1N-7

イベントハンドラを使用した Web アプリケーションを対象とした動作検証

佐藤 隆広† 伊藤 恵‡ 奥野 拓‡

公立はこだて未来大学大学院† 公立はこだて未来大学‡

1. はじめに

近年，画面遷移せずに対話的な操作が可能なリッチクライアントと呼ばれる Web アプリケーションが増加している．リッチクライアントでは，クリックやドラッグ&ドロップなどのイベントをトリガーに，指定された処理を行うイベントハンドラが用いられている．イベントハンドラは JavaScript などで動作する．イベントハンドラが複数定義され，それらが競合した場合，Web アプリケーションが予期しない動作を起こすことがある．しかし，現在そのような Web アプリケーションに対して検証は行われていない．

本研究の目的は，リッチクライアントにおけるイベントハンドラ部分の動作検証を行えるようにすることである．そのために，本研究ではイベントハンドラ部分の動作に着目した検証手法を提案する．そして，その提案手法をイベントハンドラの競合が起こる Web アプリケーションに適用し，提案手法の有効性を確かめる．この提案手法では，Web アプリケーションの動作検証では一般的なモデル検査器 SPIN を用いたモデル検査を行う[1]．しかし，SPIN を用いたモデル検査には仕様記述言語 Promela を記述するための専門的知識を要し，人的コストや時間的コストがかかってしまう．そこで本研究では，その提案手法における Promela コードの記述を自動化する手法も提案する．

2. イベントハンドラの競合

本研究ではイベントハンドラが競合するケースを含む Web アプリケーションを検証対象とする．イベントハンドラの競合は，各イベントハンドラが共有している変数に，複数の処理によって頻繁に値が代入されることが原因である．その結果，仕様とは異なる動作が起こる可能性がある．例えば，一つの UI オブジェクト上で複数のイベントが起こり得る場合，本来独立して動作すべきイベントハンドラが複数同時に動作してしまうことがある．このケースを，図 1 のような旅程編集アプリケーションの例を交えて説明する．図 1 の Web アプリケーションは，ユーザが市街地を移動する際の旅程を決めるために利用される．ユーザは駅，公園などの訪れる施設を施設リストから選択した上で，旅程を編集する．施設リストから選択された施設は旅程編集部分に表示される．旅程の編集はそれらの施設を並び替える，あるいは削除すること

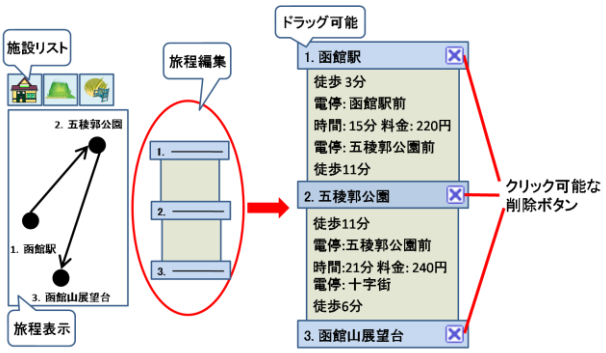

図 1: イベント競合のケース

で行われる．旅程編集部分に表示される施設はそれぞれ一つのコンテナであり，その内部にコンテナを削除するためのボタンが配置されている．この例では，コンテナを並び替える際に行うドラッグを削除ボタン上で行うことで，削除と並び替えのイベントが同時に発生する可能性がある．その際，並び替えが完了する前に，並び替えの対象となっているコンテナが削除される．その場合，並び替えた結果を表示する際に，表示すべきコンテナの一部が未定義となる．これにより，エラーが発生する．本研究では，このようなエラーを検証で検出できるようにする．

3. Web アプリケーションの動作検証

一般に Web アプリケーションが仕様通り正しく動作することを検証する手法としてモデル検査法が用いられている．その際に用いられるモデル検査器として SPIN があり，Web アプリケーションの動作検証を行う際の代表的なツールとして用いられている．そのため，本研究でも SPIN を用いたモデル検査を行う．

モデル検査における従来の手法として，画面遷移に着目した検証手法が提案されている[2][3]．この手法では，検証対象の仕様のモデル化において，Web アプリケーションの各画面を状態遷移図の状態としている．しかし，この手法では Ajax などを用いたリッチクライアントのように，同一ページ内でシステムの内部状態が変化する場合(ページ内遷移)には適用できない．そのため，2 節で挙げたイベントハンドラが競合する Web アプリケーションを検証することができない．

4. イベントハンドラの動作検証手法の構築

本研究では，ページ内遷移する Web アプリケーションに対して検証を行えるようにする．そのために，イベントハンドラの動作仕様に着目して状態遷移図を作成する手法と Promela コードの記述を自動化するための手法

A Verification for Web Applications Using Event Handlers
† Takahiro SATO, Graduate School of Future University Hakodate
‡ Kei ITOU, Future University Hakodate
‡ Taku OKUNO, Future University Hakodate

Copyright ©2013 Information Processing Society of Japan.
All Rights Reserved.

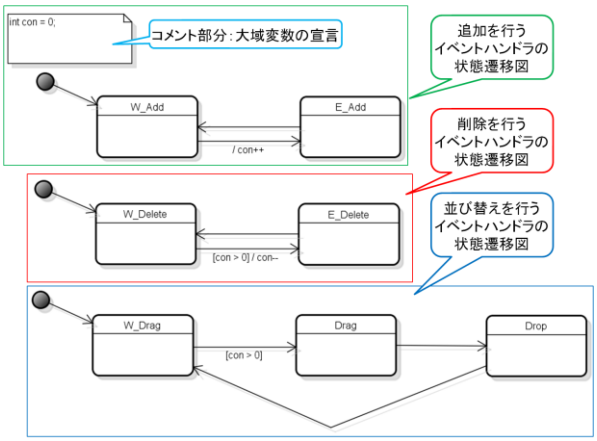

図 2: 検証対象の状態遷移図

の構築を行う．それらの手法について 2 節の旅程編集 Web アプリケーションを検証対象として説明する．

4.1 状態遷移図作成手法

従来手法では，Web アプリケーションの各画面に着目して状態遷移図を作成しているが，本研究で提案する手法ではイベントハンドラの各動作に着目して状態遷移図を作成する．これにより，ページ内遷移する Web アプリケーションの仕様を状態遷移図で表すことが可能となる．イベントハンドラの各動作を状態遷移図で表現する場合，イベントハンドラが取り得る状態と遷移を明確にする必要がある．そのため，本研究では Web アプリケーションの仕様とは別に，イベントハンドラの仕様を作成する．イベントハンドラの仕様には，イベントハンドラの状態と遷移を含む各動作が定義される．イベントハンドラの各状態には，ドラッグ&ドロップを例にすると，待機している状態，ドラッグされている状態，ドロップされている状態がある．このイベントハンドラの仕様を元に検証対象で定義されている各イベントハンドラの動作仕様を状態遷移図で表す．

4.2 Promela コード記述の自動化

Promela コードの記述を自動化する手法は，検証対象の仕様を表す状態遷移図の XML 文書に対して構文解析を行い，Promela コードへ変換するのが一般的である．本研究では，この手法を適用できるようにするための手法を構築する．

イベントハンドラの競合は各イベントハンドラが並列動作するように仕様を記述することで表現できる．それを Promela で記述するためには，イベントハンドラの動作仕様をそれぞれ個別のメソッドとして記述する必要がある．

本研究では，各イベントハンドラの動作仕様を個別の状態遷移図で表す．これにより，イベントハンドラの動作仕様の Promela コードを個別のメソッドとして出力することが可能となる．2 節の検証対象では，施設の選択，削除，並び替えの 3 つのイベントハンドラの動作仕様がそれぞれ個別の状態遷移図として表されることになる(図 2)．また，大域変数を状態遷移図のコメントとして記述することで，変換の際にその部分を大域変数として判別する．2 節の検証対象では，選択した施設の個数を表す変数が大域変数である．以上の手順に従い 2 節の検証対象の仕様の状態遷移図を Promela コードに変換すると図

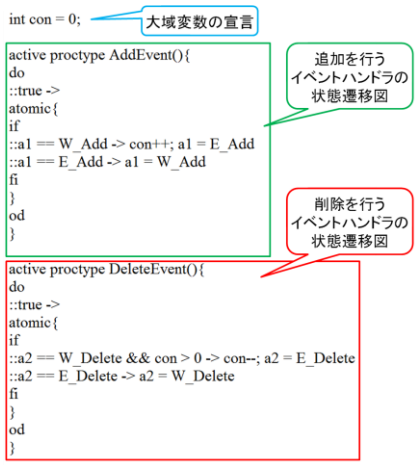

図 3: 出力される Promela コード(一部抜粋)

3 のように出力される．

5. 提案手法の適用実験

4 節で述べた提案手法を適用し，2 節で述べたエラーが発見できることを確かめた．ここでは，検査項目を「施設の並び替えが行われるとき，コンテナの数は変化しない」とした．検証の結果，SPIN は想定通り検証失敗という結果を出力した．このとき SPIN が出力した反例ファイルを調べると，「施設の並び替えが行われているときに，削除を行うイベントハンドラが実行された場合」にエラーが起こることが分かった．すなわち，並び替え中にコンテナの数が減り，検査項目を満たさない場合があることを示せた．したがって，イベントハンドラの競合が原因で起こるエラーを提案手法の適用により検出し，提案手法の有効性を確かめることができた．

6. おわりに

本研究では，イベントハンドラを使用した Web アプリケーションの動作検証手法を提案した．提案手法はモデル検査法に基づいている．イベントハンドラの動作をモデル化することで，リッチクライアントのようなページ内遷移する Web アプリケーションの動作検証を可能とした．また，SPIN を用いたモデル検査法の手順の一つである Promela コードの記述を自動化する手法も構築した．今後は提案手法をより多くのリッチクライアントに適用し，提案手法が適用できる検証対象の規模を評価したい．

参考文献

[1] Spin - Formal Verification, http://spinroot.com/spin/whatispin.html
[2] 馬場敬，結縁祥治，阿草清滋，"Apache Cocoon Flowscript のモデル検査による Web 応用プログラムの動作検証"，電子情報通信学会信学技報，Vol.108, No.444，SS2008-51，pp. 17-22，2009.
[3] 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"，電子情報通信学会信学技報，Vol.109，No.40, SS2009-8，pp. 43-48，2009.

Copyright ©2013 Information Processing Society of Japan.
All Rights Reserved.