

HTML の動的要素に対応した GUI 自動テストツールの提案

橋浦研究室 1201147 戸崎龍ノ介 1201174 平山宏人

1.はじめに

Graphical User Interface (GUI) は、現代のアプリケーションにおいて不可欠な要素である。スマートフォンや Web アプリケーションでは、GUI が仕様を満たさない場合、ユーザが機能を正しく利用できない不都合が生じる。そのため、GUI が仕様通り配置されているかを確認するテストは非常に重要である。過去には GUI テストの自動化を目指した様々なツール[1, 2]が開発されているが、多くは動的要素に対応していないという課題がある。この課題のため、現状の GUI テストは目視で行われており、コストやバグ検知率の不一致が発生している。

2.研究目的

この研究の目的は、自動 Visual Testing ツールに動的要素の検査機能を追加し、自動テストの範囲を拡大することである。具体的には、GUI コンポーネントの描画結果が仕様を満たさないという問題、つまり Presentation Failure[3]を検出するための手法である SpiderTailed[2]を改良する。これにより、自動テストの精度と範囲を拡大させる。

本研究で扱う動的要素は、要素に対してイベントハンドラを起動すると要素が変化するような要素である。具体例として、アコーディオンメニューやタブメニューが挙げられる。これらを実行した際にイベントハンドラが正常に実行されない場合や表示される要素に表示崩れが発生した場合、通常はテスト担当者が手動で確認する必要がある。しかし、本ツールを利用することでイベントハンドラの起動、要素のスクリーンショット取得までをツールが行う。

本ツールを用いることでプログラム変更前後のキャプチャ画像の差分を確認するテストである Visual Testing を自動で行うことができる。

3.提案手法

本研究で検証する 3 つの RQ を以下に示す。

RQ1: 本ツールは動的要素をどれだけテストすることができるか

RQ2: 本ツールが検知できるようになった Presentation Failure はどのようなものか

RQ3: 本ツールの欠陥発見能力はどの程度か

ユーザがテスト対象となる Web ページと正解となる Web ページをプッシュし、それぞれ図 1 に記載の手順でツールが実行されることで二つの Web ページの見た目の差異をレポート

として出力する。ユーザに出力されるレポート(図 2)は HTML 形式であり、各セレクトアの視覚的プロパティやスクリーンショット、HTML コードなどが記載されている。

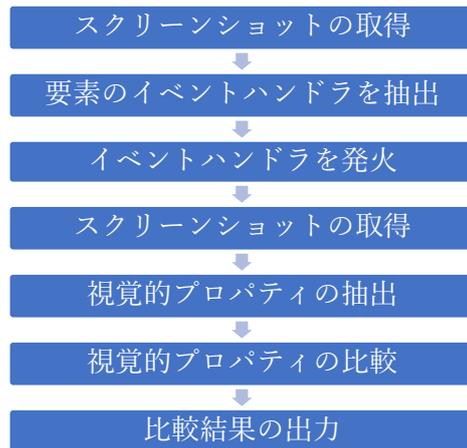


図 1. 本ツールの実行手順

本ツール実行後にスクリーンショットが取得される手順を以下に示す。

1. Web ページが読み込まれた直後の要素すべてのスクリーンショットを取得
2. イベントハンドラの取得
3. 取得したイベントハンドラを 1 つ発火
4. すべての要素をスクリーンショット

Web ページ上に配置されている最後のイベントハンドラが実行されるまで 3, 4 を繰り返す。スクリーンショットを取得後すべての要素の視覚的プロパティの抽出・比較を行う。

Test Result of Each Selector

#	selector	result
0	body	true
1	body noscript	true
2	body .drawerNavObject.drawerNavBody	true
3	body .drawerNavObject.drawerNavBody .drawerNavBody_inner	true
4	body .drawerNavObject.drawerNavBody .drawerNavBody_inner.drawerNavBodySection	true

図 2. ツールが出力するレポート画面

4.評価

日本工業大学の主な就職先一覧の中から 10 社を選び、実際に使用されている Web ページを対象にミューテーション解析を行った。ミューテーション解析とは、ソフトウェアテストの品質を測るための手法の 1 つである。具体的にはテスト対象となるプログラムに対し人為的に誤りを挿入したプログラム (ミュータント) を

生成し、挿入した誤りを検出(キルと呼ぶ)できるかを調べるものである。本実験ではプログラムの誤りを 1 つ挿入した、Web ページを 200 個作成する。作成した Web ページに対して本ツールを用いてテストを行い、本ツールの性能を確認する。本実験で使用するプログラムの誤りを表 1 に示す。

表 1. 本実験で使用するプログラムの誤り

#	ミュータントの種類	誤りの内容
1	width	±30
2	margin	±30
3	padding	±30
4	color	#000000 #FFFFFF #FF0000 #0000FF
5	プロパティ名の削除	
6	イベントハンドラの無効化	

5.結果と考察

本研究で行った実験の結果を RQ ごとに以下にまとめる。

RQ1: 本ツールは動的要素をどれだけテストすることができるか

本実験では動的要素にプログラムの誤りが挿入された Web ページは 36 個であった。36 個の内、本ツールは 14 個の Web ページでミュータントをキルすることができた。

RQ2: 本ツールが検知できるようになった

Presentation Failure はどのようなものか
本ツールでは以下の動的要素をテストすることが可能になった。

1. イベントハンドラを起動すると子要素が表示される要素
2. イベントハンドラを起動すると表示されている要素に変化がある要素

1 はアコーディオンメニューやハンバーガメニュー、2 はスライドショー等が挙げられる。イベントハンドラの無効化やイベントハンドラ起動後の要素の変化、子要素の表示崩れのバグを検知することが可能となった。しかし、イベントハンドラ実行後の挙動はあるが要素に変化がない要素(ページの TOP に戻るボタン等)によるバグやレスポンシブデザインにあたるイベントハンドラのバグ等を検出できなかった

RQ3: 本ツールの欠陥発見能力はどの程度か

サイトごとにミュータントをキルできた割合、ミューテーションスコア (MS) を表 2 に示す。

$$MS = \frac{\text{ミュータントをキルできた Web ページ数}}{\text{作成した Web ページ数}} \times 100$$

表 2. 各サイトのミューテーションスコア

#	サイト名	スコア
1	JR 東日本ビルテック株式会社	90%
2	オリンパス株式会社	50%
3	ドコモ・データコム株式会社	75%
4	ホシザキ北関東株式会社	85%
5	ヤマザキビスケット株式会社	95%
6	株式会社サンノハシ	75%
7	株式会社シミズ・ビルライフケア	70%
8	株式会社ダイキンアプライドシステムズ	60%
9	株式会社東京ビデオセンター	95%
10	新日本建設株式会社	80%

サイトごとのミューテーションスコアを比較した結果、オリンパス株式会社が全体的に低いスコアとなった。スコアが一番低い原因として cookie の承認を確認するポップアップの影響で要素のスクリーンショットを正常に取得することができない場合やイベントハンドラの実行を正常に行うことができないことが挙げられる。

6.まとめと今後の課題

本研究で行った実験の結果から本研究のツールは一部の動的要素に対応することが可能となった。動的要素に対応することにより、先行研究[1, 2]よりも検知できる要素の範囲を僅かながら広げることができた。

今後の課題として、レスポンシブデザインや cookie に対応することで Web ページごとのスコアの差を縮めていき、安定したテストツールの実現があげられる。

参考文献

- [1] Y. Tateishi and U. Inoue, "GUI static testing for object-oriented programming exercises," Proc. of the International Conference on Software Engineering, Artificial Intelligence, Networking parallel/ Distributed Computing (SNPD2019), pp.280-285, Jul. 2019.
- [2] 岡嶋 隆人, 田中 昂文, 樋山 淳雄, 橋浦 弘明, "Web GUI に対するオンラインジャッジシステムプロトタイプの実装," 第 29 回ソフトウェア工学の基礎ワークショップ (FOSE2022) 論文集, pp.217-218, Nov. 2022.
- [3] S. Mahajan and W. G. J. Halfond, "Detection and Localization of HTML Presentation Failures Using Computer Vision-Based Techniques," Proc. of the IEEE 8th International Conference on Software Testing, Verification and Validation (ICST2025), pp.1-10, Apr. 2015.