うにあたって、今回、View から画面情報、Controller から遷移情報を抽出し、その画面情報と遷移情報から検証モデルを自動で生成するモデル化支援ツールを実装した。

5.3. 検証例題

今回検証する例題として、作成したサンプルプログラム(実験(1))の画面遷移図を以下の図 5 に示す。今回の例題はコントローラー数 1 とし、画面遷移は Ruby on Rails のファイル内のリンクのみとし、外部ページへの URL へのリンクはないものとした。次に、検証手法に問題ないかどうかを確かめるために、実験(2)とし、わざと誤りを混入させ、注文完了画面からの action がない(Top ページへ戻れない)という画面遷移の行き詰まりがあるプログラムに書き換えたものに対しても、同じくこの検証手法を適用し、評価実験を行った。どちらも検証手法で提案した検証項目に対して、検証を行った。

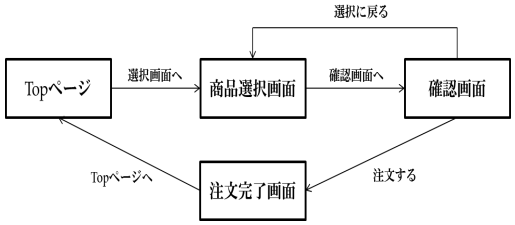

図 5. サンプルプログラムの画面遷移図

5.4. 実験結果

実験(1)では、提案手法を用いることで、例題のサンプルプログラムの画面遷移部分の検証を行うことができた。試作した支援プログラムを用いることで、画面遷移情報の抽出と、抽出した情報からモデル化手法を自動化する事ができることも確かめられた。実験(2)では、同じ検証手法を用いて、画面遷移で行き詰まりがあるというエラーが検出されたため、望まれる結果を得ることができた。

6. 今後の課題

本研究では、フレームワーク Ruby on Rails を用いた Web アプリケーションの画面遷移に焦点をあて、抽出した画面遷移情報から Promela で記述されたモデルを自動生成し、モデル検査 SPIN により検証を行う手法を提案した。この提案手法によって、実際に評価実験を行い、例題のサンプルプログラムで検証を行うことで、Web アプリケーションの定義から検証可能なモデルを作成できることが確かめられた。

しかし、本稿では画面遷移の定義方法を特定の記述のものに限定したため、実務的なアプリケーションへの適用としてはまだ不十分である。画面遷移の定義方法の制限を減らし、適用範囲を広げていくことが今後の課題と考えられる。また、複数の Controller や Model クラスといった点も考慮してモデル化を行うことで、より正確に、モデル化や検証を行うことができると考えられる。

参考文献

[1] 本間圭，高橋薫，富樫敦「形式的手法による Web アプリケーションのモデル化と検証」，電子情報通信学会，Vol.109，No.41，pp.43-48，2009.

[2] 藤原貴之，岡野浩三，楠本真二「SPIN による Struts アプリケーションの動作検証を目的としたモデル生成手法の提案」，電子情報通信学会，Vol.105，No.491，pp.73-78，2005.