AUTOMATIC TEST CASE GENERATION FROM UML ACTIVITY DIAGRAM USING ACTIVITY PATH

Yasir Dawood Salman Almulham

UNIVERSITI UTARA MALAYSIA 2010

AUTOMATIC TEST CASE GENERATION FROM UML ACTIVITY DIAGRAM USING ACTIVITY PATH

A project submitted to Dean of Postgraduate Studies and Research Office in

partial fulfillment of the requirement for the degree

Master of Science

Information Technology (IT)

Universiti Utara Malaysia

By

Yasir Dawood Salman Almulham

KOLEJ SASTERA DAN SAINS
(College of Arts and Sciences)
Universiti Utara Malaysia

PERAKUAN KERJA KERTAS PROJEK
(Certificate of Project Paper)

Saya, yang bertandatangan, memperakukan bahawa
(I, the undersigned, certifies that)

YASIR DAWOOD SALMAN ALMULHAM
(802400)

calon untuk Ijazah
(candidate for the degree of) **MSc. (Information Technology)**

telah mengemukakan kertas projek yang bertajuk
(has presented his/her project of the following title)

AUTOMATIC TEST CASE GENERATION FROM ACTIVITY DIAGRAM USING ACTIVITY PATH

seperti yang tercatat di muka surat tajuk dan kulit kertas projek
(as it appears on the title page and front cover of project)

bahawa kertas projek tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan.
(that this project is in acceptable form and content, and that a satisfactory knowledge of the field is covered by the project).

Nama Penyelia
(Name of Supervisor) : **DR. NOR LAILY HASHIM**

Tandatangan
(Signature) : _____ Tarikh (Date) : 10/10/10

Nama Penilai
(Name of Evaluator) : **MDM. MAWARNY MD. REJAB**

Tandatangan
(Signature) : _____ Tarikh (Date) : 10/10/10

PERMISSION TO USE

In presenting this project in partial fulfillment of the requirements for a postgraduate degree from the Universiti Utara Malaysia, I agree that the University Library may make it freely available for inspection. I further agree that permission for copying of this project in any manner in whole or in part, for scholarly purposes may be granted by my supervisor or in their absence by the Dean of the Postgraduate Studies and Research. It is understood that any copying or publication or use of this project or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my project.

Requests for permission to copy or to make other use of materials in this project, in whole or in part, should be addressed to

Dean of Postgraduate Studies and Research

College of Arts and Sciences

Universiti Utara Malaysia

06010 UUM Sintok

Kedah Darul Aman

Malaysia

Abstract

The most important part of the testing attempt is the test case generation. As a modeling language, Unified Modeling Language (UML) is the most generally used to describe design specifications and analysis by both academic and industry. Therefore, UML becomes the sources of Test Case Generation. Test cases are usually generated from the requirement, and UML activity diagram illustrates the sequential control flows of activities what make it possible to generate test cases for activity diagrams. This research proposes an approach to automatically generate test cases directly from UML activity diagram using activity graph. Therefore, this research will create an algorithm and implement it on a prototype using the UML activity diagram as an input to generate the test cases. The result of all these test cases will be compared to the result that has been generated manually to evaluate the usability and reliability of the proposed algorithm.

ACKNOWLEDGEMENTS

First, I would like to express my appreciation to Allah, the most merciful and, the most compassionate, who has granted me the ability and willing to start and complete this study. I do pray to His Greatness to inspire and enable me to continue the work for the benefits of humanity.

After that, my most profound thankfulness goes to my supervisor Dr. Nor Laily Hashim for her scientifically proven and creativity encouraging guidance and great support in this study.

Last, I wish to thank my Father, Mother who were always there for me by giving everything they have, my brothers and sisters for their love and support.

Thank you UUM.

Yasir Dawood Salman

September 19, 2010

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1 Introduction

The new paradigm to develop the software is to use a Model-driven approach (France et al, 2006). Among the advantages behind it is the increasing of efficiency with its supporting in many domains like solution, development, and business problems. In the development of the model-driven software, Unified Modelling Language (UML) has become the industry standard for object oriented software development, also UML diagrams are effective enough to hold most of functioning phase (Usman & Nadeem, 2009).

One of the important challenges in software testing is the test cases generation. It is especially complicated when a system contains simultaneously executing participants, since a system like that can show different responses depending on the simultaneous occurrence conditions. A UML activity diagram is a suitable modelling language for describing interactions between system objects given that an activity diagram can be conveniently used to capture business processes, workflows and interaction scenarios (Kim et al, 2007).

There are many programs and applications with different purposes and types which have been used everywhere. Test case generation is one of the most important elements for the testing efforts for programs and applications (Linzhang et al, 2004).

The contents of the thesis is for internal user only

REFERENCES

Alalfi, M., Cordy, J., & Dean, T. (2009). Automated Reverse Engineering of UML Sequence Diagrams for Dynamic Web Applications. *IEEE Computer Society*, 287-294.

Albert, E., Gomez-Zamalloa, M., & Puebla, G. (2010). PET: A Partial Evaluation-based Test Case Generation Tool for Java Bytecode. *ACM*, 25-28.

Alshammari, S. A. (2010). Generating Test Cases for LearningZone. *Thesis*, UUM .

Anneliese, A., Jeff, O., & Roger, A. (2005). Testing Web Applications by Modeling with FSMs. *Software Systems and Modeling*, 1-28.

Bourhfir, C., Aboulhamid, E., Dssouli, R., & Rico, N. (2001). A test case generation approach for conformance testing of SDL systems. *ScienceDirect Computer Communications*, 24 (2-4), 319-333.

Boutekkouk, F., Benmohammed, M., Bilavarn, S., & Auguin, M. (2009). UML for Modelling and Performance Estimation of Embedded Systems. *Journal Of Object Technology*, 8 (2), 95-118.

Canevet, C., Gilmore, S., Hillston, J., Kloul, L., & Stevens, P. (2004). Analysing UML 2.0 activity diagrams in the software performance engineering process. *ACM*, 74-78.

Cartaxo, E., Neto, F., & Machado, P. (2007). Test Case Generation by means of UML Sequence Diagrams and Labeled Transition Systems. *IEEE*, 1292-1297.

Chen, M., Qiu, X., Xu, W., Wang, L., Zhao, J., & Li, X. (2009). UML Activity Diagram-Based Automatic Test Case Generation For Java Programs. *The Computer Journal*, 52 (5), 545-556 .

Filippo, R., & Paolo, T. (2001). Analysis and Testing of Web Applications. *IEEE*, 25-34.

Fisher, M., Cao, M., Rothermel, G., Cook, C. R., & Burnett, M. M. (2002). Automated Test Case Generation for Spreadsheets. *ACM*, 141-153 .

France, R., Ghosh, S., Dinh-Trong, T., & Solberg, A. (2006). Model-driven development using UML 2.0: promises and pitfalls. *IEEE*, 39 (2), 59 - 66.

Fraser, G., & Wotawa, F. (2007). Test-Case Generation and Coverage Analysis for Nondeterministic Systems Using Model-Checkers. *IEEE*, 45-50.

Heumann, J. (2001). *Generating Test Cases From Use Cases.* Retrieved 5 July, 2010, from: http://www.ibm.com/developerworks/rational/library/content/RationalEdge/jun01/GeneratingTestCasesFromUseCasesJune01.pdf

Hoffmann, V., Lichter, H., Nyßen, A., & Walter, A. (2009). Towards the Integration of UML- and textual Use Case Modeling. *Journal of Object Technology*, 8 (3), 85-100.

Javed, A., Strooper, P., & Watson, G. (2007). Automated Generation of Test Cases Using Model-Driven Architecture. *IEEE*, 3-9.

Kaner, C. (2003). *What Is a Good Test Case?* Retrieved July 8, 2010, from: www.kaner.com/pdfs/GoodTest.pdf

Kang, M., Wang, L., & Taguchi, K. (2004). *Modelling Mobile Agent Applications in UML2.0 Activity Diagrams.* Retrieved 14 August, 2010, from: www.auml.org/auml/supplements/UML2-AD.pdf

Kansomkeat, S., & Rivepiboon, W. (2003). Automated-Generating Test Case Using UML Statechart Diagrams. *South African Institute for Computer Scientists and Information Technologists*, 47 (1), 296–300.

Kim, H., Kang, S., Baik, J., & Ko, I. (2007). Test Cases Generation from UML Activity Diagrams. *IEEE*, 556-561.

Kundu, D., & Samanta, D. (2009). A Novel Approach to Generate Test Cases from UML Activity Diagrams. *Journal of Object Technology*, 8 (3), 65-83.

Kuo, F. (2009). An indepth study of mirror adaptive random testing. *IEEE*, 51-58.

Li, X., Cui, M., Pei, Y., Zhao, J., & Zheng, G. (2001). Timing Analysis of UML Activity Diagrams. *Springer-Verlag*, 62-75.

Lilly, R., & G, U. (2010). Reliable Mining of Automatically Generated Test Cases from Software Requirements Specification. *International Journal of Computer Science Issues*, 87-91.

Linzhang, W., Jiesong, Y., Xiaofeng, Y., Jun, H., Xuandong, L., & Guoliang, Z. (2004). Generating Test Cases from UML Activity Diagram based on Gray-Box Method. *IEEE*, 284-291.

Ma, C., Wu, J., Zhang, T., Zhang, Y., & Cai, X. (2008). Automatic Test Case Generation for BPEL Using Stream X-Machine. *International Journal of u- and e- Service*, 27-36.

Manar, A., James, C., & Thomas, D. (2009). Automated Reverse Engineering of UML Sequence Diagrams for Dynamic Web Applications. *IEEE*, 287-294.

Mingsong, C., Xiaokang, Q., & Xuandong, L. (2006). Automatic Test Case Generation for UML Activity Diagrams. *ACM*, 2 - 8 .

Niere, J., Wadsack, J., & Zündorf, A. (2003). Recovering UML Diagrams from Java Code using Patterns. *In Proceedings of the 2nd workshop on Soft Computing Applied to Software*, Enschede, The Netherlands.

Pfaller, C., & Pister, M. (2008). Combining Structural and Functional Test Case Generation. *Software Engineering*, 229-241.

Raamesh, L., & Uma, G. V. (2010). Reliable Mining of Automatically Generated Test Cases from Software Requirements Specification. *International Journal of Computer Science*, 87-91.

Riebisch, M., Philippow, I., & Götze, M. (2002). UML-Based Statistical Test Case Generation. *Springer-Verlag*, 394-411.

Robert, A., Maksimchuk, R., Engle, M., Young, B., Conallen, J., & Houston, K. (2007). *Object-Oriented Analysis and Design with Applications.* New York: Random House.

Royce, W. (1970). Managing the Development of Large Software Systems. *IEEE*, 328-338.

Ryser, J., & Glinz, M. (2005). Using Dependency Charts to Improve Scenario-Based Testing. *Elsevier Science*, 85-97.

Seifert, D. (2008). Test Case Generation from UML State Machines. *Scientific Commons*, 1-11.

Sujana, J., Claire, M., & John, K. (2007). An Interaction Visualisation Tool for a Learning Management System. *ACM*, 326-331.

Tsai, W. T., Wei, X., Chen, Y., Paul, R., & Xiao, B. (2005). Swiss Cheese Test Case Generation for Web Services Testing. *Oxford University Press*, 2691-2698.

Usman, M., & Nadeem, A. (2009). Automatic Generation of Java Code from UML Diagrams using UJECTOR. *International Journal of Software Engineering and Its Applications*, 21-38.

Vaishnavi, V., & Kuechler, W. (20 January , 2004). *Design Research in Information Systems.* Retrieved 16 August , 2009, from: http://desrist.org/design-research-in-information-systems

Yan, J., Li, Z., Yuan, Y., Sun, W., & Zhang, J. (2006). BPEL4WS Unit Testing: Test Case Generation Using a Concurrent Path Analysis Approach. *IEEE*,6 (1), 75-84.

Yongyan, Z., Jiong, Z., & Paul, K. (2007). An Automatic Test Case Generation Framework for Web Services. *Journal of Software*, 64-77.

Yuan, X., & Memon, A. M. (2010). Generating Event Sequence-Based Test Cases Using GUI Run-Time State Feedback. *IEEE*,36 (1), 1-16.