

MVC Web アプリケーションの機能縮退を考慮した振舞い 検証

澤野 宏貴^{1,a)} 前澤 悠太^{2,b)} 高橋 竜一^{3,c)} 本位田 真一^{2,4,d)}

概要: Web アプリケーションを保守する時、保守担当者が機能を部分的に制限することがある（機能縮退）。保守担当者は、機能縮退に応じたナビゲーション要求（例えば、Web ページへの到達可能性）が満たされていることを期待するが、近年の動的な Web アプリケーションの振舞いを人手で確認することは難しい。既存手法を用いると実装コードを入力に振舞いを自動検証できる。しかし、機能縮退後の振舞いに対しては、それに応じたコード修正が必要であり、誤った縮退操作を計画すると不要なコード修正が発生するため、保守コストの増大につながる。そこで本論文では、MVC Web アプリケーションからナビゲーションモデル（NM）を抽出し、この NM 上で機能縮退を表現するツールを提案する。縮退対象の機能に関する情報は、保守担当者から与えられる。また本ツールは、検証式テンプレートをを用いてナビゲーション要求を半自動で検証する。保守担当者はアプリケーション依存の値（例えば、Web ページ間の最小リンク数）を入力できる。オープンソースな実アプリケーションである CandyCane を用いて適用事例を実施し、提案ツールが誤った縮退計画に対し、要求違反を出力できることを確認した。したがって、保守担当者が提案ツールによって縮退計画を改善し、正しく機能縮退できると考えられる。

1. はじめに

近年、Web アプリケーションは広く普及し、その数は増加の一途を辿っている。Web アプリケーションはインターネット上で電子商取引などの様々なサービスで利用されており、我々の生活の必須基盤となっている。

Web アプリケーションを保守する時、三菱東京 UFJ など^{*1}のように、保守担当者が機能を部分的に制限し運転することがある（機能縮退）。機能縮退すると Web アプリケーションの Web ページとそれらのリンク関係を表すナビゲーションが変わり、Web ページの到達可能性などのナビゲーション要求 [1], [2] が満たされなくなる可能性がある。すると、停止中の DB へアクセスしアプリケーション

がクラッシュするなどの致命的な問題につながる。

Web アプリケーションのナビゲーションはナビゲーションモデル（NM）で表現できる。NM は Web ページを状態とみなし、ユーザアクション（リンクやボタンのクリックなど）によって状態を遷移する状態遷移図である [2]。近年の Web アプリケーションは、ユーザリクエストに応じて実行時に Web ページを生成する動的性を持つため、保守担当者が正確に振舞いを理解し NM を作成することは難しい。そこで、既存手法では実装コードから NM を抽出し、ナビゲーション要求を満たすか検証する支援をしている [3], [4]。しかし、既存手法は実装コードを入力に NM の抽出をするため、縮退運転時の性質検証には、機能縮退のための実装（縮退実装）が必要である。縮退計画が誤っていると、性質が満たされるまで縮退実装・既存手法による検証を繰り返す必要があり、コストが高い。

そこで本研究では、MVC デザインパターンで設計された Web アプリケーション（MVC Web アプリケーション）を対象とし、機能制限された NM（RNM）を抽出・検証するツールを提案する（3章）。提案ツールの概要を図 1 に示す。ステップ 1（抽出）：提案ツールではまず、MVC Web アプリケーションを静的に解析し、NM と不変条件を抽出する。ステップ 2（制限）：保守担当者の縮退計画から得られる制限情報（制限するビューなど）を入力とし、抽出

¹ (株)日立製作所 インフラシステム社
Hitachi, Ltd., Infrastructure Systems Company, Shinagawa,
Tokyo 140-8572, Japan

² 東京大学
The University of Tokyo, Bunkyo, Tokyo 113-8656, Japan

³ 早稲田大学
Waseda University, Shinjuku, Tokyo 169-8555, Japan

⁴ 国立情報学研究所
National Institute of Informatics, Chiyoda, Tokyo 101-8430,
Japan

a) hiroki.sawano.dg@hitachi.com

b) maezawa@nii.ac.jp

c) ryu1-t@fuka.info.waseda.ac.jp

d) honiden@nii.ac.jp

*1 mzw.jp/yuta/research/ex/failsoft/examples.html

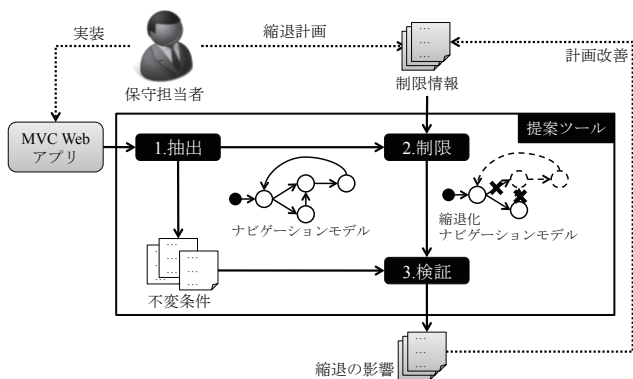


図 1 提案ツールの概要

した NM に制限を加え RNM を得る。ステップ 3 (検証) : RNM をモデル検査器 NuSMV (nusmv.fbk.eu) を利用して検証する。検証性質は、NM から抽出した不変条件と与えられた制限情報から生成する。最後に保守担当者は、検証結果より縮退計画がナビゲーションに与える影響を分析することで、縮退計画を改善し縮退実装できる。

本研究の貢献は次の通りである。1) 機能縮退のための MVC Web アプリケーションの静的な RNM 抽出・検証手法を提案し、2) 提案手法をツールに実装した。また、3) 実アプリケーションを用いた適用事例で、提案手法の有用性を評価した。

2. 背景

2.1 MVC デザインパターン

MVC デザインパターンは Web アプリケーションの設計に広く利用されている [4], [5]。MVC Web アプリケーションでは、データと手続き (モデル) をユーザが直接参照する情報から分離する (ビュー)。コントローラはリクエストを処理し、モデルおよびビューを制御することで、それぞれ主にデータの取得・保存、描画するビューの決定を果たす。コントローラは複数のアクションで構成される。

MVC Web アプリケーションではユーザリクエストをアクションが処理する。この仕組みから機能縮退はアクションの制限だと考えられる。提案ツールでは、MVC デザインパターンを採用したフレームワーク CakePHP (cakephp.org) で実装した PHP Web アプリケーションを対象とする。

2.2 機能縮退とナビゲーション要求

Web アプリケーションを保守する時、保守担当者は機能を部分的に制限することがある。機能縮退のために MVC Web アプリケーションのアクションを制限すると、Web アプリケーションのナビゲーションが変化する。ナビゲーションとはユーザが訪れることのできる Web ページの列であり、通常、次のページは現在のページとユーザアクションによって決定される [2]。

Web アプリケーション開発では、ナビゲーションは豊かなユーザ体験を与えるために重要視される [6]。ゆえに、

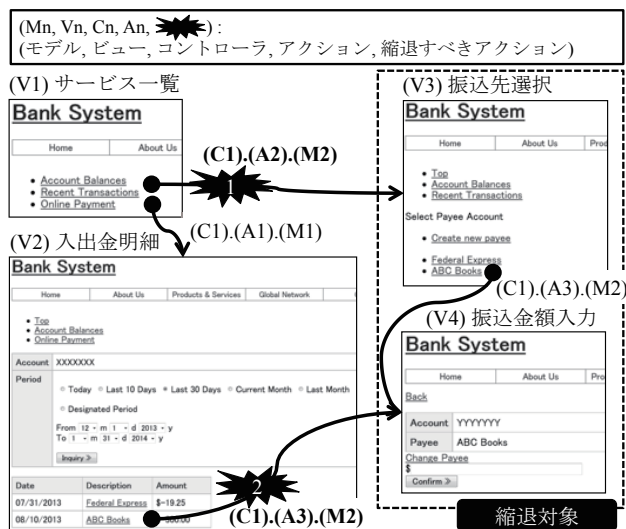


図 2 動機付けの例のスクリーンショット

設計者はナビゲーション構造の構築に多大な労力を費やす [7]。この時、Web ページの到達可能性などの要求が関心事となるが [1], [2]、縮退運転時もナビゲーション要求があると考えられる。もしナビゲーション要求を無視した縮退運転を行えば、例えばユーザが停止した DB にリクエストを送信する Web ページに到達できてしまい、アプリケーションがクラッシュするといった致命的な問題が発生する。したがって、保守担当者は機能縮退する際、ナビゲーション要求が満たされていることを保証しなければならない。

ナビゲーションをモデル化した NM は、ナビゲーションに関わる性質検証に役に立つ。しかし近年の Web アプリケーションは、ユーザリクエストに応じて実行時に Web ページを生成するといった動的性を持つため、保守担当者は NM を作成することは難しい。そこで、実装から設計を回復するリバースエンジニアリング技術が役立つ。Web アプリケーションの動的性を考慮して、多くの既存手法は動的解析に基づく [8], [9]。しかし、動的解析は実行環境に依存するため、ナビゲーション要求に反する状態や遷移を見落としかねない。そこで、実装コードを静的に解析し、NM の抽出・検証する手法が提案されている [3]。しかし、本研究で着目する機能縮退時の振舞い検証には縮退実装が必要であり、既存手法が保証するまで縮退実装を繰り返すことはコストが大きい。そこで我々は、縮退実装なく RNM を獲得し、縮退運転時の性質検証ができるツールを提案する。

2.3 動機付けの例

本研究の動機付けの例として、インターネットバンキング Web アプリケーションを示す (図 2)。この例では、次の 3 つのサービスを提供する。1) 残高照会サービス: 口座の残高を照会する。2) 入出金明細照会サービス: 指定した期間内の入出金明細を表示し、明細から振込先を指定できる。3) 振込みサービス: 指定した振込先に振込みを行う。また、過去の振込先を指定できる。

保守のために振込みサービスを一時的に制限する計画を考えると、次の要求を満たさなければならない。要求1：振込みサービスに関わる Web ページ（図2の破線内）に到達できないこと。要求2：残高照会サービスと入金金明細照会サービスに関わる Web ページに到達できること。

図2より、振込みサービスへの経路は次の2通りある。一方の経路では、(V3)から振込先を選択し(V4)に遷移する。他方では、(V2)から過去に利用した振込先を指定し(V4)に至るが、取引の有無や期間の指定によってはこの経路は現れないため、この経路を見逃す恐れがある。保守担当者はプログラムを精査し全ての状態遷移を把握する必要があるが、実行時に決まる Web アプリケーションの動作から要求に反する全ての経路を発見することは難しい。

3. 提案ツール

3.1 MVC 構造に着目したナビゲーションモデルの抽出

提案ツールではまず、MVC Web アプリケーションのソースコードを解析し、NM を静的に抽出する。縮退計画では MVC デザインパターンの主要素であるモデルとビュー、コントローラそれぞれが制限対象になると考えられるため、NM にこれらオブジェクト間の関係を表現する。

抽出する NM では、ビューを状態とみなし、状態はユーザアクションが呼び出すコントローラのアクションによって遷移する。各遷移のラベルは、その遷移中に実行されたコントローラとアクションの組および参照されたモデルの集合を保持する。初期状態はひとつだけ与えられ、外部サイトへの遷移は扱わない。CakePHP の仕様^{*2}に基づき、モデルとビュー、コントローラを次のように解析する：

モデル：モデル同士は、アソシエーション機能を通じて、\$hasOne や \$hasMany などに関連を持ち、他のモデルにアクセスできる。そのため、関連モデルを参照するメンバを解析する。さらに、関連モデルのエイリアスを解決する。

ビュー：リンク先のアクションを決定するために、リンクやフォームの生成に利用される `HtmlHelper::link()` や `FormHelper::create()` などのコアヘルパーのメソッドを解析する。これに加え、開発者が実装したヘルパーメソッド（アプリ依存ヘルパー）がリンクを生成する場合がある。これらは CakePHP の仕様からアクションを決定できないため、次の情報の入力を求める：1) 戻り値となるコントローラとアクション。2) 内部で使用されるリンクを生成するヘルパーメソッド名。3) リンク情報を渡す引数の位置。このようなメソッドは通常、その内部で前述のコアヘルパーを使用しているため、コアヘルパーのメソッド名でキーワード検索し情報入力が必要なメソッドを特定できる。

部品化されたビューである要素は `element()` によってビューに読み込まれる。要素が持つリンクはそ

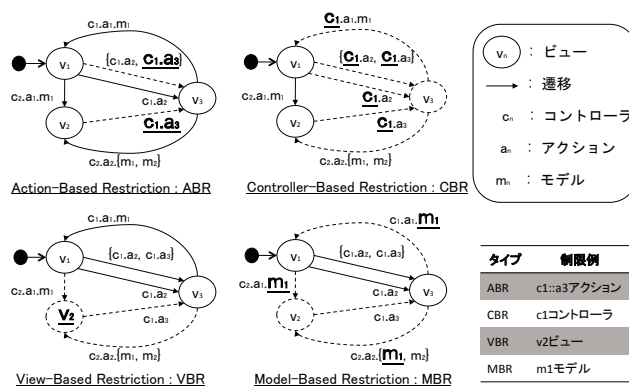


図3 抽出したナビゲーションモデルの制限種別

の要素を読み込むビューが持つ。多くのインタフェースをラップするレイアウト (`$layout`) は、レイアウトが適用されるビューからアクセスできる。アクション名などが変数を評価する場合、ビューでの変数定義に加え、`Controller::set()` でコントローラから渡されたデータを利用する。以上のように、解析を実行することでアクションとその遷移元状態を決定する。

コントローラ：ビューからのリクエストはコントローラ内部で単一のアクションを呼び出す。各アクションはリクエストに応じて描画するビューを選択するため、`render()` を解析することでアクションの遷移先状態を決定する。また、フローのコントロールには `redirect()`、メッセージ表示後のリダイレクトに `flash()` が利用され、これらも同様に遷移と捉える。リクエストが DOM (www.w3.org/DOM) の操作を目的とした Ajax の非同期通信であっても、アクション呼び出しによって状態は遷移する。その時にクライアントに出力するビューや要素はひとつの状態とし、同期的に遷移する状態と区別しない。また、コントローラ変数の `$uses` から参照するモデルを解析し、その参照情報をアクションごとに保存する。

ビューとコントローラの解析結果から状態間を結び、状態間の遷移上にはラベルとして、リダイレクト先を含む実行されたコントローラとアクションの組および参照したモデルの集合を保持する。

3.2 ナビゲーションモデルの制限

提案ツールでは、抽出された NM の最小単位であるアクションに対して機能制限を表現する。保守担当者の制限方法を次の4つに整理する（図3）：**ABR**：特定のアクション。**CBR**：特定のコントローラのすべてのアクション。**VBR**：特定のビューを呼び出すすべてのアクション。**MBR**：特定のモデルを参照するすべてのアクション。提案ツールは、保守担当者から指定された制限対象に応じて、抽出された NM 上の該当箇所を特定し取り除く。図3の破線は、保守担当者によって指定された機能制限を表す。

^{*2} book.cakephp.org/2.0/_downloads/en/CakePHPCookbook.pdf

表 1 検証カテゴリと CTL 式 [2]

ID	カテゴリ	CTL 式
P1	到達可能性	EF (ViewVar = ViewName)
P2	デッドロック	AG EF (ViewVar = ViewName)
P3		AG ((ViewVar = View1Name)->EF (ViewVar = View2Name))
P4	到達ステップ	AG ((ViewVar = View1Name)->AX (ViewVar = View2Name))
P5		AG ((ViewVar = View1Name)->EX (ViewVar = View2Name))
P6		AG ((ViewVar = View1Name)->EX EX (ViewVar = View2Name))

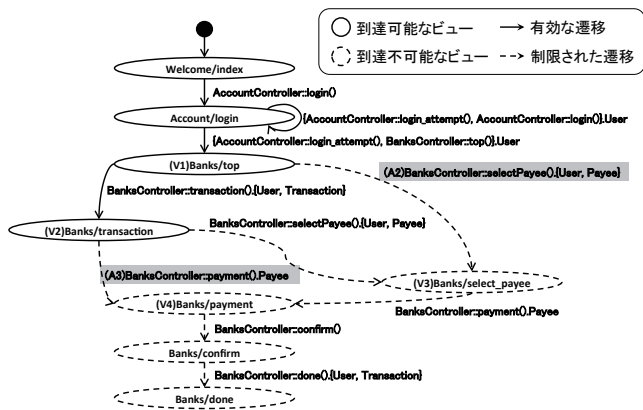


図 4 動機付けの例への適用結果

2.3 章で述べた通り，保守担当者が機能制限したい実装箇所を正確に指定することは難しい．そのため提案ツールでは，機能要求とその実装箇所を関連付ける要求追跡マトリクス (Requirements Traceability Matrix : RTM) [10] を利用する．開発者は，MVC Web アプリケーションに機能 (振込みサービスなど) を実装する際に，その実装箇所 (ビューやアクションなど) を RTM に記述する．保守担当者は，RTM から制限したい機能を選択することで，提案ツールはその実装箇所を特定できる．このようにして，提案ツールは保守担当者の制限箇所の指定を支援する．

3.3 縮退運転時のナビゲーション要求検証

提案ツールは，RNM の正しさを検証するために NuSMV モデル検査器を利用する．一般的な RNM の正しさは検証できないため，提案ツールではナビゲーション要求の性質 [2] に対して検証する．表 1 に示す通り，これらの性質を表す CTL 式のテンプレートを得られる．また，テンプレート変数 (ViewVar) は滞在中のビューを示す状態変数であり，ViewName は検証するビューである．

表 1 の性質 P1 と P2 を検証することでそれぞれ，期待する状態に到達可能か，他の状態に遷移できない状態がないか，確認できる．これらの性質は，抽出した状態集合を用いて要求に依存することなく定義できる．一方，性質 P3～P6 を定義するには要求に依存するパラメータが必要となるため，保守担当者に特定のビューの到達にかかる最小リンク数などの入力を求める．

3.4 動機付けの例への適用結果

提案ツールが抽出した RNM の一部を図 4 に示す．2.3

表 2 CandyCane の構成ファイル数と解析結果

対象	全体数	解析できた数
モデル	51	51
ビュー	97	95
エレメント	64	61
レイアウト	1	1
コントローラ	29	29

章で示した要求 1 は，振込みサービスに関わる Web ページに到達できないことである．そこで，制限情報として (V3) 振込先選択のビューなどを指定する (図 3 の VBR)．すると提案ツールは，図 4 の破線部分を制限された機能と特定し RNM を抽出する．

提案ツールはさらに，到達可能性 (表 1 の性質 P1) を検証する．その結果から，抽出された RNM が要求 1 を満たしていることがわかる．また保守担当者は，図 4 の破線の遷移ラベルから，縮退すべきアクションを判断できる．2.3 章で見逃す恐れがあると述べた状態遷移も，制限すべきアクションであると発見できる．

4. 評価実験

提案ツールの有用性を評価するために，次の研究課題を設定した．適用事例には，CandyCane (yandod.github.io/candycane) を利用した．

RQ1 縮退運転時の振舞い検証によって縮退計画の成否を判断できるか？

RQ2 実用可能な時間で検証できるか？

4.1 実験手順

まず我々は，3.1 章のアプリ依存のヘルパーを特定するために，22 個のメソッド情報を定義した．そのうえで，解析できたファイル数と全体数を表 2 に示す．

続いて，図 3 の制限種別を網羅する縮退要求と RTM (表 3) を縮退計画として用意し提案ツールに入力した．提案ツールの実行結果から，入力した縮退計画の影響が及ぶ MVC それぞれ (表 4 の 2 と 3, 4 列目) と抽出した NM 上で削除された遷移の遷移元状態数と削除された遷移数 (表 4 の 5 と 6 列目) を確認した．また，提案ツールの実行時間を計測した (表 4 の 7 列目)．最後に，提案ツールが誤った縮退計画の性質違反を検出できることを確認した．

4.2 結果と考察

4.2.1 縮退運転時のナビゲーション要求違反の検出 (RQ1)

到達可能性 : 到達可能性の要求定義と検証結果を表 5 に示す．本実験では，3 列目の状態以外のすべての状態が到達できること (P1) も検証した．4 列目の True はそれぞれ性質が満たされていたことを示す．また False の場合，検証結果の反例から性質を満たさない経路 (ビューとアクションの列) を得ることができた．

表 3 縮退要求と要求追跡マトリクス (RTM)

ID	タイプ	縮退要求	要求追跡マトリクス(RTM)				
			機能要求	モデル	ビュー	コントローラ	アクション
1	ABR	ユーザ作成機能が利用不可	新規ユーザを作成する機能	-	-	UsersController.php	add()
2	CBR	ニュース機能全体が利用不可	ニュースの追加, 編集, 削除などの機能を制御	-	-	NewsController.php	-
3	VBR	バグ一覧画面が到達不可	プロジェクトに登録されたバグを管理する画面	-	Issues/index.ctp	-	-
		バグ登録画面が到達不可	プロジェクトにバグを新規登録する画面	-	Issues/new.ctp	-	-
4	MBR	Wiki データにアクセス不可	プロジェクトの Wiki に関連するデータを扱う	Wiki.php	-	-	-

表 4 表 3 の縮退要求を用いた実験結果

ID	関連モデル数	ビュー/エレメント/レイアウトファイル数	アクション数/関連アクション数	遷移元状態数	遷移数	実行時間(分)
1	-	2/1/0	1/2	27	56	6.55m
2	-	5/1/1	7/0	98	119	5.15m
3	-	6/8/1	2/2	98	1126	4.94m
4	2	8/4/1	11/3	98	257	5.51m

表 5 到達可能性の要求と検証

ID	到達不可能状態数	期待する到達不可能状態(-P1)	結果
1	1	Users/add	True
2	4	News/*	True
3	2	Issues/*	False
4	9	Wiki/*	False

初期の縮退計画として、保守担当者は次のように考えると仮定した: ID1) Users/add を描画するアクションは UsersController::add() のみである。ID2) News 関連のビューへの遷移はすべて NewsController が負う責務である。ID3) メニューから直接遷移可能なすべての Issues 関連のビュー (Issues/index および Issues/new) が到達不可能であれば、Issues 関連の全てのビューには到達不可能である。ID4) Wiki のデータを扱うのは WikiController のみであり、到達できなくなるのは Wiki 関連のビューのみである。

上記の縮退計画に則り検証をした結果、ID3 と ID4 が性質違反を引き起こした。ここで得られた反例はそれぞれ、到達不可能であるべき複数の状態が到達可能、到達可能であるべき状態の一つが到達不可能であることを指摘していた。反例を分析した結果、ID3 ではマイページやメニューの検索機能から Issues のビューにアクセスでき、ID4 ではプロジェクトのバージョン情報表示で Wiki データを参照する画面が到達不可能になることが判明した。

デッドロック・到達ステップ: CandyCane では、各状態から初期状態への遷移を可能とするメニューがデッドロックの発生を防いでいた。通常の Web アプリケーションはトップページへのリンクをメニューに持つため、機能縮退が原因でデッドロックが起きないと考えられる。さらに、メニューがナビゲーション構造を浅くした結果、到達不可能になる場合を除き、到達ステップ数は変わらなかった。適切にメニューが実装されているアプリケーションではこれら性質は満たされるが、不適切な場合には、機能縮退による到達性低下を検証するために提案ツールが有効であると考えられる。

以上の通り、提案ツールは機能縮退によるナビゲーション

要求違反を検出できた。保守担当者は、検証結果を元に縮退計画の影響を分析することで、ナビゲーション要求を満たすよう縮退計画を改善できると考えられる。

4.2.2 実用的な実行時間 (RQ2) と有効性

CandyCane は、98 の状態と 5340 の遷移を持つ中規模なアプリケーションであった。なお、この状態数とビューの解析数の差分は DOM 操作に使われたエレメントの数である。表 4 の結果から、提案ツールは中規模程度のアプリケーションに対して約 6 分間の解析時間が必要だったことがわかる。しかし、実行時間の大半は NM の抽出に費やされており、制限と検証の繰り返しはごく短時間で行うことができた。したがって、保守担当者がナビゲーション要求を満たすよう縮退計画を改善し、提案ツールがその改善を検証するためには十分に実用可能である。

表 4 が示すように、従来の縮退プロセスではリンク構造や制限される遷移を把握するために、エレメントの利用箇所やレイアウトの存在、リダイレクトにも注意しなければならない。さらに、モデルの制限では、モデルの関連にも気を配らなければならない。このことから、人手でナビゲーション要求を満たす縮退計画を立てることは困難であるといえる。提案する検証手法では、上記の作業にかかる時間や労力を大幅に削減することができる。

4.2.3 ヘルパー情報設定のコスト

この実験で、我々はアプリ依存のヘルパーメソッドの情報定義を入力した。開発者がメソッドを実装した時にこの情報は定義することで、保守担当者が提案ツールを利用する際に、実装コードを精査して用意する必要はないと考えられる。また、ヘルパーはプレゼンテーション層での頻繁な処理をまとめたモジュールであるため、入力情報の修正の機会が少ないと考えられる。

4.2.4 状態・遷移の見逃しおよび不十分な制限

表 2 の結果では、提案ツールはコントローラ解析中に 2 つのビュー、ビュー解析中に 3 つのエレメント読み込みを見逃した。前者は、変数を用いた文字列操作によってビュー名を決定していたことが原因であった。後者は、分散した

コード片をまたがる複雑なデータフローが原因であった。また、動的なモデル読み込みを解析できていなかった。

ビューやエレメントの指定に変数を使わず、動的なモデル読み込みを実装しない規約を定めることによって、この問題を軽減できると考えられる。CakePHP は規約を尊重するフレームワークであるため、この対策は妥当であると考えられる。規約に従えない場合は、問題となる箇所に絞り込んだ動的解析による RNM の修正が有効である。

4.3 妥当性への脅威

内的妥当性：適用事例では、CandyCane を利用し提案ツールの有用性を示した。これはオープンソースな実アプリケーションであり、内的妥当性への脅威にはならない。しかし、縮退計画は我々が用意したため、内的妥当性への脅威となりうる。したがって、実際の Web アプリケーションの保守現場に提案ツールを導入し、その縮退計画に沿った評価実験の実施が今後の課題となる。

さらに検証では、表 1 で示した到達可能性を検証する有用性のみを示した。今後の課題として、デッドロックや到達容易性低下などの性質に対して、機能縮退による要求違反を検出できるか評価する必要がある。

外的妥当性：提案手法は、MVC デザインパターンの構造に着目し NM の抽出・制限を実現した。したがって、本手法は MVC Web アプリケーション全般に適用できると考えられる。しかし、提案ツールでは CakePHP フレームワークに特化した実装をしたため、Struts*³ といった他のフレームワークで設計された MVC Web アプリケーションへの適用が課題となる。

また、評価実験で利用した CandyCane は中規模程度の実アプリケーションであった。今後の課題として、大規模な Web アプリケーションでも、提案ツールが実用的な時間で解析できるか評価する予定である。

5. 関連研究

Ricca らは Web アプリケーションの状態は Web ページであると主張し、ナビゲーションの静的解析によるテストに成功した [1]。これに対し本研究では、Web ページのテンプレートであるビューを状態とみなしている。

Kubo らは Struts Web アプリケーションの設定ファイルと JSP テンプレートファイルから抽出した NM を SPIN (spinroot.com) を用いて検証した [3]。これは提案ツールと同様に、MVC デザインをベースに設計された Web アプリケーションの実装から NM を抽出している。しかし、縮退運転時の振舞い検証は考慮していない。

Ajax 技術は Web ページ内での状態変化を可能にしたため、Ajax Web アプリケーションのための状態ベーステ

スト・検証手法が提案されている [8], [9], [11]。これらは、クライアント側の振舞いを解析する点で、本手法とは異なる。

6. おわりに

本研究では、縮退実装なく縮退運転時のナビゲーション要求を検証することを目的とし、MVC Web アプリケーションから縮退計画に応じて縮退された NM を抽出・検証するツールを提案した。評価実験では、本ツールが実用的な実行時間で、不適切な縮退運転における要求違反を検出できることを確認した。したがって、提案ツールはナビゲーション要求を考慮した縮退計画の策定を支援できると考えられる。

今後の課題として、反例として出力された実行列が実在することを確認するための動的解析手法との併用が挙げられる。さらには、クライアントサイドの振舞い検証手法との連携が考えられる。また、手法の一般性を評価するために、さらなる適用事例を実施する予定である。

参考文献

- [1] Ricca, F. and Tonella, P.: Analysis and Testing of Web Applications, *Int'l Conf. on Software Engineering*, pp. 25–34 (2001).
- [2] Han, M. and Hofmeister, C.: Modeling and Verification of Adaptive Navigation in Web Applications, *Int'l Conf. on Web Engineering*, pp. 329–336 (2006).
- [3] Kubo, A., Washizaki, H. and Fukazawa, Y.: Automatic Extraction and Verification of Page Transitions in a Web Application, *Asia-Pacific Software Engineering Conf.*, pp. 350–357 (2007).
- [4] 今関雄人, 高田真吾, 土居範久: ソフトウェアの動的モデルに着目したラウンドトリップエンジニアリングの支援, *情報処理学会論文誌*, Vol. 49, No. 7, pp. 47–54 (2007).
- [5] Bennett, T. and Bayrak, C.: Bridging The Data Integration Gap: From Theory to Implementation, *ACM SIGSOFT Software Engineering Notes*, Vol. 36, No. 3, pp. 1–8 (2011).
- [6] Nielsen, J.: User Interface Directions For The Web, *Communications of the ACM*, Vol. 42, No. 1, pp. 66–72 (1999).
- [7] Hollink, V., Someren, M. and Wielinga: Navigation Behavior Models for Link Structure Optimization, *User Modeling and User-Adapted Interaction*, Vol. 17, No. 4, pp. 339–377 (2007).
- [8] Marchetto, A., Tonella, P. and Ricca, F.: State-Based Testing of Ajax Web Applications, *Int'l Conf. on Software Testing, Verification, and Validation*, pp. 121–130 (2008).
- [9] Mesbah, A. and van Deursen, A.: Invariant-Based Automatic Testing of AJAX User Interfaces, *Int'l Conf. on Software Engineering*, pp. 210–220 (2009).
- [10] Ghabi, A. and Egyed, A.: Code Patterns for Automatically Validating Requirements-to-Code Traces, *Int'l Conf. on Automated Software Engineering*, pp. 200–209 (2012).
- [11] Maezawa, Y., Washizaki, H., Tanabe, Y. and Honiden, S.: Automated Verification of Pattern-Based Interaction Invariants in Ajax Applications, *Int'l Conf. on Automated Software Engineering*, pp. 158–168 (2013).

*3 struts.apache.org