

画像認識と DOM を使用した Web Visual Testing の定量化手法の提案

岡嶋 隆人[†] 竹田 浩紀[†] 田中 昂文[‡] 樫山 淳雄[#] 橋浦 弘明[†]

日本工業大学[†] 玉川大学[‡] 東京学芸大学[#]

1. はじめに

Visual Testing[1]とは、アプリケーションの GUI コンポーネントのレイアウトが仕様通りに実装されているかを確認するための回帰テストの一種である。Visual Testing は GUI コンポーネントのレイアウト状態を保証し、正常なコード状態でリファクタリングするために行われている。現状、Visual Testing はその特性上、多くの工程を人力に頼っているという問題がある。このため、テストの所要時間や評価が、エンジニアのテスト対象や技術の理解度に依存し、定量的な評価を困難にしている。

2. 研究目的

本研究は前述の問題を踏まえ、Visual Testing の所要時間を予想可能にし、テストの評価結果を均質にすることによって、Visual Testing の効果を定量的に評価することのできる手法を提案する。

3. 関連研究

アプリケーションの GUI コンポーネントのレイアウト仕様を記述する研究については、これまでも様々な手法が提案されている。一例として、Sylvain ら[2]の研究を取り上げる。Sylvain らは仕様記述言語 Cornipickle を開発し、Web アプリケーションの DOM と記述された仕様を比較することによって、GUI の見た目のバグを検知している。

その一方で、この手法は仕様と比較されるのはあくまで Web アプリケーションの DOM であり、実際に Web ブラウザ上にレンダリングされた GUI コンポーネントのレイアウトが仕様通りであることを保証しているわけではないという問題があった。

4. 提案手法

本研究では、画像認識と DOM 解析を使用し、Web アプリケーションのスクリーンショットから、そのアプリケーションの GUI コンポーネントのレイアウトの仕様を生成し、生成された仕様をアプリケーションの修正前後で比較することにより、GUI コンポーネントのレイアウトの変化を検知するツールを開発する。さらに、仕様の総数における充足した仕様の割合や、ツールの所要時間を使用し、エンジニアによる Visual Testing の定量的な評価を可能にする。

5. ツールの実装

ツールは、HTML と CSS で構成された静的な Web アプリケーションを入力として仮定し、Python により実装を行った。提案ツールの概要を図 1 に示す。GUI コンポーネントのレイアウトの仕様の記述には XML (図 2) を用い、実行結果の出力は HTML 形式によるレポートとした。

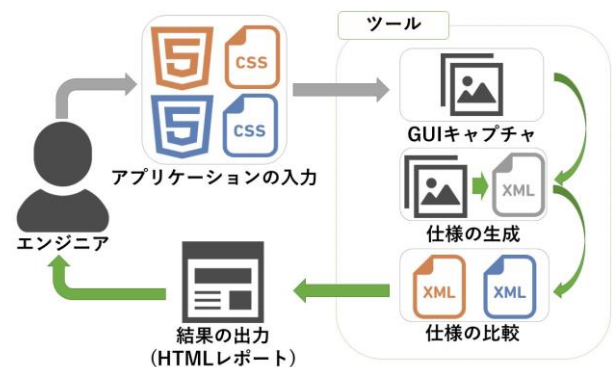


図 1: 提案ツールの概要



図 2: XML による仕様の記述の例

6. 実験

ツールの有効性を確認するために日本工業大学の学生 14 名を被験者とし、人力による Visual

A Proposal of Web Visual Testing Quantification
Technique Using Computer Vision and DOM
Okajima Takato[†] Takeda Koki[†] Tanaka Takafumi[‡]
Hazeyama Atsuo[#] Hashiura Hiroaki[†]
Nippon Institute of Technology[†]
Tamagawa University[‡]
Tokyo Gakugei University[#]

Testing と本提案ツールによる Visual Testing を比較する実験を行った。

実験内容の概要は以下の通りである。まず、HTML と CSS で構成され、LOC が計 844 の静的な Web ページを 2 種類用意する。このうちの一方にはあらかじめ GUI コンポーネントのレイアウトが仕様通りでない箇所を 63 箇所混入させておく。被験者や提案ツールはこれらの 2 つに対して Visual Testing を行い、その結果について比較を行うものである。

被験者が利用するブラウザは Google Chrome とし、発見したバグを記録するための解答用紙を配布した。解答用紙にはあらかじめ仕様通りでない箇所を混入された方のウェブページが印刷されており、被験者はブラウザ上に表示した Web ページを元に、これにペンで印を付けることによって、Visual Testing の結果を記録する。記録された結果から、真陽性(TP)や真陰性(TN)、偽陽性(FP)や偽陰性(FN)について確認を行う。また、提案ツールに対しても同様の評価を行った。

7. 実験結果と考察

実験結果の概要を表 1 に示す。人力については多くの偽陰性(FN)が見られたのに対し、ツール側は多くの偽陽性(FP)が見られたが、偽陰性(FN)数は 0 であった。

さらに、実験結果から検知数(TP+TN)と誤検知数(FP+FN)を求め、これらについて χ^2 検定 ($\alpha=0.05$) を行った結果を表 2 に示す。検定の結果、有意差が認められた。

表 1：人力と提案ツールの検出数の比較

#		真陽性 (TP)	真陰性 (TN)	偽陽性 (FP)	偽陰性 (FN)
1	人力(平均)	59.6	54.9	0.1	3.6
2	人力(標準偏差)	8.61	0.27	0.27	8.61
3	提案ツール	63	20	35	0

表 2：検知数と誤検知数の検定結果

#		検知(TP+TN)	誤検知(FP+FN)
1	人力(総和)	1604	48
2	提案ツール	83	35
3	χ^2 値	176	
4	p 値	2.96×10^{-40}	

実験の所要時間の概要を図 3 に示す。所要時間については、人力は 300 秒から 600 秒超と大きな開きがあるのに対して、ツールは 127 秒と一定時間で結果の出力が可能であった。

人力と提案ツールの Visual Testing 精度の比較を行うために F 値を算出した結果を表 3 に示す。精度では、人力の方が精度が高い結果となった。

その一方で、ツール側は再現率が 1.000 となっており、テスト漏れがないことが確認できた。これらの結果から、提案ツールには現状で、適合率や特異度に問題は残っているものの、テスト漏れがない Visual Testing が可能であることが分かった。

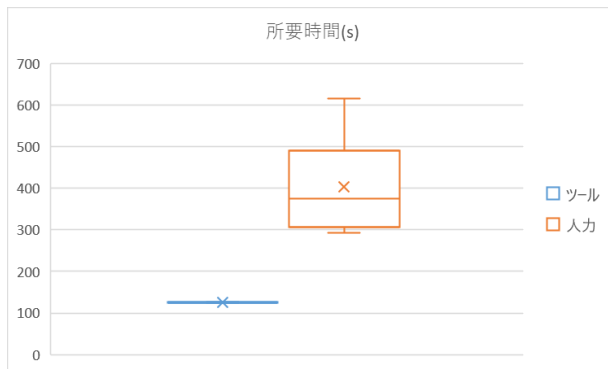


図 3：実験の所要時間

表 3：F 値の比較

#		正解率	適合率	再現率	特異度	F 値
1	人力	0.971	0.999	0.947	0.999	0.972
2	提案ツール	0.703	0.643	1.000	0.364	0.783

8. 結論

本研究は、画像認識と DOM を使用し、Web アプリケーションのスクリーンショットからそのアプリケーションの GUI の見た目の仕様を生成し、バージョン前後で比較することによって GUI コンポーネントのレイアウトの問題を検知する手法を提案した。実験を通じて本手法を使用することによって、人力よりも Visual Testing の所要時間を短縮し、定量的な評価が可能になったことが示唆された。今後は実験で明らかになった精度の低さ等に対しツールの改良を行いたい。

謝辞

本研究の一部は JSPS 科研費 18K11579 の助成を受けた。

参考文献

- [1] A. Issa, J. Sillito and V. Garousi, "Visual Testing of Graphical User Interfaces: An Exploratory Study Towards Systematic Definitions and Approaches," Proc. of 14th IEEE International Symposium on Web Systems Evolution (WSE 2012), pp. 11-15, Oct. 2012.
- [2] H. Sylvain, B. Nicolas, G. Francis, L. B. Gabriel and B. Oussama, "Declarative Layout Constraints for Testing Web Applications," Journal of Logical and Algebraic Methods in Programming, Vol. 85, pp.737-758, Aug. 2016.