

A Tool for Impact Analysis of Test Cases Based on Changes of a Web Application

Surasak Phetmanee and Taratip Suwannasart

Abstract—The impact based on changes of a web application affects to the existing test cases which can not be further used to test. This study presents a tool for impact analysis of test cases based on changes of a web application that has abilities to compare differences between two versions of a web application, analyze the effects and also update the test cases as well as additionally generate the new test cases. The result from this tool is the successfully updated test cases that can be used to test the changed web application.

Index Terms—impact analysis, test case, web application

I. INTRODUCTION

WEB application which is recently very important in human daily life, is rapidly developed and changed. The changes unavoidably affect the process of testing because some parts of the existing test cases can not be used to test the changed web application. The traditional solutions of this problem are generation of new test cases to test a new system and validate test cases every time before they are used in web application testing. This solution is also time and resource consuming.

The results from the previous studies about test cases generation [1], [2], web application testing [3], [4], [5], and recent testing tools [6] found that solutions of testing focus only on test cases generation and web application testing. These researchers have not considered changes that can be occurred. Although some researchers studied about impact analysis of changes [7], [8], [9], but it did not include impact analysis of test cases based on changes of a web application. Therefore, we have considered this problem, and our previous research [10] suggested the concept of impact analysis of test cases based on changes of a web application.

This paper presents a tool for impact analysis of test cases based on changes of a web application that can compare differences between two versions of a web application, analyze effects and also update the test cases as well as additionally generate the new test cases.

We have organized the rest of this paper as follows. Section 2 describes a background of the impact analysis of test cases based on changes of a web application. Section 3 presents a tool for impact analysis of test cases. Finally, conclusion and future work are shown in section 4.

II. BACKGROUND

A. Impact Analysis

Impact analysis is a process to identify factors that cause a change. Examples of methodology of impact analysis are using cross reference listings, program slicing, traceability

relationships, configuration management, as well as consulting designs and specifications. [9]

The previous studies [8], [9] have reported that there are studies about impact analysis using several methods, however the researchers did not consider the impact analysis of test cases when a web application is changed. From the definition of impact analysis, we focus on these main issues including:

1. What are changed parts of a web application?
2. How the changes affect test cases?
3. Can we update or generate new test cases to test the changed web application?

B. Test Cases

In this study, the test cases were generated for black-box web application testing including:

1) *Equivalence Class Testing* [11]: Equivalence class testing is a software testing technique to find errors by dividing and partitioning input data to equivalence classes. we use its strategies to divide input data by determining types of values. Constant value, set value, or boolean value are determined as a valid equivalence class and an invalid equivalence class whereas a range value is classified as a valid equivalence and two invalid equivalence classes.

2) *Boundary Value Testing* [11]: Boundary value testing is a software testing technique to find errors using boundary values. The principle of this technique is that errors may appear near the boundary value of input variables. This technique is effective especially for independent ranged variables. This technique determines boundary values of input variables.

C. Web Application

1) *HTML: Hyper Text Markup Language*: In this study, we are interested in HTML form of input variables from users because they are used for sending information on a web application and for testing system. The HTML form consists of textbox fields, password fields, radio, checkbox, select, textarea, and submit buttons. An example of HTML used in this study is shown in figure 1.

2) *XML Schema*: XML Schema is a description of grammatical rules of XML document that supports boundary value definitions and can be used for well-formed input validation. In this study, we use the XML schema concept from these studies [1], [2], [12] to find changes and validate the input data including variable names, data types, variable values, order, and links. Values of XML schema are determined according to standard format and language specification [13]. An example of XML schema used in this study shows in figure 2.

Manuscript received January 6, 2014; revised January 25, 2014.

S. Phetmanee and T. Suwannasart are with the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand e-mail: Surasak.Ph@student.chula.ac.th, Taratip.S@chula.ac.th.

```

<!DOCTYPE html>
<html>
<head><title>Demo</title></head>
<body>
<h2>Demo</h2>
<form name="demo" action="landing.html">
  First name:<input type="text" name="firstname">
  Last name:<input type="text" name="lastname">
  Password:<input type="password" name="pwd"> * require 6-18 characters
  <input type="radio" name="sex" value="male">Male
  <input type="radio" name="sex" value="female">Female
  <input type="checkbox" name="vehicle" value="bike">I have a bike
  <input type="checkbox" name="vehicle" value="car">I have a car
  Where are you from?
  <select name="country">
    <option value="thailand">Thailand</option>
    <option value="united states">United States</option>
    <option value="france">France</option>
    <option value="uk">UK</option>
  </select>
  Comments:<textarea name="comments" rows="4" cols="50">
    Compose new comments...
  </textarea>
  <input type="submit" name="submit" value="Submit">
</form>
</body>
</html>

```

Fig. 1. An example of HTML

```

<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="firstname" type="xs:string"/>
  <xs:element name="lastname" type="xs:string"/>
  <xs:element name="pwd">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:minLength value="6"/>
        <xs:maxLength value="18"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="sex">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="male"/>
        <xs:enumeration value="female"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="vehicle">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="bike"/>
        <xs:enumeration value="car"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>

```

Fig. 2. An example of XML Schema

III. A TOOL FOR IMPACT ANALYSIS OF TEST CASES

This study presents a tool for impact analysis of test cases based on changes of a web application shown in figure 3.

Figure 3 shows the overview of the tool. The process of this tool is divided into three main procedures including the comparison of the difference between two versions of document files, impact analysis and test case improvement, as well as new test cases generation. The result from this tool provides the information of the changed parts and how the parts on a web application that are changed. In addition, completely updated test cases could be used to test the changed web application. The details of this process and methodology are described as the following topics.

A. Two Versions of the Document File Comparison

A user can import a new version of document file to compare with the original version from the database as follows.

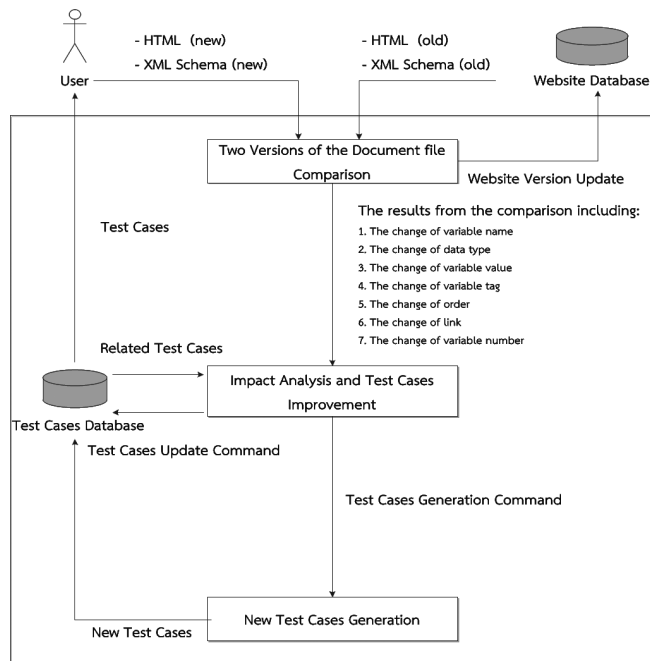


Fig. 3. The Overview of the tool for impact analysis of test cases

1) *HTML document file*: which is according to standard format and language specification. After the document file is imported, then it is analyzed to find HTML tags and variable names. The interesting tags are textbox fields, password fields, radio, checkbox, select, textarea, and submit buttons.

2) *XML Schema file*: which is according to standard format and language specification. After the document file is imported, then it is analyzed to find variable names, data types, variable values, orders, links, and variable numbers. The variable names must be identical with the name of variables in the HTML file. In this study, the tool supports several data types including string, integer, positiveInteger, boolean, decimal, and dateTime. A valid variable value can be constant value, range value, set value, and boolean value. Then, the user can import new version of document file. This version is used to compare with the original version from the database.

The methodology of the document file comparison is that the values from two versions of HTML files and XML Schema files are compared to find the differences. The result obtained from this tool consists of several changed values. Our previous study suggests 7 patterns of the change of a web application as follows.

1. The change of variable name
2. The change of data type
3. The change of variable value
4. The change of variable tag
5. The change of order
6. The change of link
7. The change of variable number

The change might be add, update, or delete of variables. From this procedure, the user obtain the information of the changed parts and how the parts on a web application that are changed. The user interface of this concept is shown in figure 4.

Upload HTML & XML Schema



Fig. 4. User interface for two versions of the document file comparison

B. Impact Analysis and Test Case Improvement

This procedure consists of original test cases searching, impact analysis, and test cases improvement as shown in figure 5.

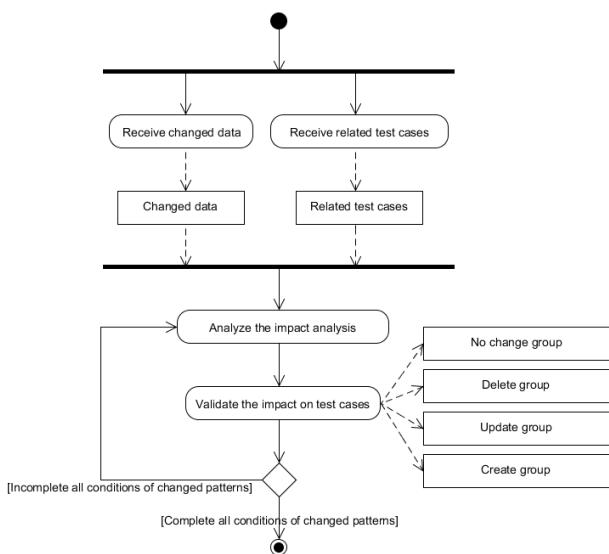


Fig. 5. Impact analysis and test case improvement procedure

Firstly changed data and related test cases are imported. Then, these data are analyzed and validated the impact on test cases. The test cases are classified into 4 groups including no change group, delete group, update group, and create group. This method supports several patterns of orchestral changes by analyzing and validating the impact until finishing all test cases. The patterns of changes and impact on test cases are presented in our prior study showing how the changes affect to existing test cases.

Updated test cases in this procedure are update group and delete group. Then the tool updates the test cases as new test cases according to the changes of new version of the web application. For equivalence class testing, this tool updates the original test cases as new test cases based on equivalence class data whereas the boundary value testing updates the original test cases as new test cases based on boundary value analysis data. An example of the updated of test cases using equivalence class testing is shown in figure 6.

From the example in figure 6, the updated test cases using equivalence class testing is shown. The pwd variable is updated according to the equivalence class. After processing, pwd variable of test cases are updated as the changes of new version of web application as shown in figure 7, and the user interface of this method is shown in figure 8.

Project 1: index.html

Equivalence class

Variable	Class no.	Input equivalence class	Class type	Value
firstname	1	Not specific	valid	xyz
lastname	1	Not specific	valid	abcdeffg
pwd	1	6 <= length <= 18	valid	123456789
	2	length < 6	invalid	12345
3	3	length > 18	invalid	1234567890123456789
	sex	1	sex in {male, female}	valid
2	2	sex not in {male, female}	invalid	other
	vehicle	1	vehicle in {bike, car}	valid
2	2	vehicle not in {bike, car}	invalid	other
	country	1	country in {thailand, united states, france, uk}	valid
2	2	country not in {thailand, united states, france, uk}	invalid	other
	comments	1	default is Compose new comments...	valid
2	2		invalid	other
	submit	1	link is landing.html	valid
2	2		invalid	e2d2c558114.html

Fig. 6. An example of equivalence class

ID	Variable Name	Value	Sequence	Output
1	firstname	xyz	1	valid
	lastname	abcdeffg	2	
	pwd	123456789	3	
	sex	male	4	
	vehicle	bike	5	
	country	thailand	6	
	comments	Compose new comments...	7	
	submit	landing.html	8	
2	firstname	xyz	1	invalid
	lastname	abcdeffg	2	
	pwd	12345	3	
	sex	male	4	
	vehicle	bike	5	
	country	thailand	6	
	comments	Compose new comments...	7	
	submit	landing.html	8	
3	firstname	xyz	1	invalid
	lastname	abcdeffg	2	
	pwd	1234567890123456789	3	
	sex	male	4	
	vehicle	bike	5	
	country	thailand	6	
	comments	Compose new comments...	7	
	submit	landing.html	8	

Fig. 7. The new version of test cases

However, some patterns of changes are the necessary to be used in additional new test cases generation to cover the web application testing that we mention earlier.

C. New Test Case Generation

New test cases are generated using either equivalence class or boundary value analysis. In this study we apply the method of new test cases generation from [10]. Two methods are described as the following. All of generated new test cases are shown in figure 9.

1) *Test case generation using equivalence class:* Test case generation starts with the generation of equivalence classes of imported data using correct variable values. After that each equivalence class of imported data is randomized. The random value then is used to generate test cases followed by these alternative methods including weak normal equivalence class testing, strong normal equivalence class testing, weak robust equivalence class testing, and strong robust equivalence class testing.

Project 1: index.html

Impact analysis & test cases improvement

Equivalence class test cases		
#	Impact classification	Total
1	No change	64
2	Delete	0
3	Update	32
4	Create	0

Boundary value test cases		
#	Impact classification	Total
1	No change	4
2	Delete	0
3	Update	3
4	Create	0

Do you want to update test cases?

Fig. 8. User interface for impact analysis and test case improvement

2) *Test case generation using boundary value:* Test case generation starts with the analysis of boundary value of variables using correct variable values including min-, min+, nom, max-, max, and max+. Then these values are generated as test cases followed by the alternative method of test case generation including boundary value analysis, worst case testing, robustness testing, robust worst-case testing.

Project 1: index.html

Test cases

Finally, you can get test cases that you want to use it for testing the `index.html` page from here.

Equivalence Class Testing		Boundary Value Testing	
Weak Normal	1	Boundary Value Analysis	5
Strong Normal	1	Worst-Case Testing	5
Weak Robust	8	Robustness Testing	7
Strong Robust	96	Robust Worst-Case Testing	7

Fig. 9. User interface for new test cases generation

REFERENCES

- [1] S. Munpeansuk, *A Tool for Generating Test Cases from HTML Document and XML Schema*. Master Thesis, 2003.
- [2] S. Supapon, *Test Case Generation for Web Application using Black-box Testing Technique*. Master Thesis, 2004.
- [3] J. F. M. Benedikt and P. Godefroid, "Veriweb: Automatically testing dynamic web sites," *World Wide Web Conference Series*, 2002.
- [4] H. M. Sneed, "Testing a web application," *Website Evolution*, 2004.
- [5] A. K. A. Bruns and D. Wichmann, "Web application tests with selenium," *IEEE Software*, vol. 26, pp. 88–91, 2009.
- [6] J. Huggins, "Selenium - web browser automation [online]," Available: <http://www.seleniumhq.org>, 5 Dec 2013.
- [7] S. A. Bohner, "Software change impacts - an evolving perspective," *International Conference on Software Maintenance*, pp. 263–272, 2002.
- [8] S. Lehnert, "A taxonomy for software change impact analysis," *Proceedings of the 12th International Workshop on Principles of Software Evolution and the 7th annual ERCIM Workshop on Software Evolution*, p. 4150, 2011.
- [9] R. S. Arnold and S. A. Bohner, "Impact analysis - towards a framework for comparison," *International Conference on Software Maintenance*, p. 292301, 1993.
- [10] S. Phetmanee and T. Suwannasart, "Impact analysis of test cases based on changes of a web application," *International Computer Science and Engineering Conference: ICSEC*, vol. 2, pp. 577–582, 2013.
- [11] P. C. Jorgensen, *Software Testing: A Craftsmans Approach*, 3rd ed. United States of America: CRC Press LLC, 2007.
- [12] Y. Bassil and M. Alwani, "Automatic html interface generator for web applications," *International Journal of Web and Semantic Technology (IJWesT)*, vol. 3, 2012.
- [13] W3C, "W3c xml schema specifications [online]," Available: <http://www.w3.org/XML/Schema.html>, 5 Dec 2013.

IV. CONCLUSION AND FUTURE WORK

The objective of this study is to present a tool for impact analysis of test cases based on changes of a web application. The result from this study is updated test cases that can be used to test the changed web application. The new solution solves the old problem that the user has to generate all of test cases and the benefit of this solution is the reduction of time and non necessary resource consuming.

For future work, we will use this tool for impact analysis with changed web applications that have 7 patterns of the change such as the change of variable name, data type, variable value, variable tag, order, link, and variable number, then evaluate by comparing the reduction of time and resource with traditional methods of test case generation.