

Selenium を用いた Web システムのテスト工程支援

井上 裕太 伊藤 恵

テスト工程は非常に重要な工程である。なぜならテストを行うことで不具合を検出し、それを修正することでソフトウェアの品質を向上させることができるためである。現在のソフトウェア開発はコストの削減や開発期間の短縮が求められており、テスト工程に十分な工数を割くことができず品質を確保できない恐れがある。そこで本研究では、Web システムのテストを対象として、手動テスト時の作業効率化を図るために Web ブラウザテストツールである Selenium [1] を使い、テストデータを用いたテスト作業の支援を行う。具体的には、テスト実行者から与えられた情報をもとに様々な Selenium 用テストケースを生成可能な支援ツールを提供する。生成されたテストケースを実行することによって、テストの作業時間短縮や繰り返しテストの容易化が図れる。

1 はじめに

ソフトウェア開発においてテスト工程は非常に重要である。それは決められた仕様通りに動作するか、プログラムの不具合を検出し修正することでソフトウェアの品質を確保できるため、なくてはならない工程である。また、テストの不備による手戻りは開発工数やエンジニアのモチベーションに影響を及ぼすため、決して疎かにはできない。現在のソフトウェア開発はコストの削減や開発期間の短縮が求められている。ソフトウェア開発におけるテスト工程の割合は開発プロセスの中で大きな割合を占め、ここを効率化することにより開発期間の短縮に繋がるのではないかと考える。

そこで、本研究では Web システムを対象として、手動テスト時の作業効率化を図るために Web ブラウザテストツールである Selenium を使い、テストデータを用いたテスト作業の支援を行う。Selenium はシステムテストや回帰テストを容易に行うことができる。具体的な支援方法は、テスト実行者から与えられた情報を元に様々な Selenium 用テストケースを生成

可能な支援ツールを提供する。生成されたテストケースを実行することによって、テストの作業時間短縮や繰り返しテストの容易化が図れる。

2 関連研究

2.1 ソフトウェアテスト

ソフトウェアテストとは、そのソフトウェアの品質を確保するために、仕様で定義された通りにプログラムが動作するのを確認する作業のことである。ソフトウェアテストを行う最大の目的は、できるだけ多くのバグを発見しそれを修正することである。それはソフトウェアは最初から正常に動作するのは非常に難しく実装工程でのコーディングミスやインタフェースミスなど何らかのバグが必ず混入しているためである。

テストは基本的に、テストレベルとして、単体テスト（ユニットテスト）、結合テスト（統合テスト）、システムテスト、受け入れテストの 4 つのレベルが定義されている [2]。また、バグ修正後に行う回帰テストがある。

● 単体テスト

個々のモジュールでそれぞれのモジュール仕様に対して機能が満たされているか行うテストである。

● 結合テスト

複数のモジュールを組み合わせるモジュール間のイ

A Support for Test Process of Web Systems by Using Selenium

Yuta Inoue, Kei Ito, 公立はこだて未来大学システム情報科学部, Department of Systems Information Science, Future University Hakodate.

ンターフェースがうまく機能するか確かめるテストである。

- システムテスト

システム全体に対して行われるテストで、本番に近い環境で行われ、要求仕様を満たしているかや性能は十分かなどを確かめる。

- 受け入れテスト

発注元の顧客が実際に検証するテストのことである。最終段階のテストであり、ユーザが業務などで実際に使用するデータや操作方法を使って行われる。

- 回帰テスト

バグ修正後にそのバグを検出したテストデータの再実行や、バグ修正後の変更が他の機能に悪影響を与えてないかを確認することである。

2.2 Selenium

Selenium とは、Web ブラウザを使った Web アプリケーションをテストするツールである。JavaScript を用いて、ブラウザ上の操作をテストケースに記述した通りにテストを実施することができる。例えば、ユーザ ID とパスワードを入力ログインボタンを押す、リンクをクリックするなどの操作をテストケースに書いておけば Selenium がその通りに実行してくれる。

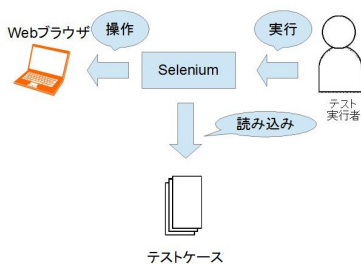


図 1 Selenium でのテスト実行

近年、多くの Web ブラウザや複数のバージョンが混在しており、テスト作業をすべて手動で行うのは非常に時間が掛かり莫大なコストを要するため、困難である。Selenium は複数のブラウザでテストすることが

できる。Selenium 用のテストケースを作成しておけば複数のブラウザでテストを簡単に行うことができるので、テスト作業の効率化を図ることができる。

3 提案

3.1 研究目的

本研究の目的は、Web システムのテストを対象として手動テスト時の作業効率化を図るために Web ブラウザテストツールである Selenium を用いて、テスト作業の支援を行うことである。

Web システムのテストは様々なブラウザやバージョンに対してテストを行い、仕様通りに動作するか確認する必要がある。しかし、それをすべて手動で行うのは時間が掛かり、バグ修正後も繰り返しテストを行わなければならないため効率的ではない。Selenium はブラウザを選ばずにテストすることができるためこれに対して有用であるが、Selenium 用テストケースは手動で作らなければならない。そこで、この部分を生成可能な支援ツールを提供することで、さらなる効率化を図れるのではのかと考える。

3.2 支援ツールの処理手順

3.1 で述べた支援ツールの処理手順を以下のように示す。

1. テスト対象となる Web システムからテストデータを抽出する。
2. テスト実行者からの情報をもとにテストケースを生成する。
3. そこから Selenium 用テストケース群を HTML 形式で生成する。

生成された Selenium 用テストケース群を Selenium で実行することにより繰り返しテストすることができる。ここで、Selenium 用テストケースとは図 3 のような HTML 形式のファイルのことである。

4 実験

4.1 実験で使用する Web システム

実験で使用する Web システムは図 4 のような Java の小テストページを題材とする。小テストの内容は Java の基礎的なもので、クラス名やメソッド名や「イ

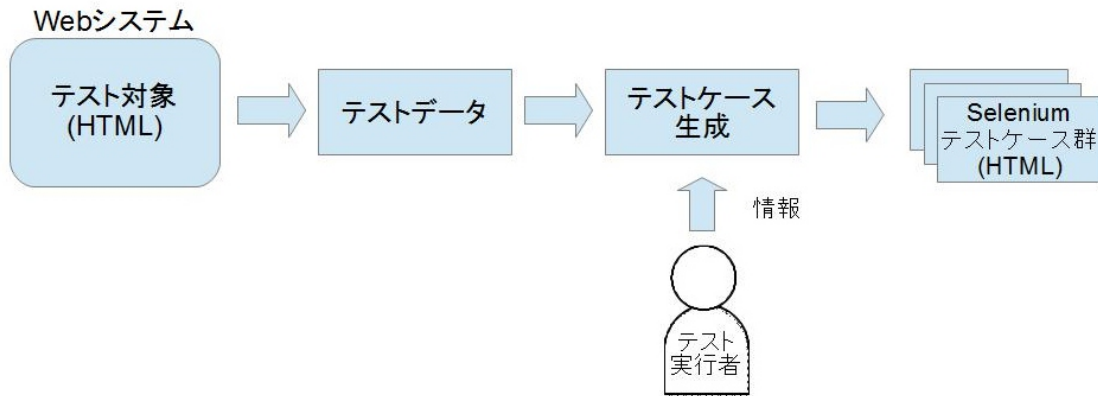


図 2 支援ツール

テストケースを作成した．部分正解にはアルファベットの太文字であるべき箇所が小文字であった場合不正解や，アルファベットの間スペースがあれば不正解となるかを確認するテストケースを作成した．全部不正解には何も入力しなかった場合と明らかに不正解な文字列「aaa」を入力した場合不正解になるか確認するテストケースを作成した．また，13 個のテストケースを Selenium でまとめて実行できるように TestSuite を作成した．TestSuite とは図 5 のようなテストケースをまとめた HTML 形式のファイルのことである．

```

<tr>
  <td>open</td>
  <td>/log/in/index.php</td>
  <td></td>
</tr>
<tr>
  <td>type</td>
  <td>#username</td>
  <td>ct10499</td>
</tr>
<tr>
  <td>type</td>
  <td>#password</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>#css=div.form-input &#t; input[type=submit]</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>#input[&#t;value=編集モードの開始]</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>#css=module=1477 &#t; span.commands &#t; a.editing_update &#t; img.iconsmall</td>
  <td></td>
</tr>
<tr>
  <td>select</td>
  <td>#id=conditionradegroup_0_conditionrademid</td>
  <td>#label=サイズ</td>

```

図 3 Selenium 用テストケース

インスタンス」などのキーワードを答える問題である．具体的な実験内容は，小テストページのフォーム情報から，全部正解，部分正解，全部不正解となるようなテストケースを作り，先の Web システムを Selenium でテストする．そこでバグを見つけ出せるか実験を行う．テストケースは全部正解を 1 個，部分正解を 10 個，全部不正解を 2 個作成した．全部正解には各フォームの回答がすべて正解となるような

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium.openqa.org/profile/testsuite">
<meta http-equiv="Content-type" content="text/html; charset=UTF-8" />
</head>
<title>Test Suite</title>
</head>
<body>
<table id="suiteTable" cellpadding="1" cellspacing="1" border="1" class="selenium"><tbody>
<tr><td>Test Suite</td></tr>
<tr><td><a href="#">TestCaseCorrect01.html</a></td></tr>
<tr><td><a href="#">TestCasePartialCorrect01.html</a></td></tr>
<tr><td><a href="#">TestCasePartialCorrect02.html</a></td></tr>
<tr><td><a href="#">TestCasePartialCorrect03.html</a></td></tr>
<tr><td><a href="#">TestCasePartialCorrect04.html</a></td></tr>
<tr><td><a href="#">TestCasePartialCorrect05.html</a></td></tr>
<tr><td><a href="#">TestCasePartialCorrect06.html</a></td></tr>
<tr><td><a href="#">TestCasePartialCorrect07.html</a></td></tr>
<tr><td><a href="#">TestCasePartialCorrect08.html</a></td></tr>
<tr><td><a href="#">TestCasePartialCorrect09.html</a></td></tr>
<tr><td><a href="#">TestCasePartialCorrect10.html</a></td></tr>
<tr><td><a href="#">TestCaseWrong01.html</a></td></tr>
<tr><td><a href="#">TestCaseWrong02.html</a></td></tr>
</tbody>
</table>
</body>
</html>

```

図 5 TestSuite

1 行
 得点: 8

リスト1およびリスト2に関する以下の記述において、空欄部分に適切な語句を埋めよ。

```

リスト1

public class SimpleVector {

    private double x = 0.0;

    private double y = 0.0;

    public SimpleVector(double x, double y) {

        this.x = x;

        this.y = y;

    }

    public static double norm(SimpleVector v) {

        return ( Math.sqrt(v.x * v.x + v.y * v.y) );

    }

}

リスト2

public class SimpleVectorTest {

    public static void main(String[] args) {

        SimpleVector v1 = new SimpleVector(1.0, 2.0);

        SimpleVector v2 = new SimpleVector(-1.0, -2.0);

        double normdiff = SimpleVector.norm(v1) - SimpleVector.norm(v2);

        System.out.println(normdiff);

    }

}

```

- リスト1で新たに宣言されたクラスは、 クラスであり、メソッドとして を持つ。また、Javaの標準クラスライブラリに含まれている クラスが参照され、そのメソッド が使われている。
- リスト2のSimpleVectorTestクラスにおいて、mainメソッド内の変数v1, v2は クラスの である。変数normdiffは基本型である 型の変数として宣言されており、v1,v2と異なり ではない。

図 4 実験対象の Web システム

4.2 実験結果

今回の実験はまだ支援ツールが未実装で、テストケースの作成は手動で行った。作成にかかった時間は、どういったテストケースがいいのか考えた時間も含め、約 1 時間半であった。作成したテストケースをまとめた TestSuite を Selenium IDE に読み込ませて行った。テストの実行時間は 52 秒であった。実験結果は以下の表 1 のようになった。アルファベットが大文字ではない箇所が小文字やアルファベットの間にスペースがあった場合、不正解にならなければならないところが正解となり、バグを検出することができた。

5 まとめと今後の課題

テスト工程の効率化を図るために Web システムのテストに着目し、Selenium を用いた支援方法の提案

表 1 実験結果

テストの種類	テストケース生成数	テスト成功数	テスト失敗数
小テスト全正解	1	1	0
小テスト部分正解	10	0	10
小テスト全不正解	2	2	0

を行った。この提案する支援方法が有効か一部手作業で予備実験を行い、一定の有効性を確認することができた。今後は提案した支援ツールを実装し、これが有効か Web システムに対して実験を行う。

参考文献

- [1] Selenium, <http://seleniumhq.org/>
- [2] リー・コブラウン, "はじめて学ぶソフトウェアテストの技法", 2006