Automatic test case generation using collaboration UML diagrams

Article *in* World Journal of Science Technology and Sustainable Development · February 2013

Sayyed Usman Ahmed
Aligarh Muslim University

World Journal of Science and Technology 2012, 2(x):xx-xx
ISSN: 2231 – 2587
Available Online: www.worldjournalofscience.com

Automatic test case generation using collaboration UML diagrams

Syed Usman Ahmed[1] , Sneha Anil Sahare[1] and Alfia Ahmed[2]

[1]Assistant Professor, Department of Information Technology, Jodhpur Institute of Engineering and Technology, Jodhpur, Rajsthan - 342005, India
[2]Department of Computer Science, Lachoo Memorial College of Science and Technology, Jodhpur, Rajasthan – 342005, India

Abstract
Software Testing have changed its virtue from mere validation and verification to necessity of information and technology industry. As testing a software work product is time consuming task and it involves reliability issues, it must be done very carefully. This paper presents an approach to generate automate test cases so that the reliability and efficiency related to software testing increases. In this paper we have proposed to use collaboration UML diagram for the Online Exam System and generate collaboration Weighted graph (CWG) with the help of a parser written in C# language. This CWG will then be traversed with the help of Prism's and Dijkstra's algorithm for generating automatic test cases. These test cases so generated will cover the branch, statement and decision coverage cyclomatic complexity.

Key words: software testing, test cases, automated, UML-based testing, UML diagrams, collaboration diagram

INTRODUCTION

Software testing has become a part and parcel of today's industry almost everything we use today has an element of software in it [1]. Quality of software work product depends on the amount and quality of testing being done software reliability, scalability and performance are some of the factors that are very much valued by the customer. An ideal **test suite** is one which has minimal test cases that can test both the functional and Non-functional requirements of software work product.

In order to generate test case we generally acquire the manual mode and write the test cases in accordance to the specification provided by the customers. However this mode suffers from the drawback of time and accuracy. The present IT industry is switching from manual mode to automated test case generation to overcome these drawbacks. Now we do have a huge mass of testers that do not have basic coding knowledge to perform white box testing and execute all branches, statements and code level coverage. This paper presents some of the work done in this field to use Unified Modelling Language (UML) diagrams that was used till now to write the specification of users or customers, to generate test case data and optimize the same for improving efficiency and code coverage.

Collaboration diagrams are being created in the design phase of any software work product. Our methodology uses the same so that the inconsistency and ambiguity in design specification can be handled in a better manner. Our proposed system is being centred on the concept of Unified Modelling Language (UML). UML [2, 3] is the most widely used methodology for the development of object-oriented systems. It uses various diagrams to create the structure of a software work product, in our case we have used collaboration diagram as the basis of our study. Collaboration diagram depicts message passing between different objects involved in any system implementation. We have proposed to generate automatic test cases from UML collaboration with the help of Rational® Software Architect from Rational Software Corporation.

Literature Survey

Automated test cases can be generated based on the following approaches:

- Scenario based test case generation
- Model based test case generation
- Genetic based test case generation

These approaches are used in various researches at undergraduate and post graduate level. Scenario is a set of events which is studied in form of case studies. Generally scenario based test case generation focuses on concurrent approach with concurrence coverage criteria. In comparison scenario based approach which works on case studies, model based techniques uses UML diagrams - *use-case diagram, class diagram, activity diagram, communication (or collaboration) diagram, sequence diagram, state chart diagram*, to identity the test cases. Model based technique converts the UML diagrams mentioned above into control graphs using various techniques, like Java parser or XMI notation,

Received: Jan 03, 2013; Revised Jan 28, 2013 Accepted: Jan 30, 2013

*Corresponding Author
Syed Usman Ahmed
Assistant Professor Jodhpur Institute of Engineering and Technology
Jodhpur, Rajasthan - 342005, India

Email: Syedusman.ahmed@jietjodhpur.com

and thus create set of test cases. Genetic base test case generation involves the usage of genetic algorithms to generate optimized test cases which then can used be used in the field of date mining and warehousing.

Researcher working on automated test case generation techniques uses certain tools to generate UML diagrams, tools which are generally used are Rational® Rose, Rational® Software Architect etc

Related work

As this field of software testing is becoming more and more promising day by day, many researchers and practitioners are working in this field. Linzhang et al. [2] has work on UML activity diagram and they generated automatic test cases using grey box testing techniques and a prototype called UMLTGF. Santosh et al. [3] proposed a system to generate automatic test cases based on sequence and activity diagram combined together. Their work covers three important faults message sequence faults, operation consistency faults, and activity synchronization faults. They implemented their system using Model Flow Graph (MFG) approach. Sharma et al. [4] used use case and sequence diagrams to generate automatic test cases. They have used Use Case Diagram Graph (UDG) and Sequence Diagram Graph (SDG) and uprooted three important defects: use case initialization faults, use case dependency faults, and operational faults. M. Prasanna et al. [5] has worked on collaboration diagram and converted the collaboration diagram into intermediate graph using Java parser and applied Prism's and Dijkstra's to transverse the graph structure. They performed mutation analysis on the buy ticket system and achieved 88% yield by finding 81 % unit faults and 87% integration faults in comparison to Aynur et al. [6] approach which has 77% unit faults and 83% integration faults

Proposed Method

The system that we are proposing will work as mentioned below:

1. Elaborate a problem statement for Online Exam System.
2. Generate a Collaboration Diagram (CD) using Rational® Software Architect software.
3. Convert the CD generates in step 2 into XML/XMI format and store it as .mdl file.
4. A parser will be developed in C#.Net framework v3.5 that will read the .mdl file and generate a graph shown in fig 3 G= (V, E), where G represents the graph, V represents the set of nodes (Objects) and E represents the set of connecting edges (Messages).
5. Sequence number for each edge (Message) is used as weight of the edge between two nodes.
6. Using Prism's [7] algorithm find the node with minimum weighted edge, and using Dijkstra's [7] concept transverse the node with minimum weighted edge and create test cases.
7. The complete test sequence for the application will be obtained when maximum-weighted edge is reached.

The test case so produced will cover the branch, statement and decision coverage. The test cases will be optimized using the mutation testing method described by M. Prasanna [5]. The percentage yield achieved in [5] is 81 % unit faults and 87% integration faults.

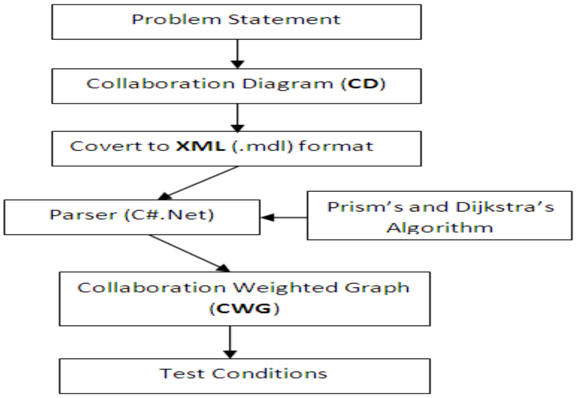
Problem Statement
Collaboration Diagram (CD)
Covert to XML (.mdl) format
Parser (C#.Net)
Prism's and Dijkstra's Algorithm
Collaboration Weighted Graph (CWG)
Test Conditions

Fig 1 Flow chart for the proposed system

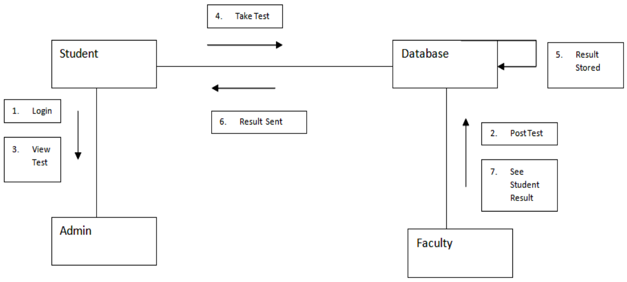

Fig 2 Collaboration diagram for Online Exam System

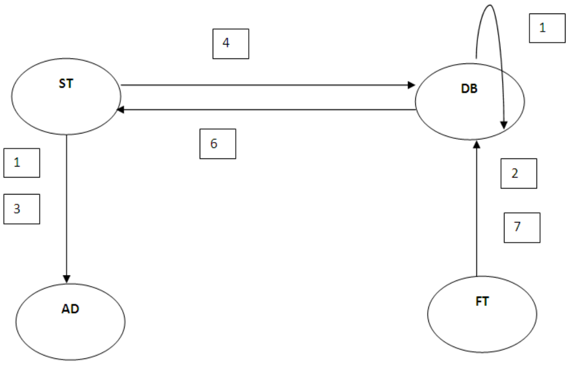

Fig 3 CWG for the collaboration diagram of Online Exam System

Case Study

Online Exam System is a system that is very widely being used in assessing and analysing the educational abilities of students across the globe. This system basically has four basic elements: Student, Database, Administrator and Faculty in-charge.

Fig 2 represents the complete system inform of collaboration diagram. The faculty post the test paper in to the database, then the student logins the system with his specific username and password.

On successful login a student can view the test available for him. Next he takes one of the test for which he is eligible.

Once the test is successfully completed the result is being stored in the database and the same is send to the student. The system also gives an option to the faculty that he can view the score of the student and analyse the structure of the paper and competency level of the student.

Conclusion

The proposed system will be study the work flow of Online Exam System and will generate automatic test cases based on UML collaboration diagram shown in Fig 2. The Collaboration Weighted Graph (CWG) shown in Fig 3 will be transverse using Prism's and Djakarta's algorithm [7].

The percentage yield of the proposed approach will be compared with Aynur et al. [6] approach and M. Prasanna [5] approach, so that we can conclude the proposed system performance with standard research done in the field of software automated test case generation.

REFERENCES

[1] Software Testing: Principles and Practice, 1/e By Srinivasan Desikan& Gopalaswamy Ramesh, Published: 2008 ISBN 978-81-7758-121-8, pp 4-6

[2] M. Priestley, "Practical object-oriented design with UML", Tata McGraw Hill Press, 2005.

[3] A. Bahrami, "Object oriented systems development", McGraw-Hill Press, 1999.

[4] W. Linzhang, Y. Jiesong et al.: "Generating Test Cases from UML Activity Diagram based on Gray-Box Method". In Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC04), pages 284-291. IEEE, 2004.

[5] Santosh Kumar Swain, Durga Prasad Mohapatra, "Test case generation from Behavioral UML Models", International Journal of Computer Applications (0975-8887), Volume 6-No. 8, September 2010

[6] M Sharma, R Mall, "Automatic test case generation from UML models", International Conference on Information Technology, pp. 196-201, Dec 2007.

[7] M. Prasanna, K. R.. Chandran, K. Thiruvenkadam, "Automatic test case generation for UML collaboration diagrams", IETE Journal of research, vol. 57, Issue 1, Jan – Feb 2011

[8] A Abdurazik, J Offutt, A Baldini, "A controlled experimental evaluation of test cases generated from UML diagrams", Technical Report, George Mason University, ISE-TE-04-03, 2004.

[9] S Sahni, "Data Structures, algorithms and applications in C++", McGraw Hill Press, 2006.

[10] A. Abdurazik, J. Offutt, "Using UML collaboration diagrams for static checking and test generation", In 3rd International Conference on the UML, pp. 383-395, 2000.

[11] J. Offutt, A. Abdurazik: *"Generating tests from UML specifications",* In Proceedings of 2nd International Conference on the UML, pp. 416-429,

[12] S. K. Swain. UML-based Testing of Software System, Technical Report, KIIT, 2005.

[13] Philip Samuel, Sunitha E. V., "Document Type Definition for the XMI Representation of UML 2.0 Activity diagram", International Journal of Recent Trends in Engineering, vol. 1, No. 1, 2009

[14] Manoj Kumar, Arun Sharma, Rajesh Kumar, "Towards Multi-Faceted Test Cases Optimization", Journal of Software Engineering and Applications, Issue 4, pp 550 – 557, 2011

[15] A. V. K. Shanthi, G. Mohan Kumar, "Automated Test Case Generation from UML Sequence Diagram", International Conference on Software and Computer Applications (ICSCA 2012), IPCSIT vol. 41, 2012

[16] Mingsong chen, Xiaokang Qiu, Wei Xu, Linzhang Wang, Jianhua Zhao, Xuandong Li,"UML Activity Diagram-Based Automatic Test Case Generation For Java Program", The Computer Journal, 2007

[17] P Samuel, R Mall, and A K Bothra, "Automatic test case generation using UML State diagrams", IET Software, pp 79-93, 2008

[18] A. V. K. Shanthi, D. Parthiban, G. Mohan Kumar, "A survey of UML-Base Automatic Test Cases Generation for Software Testing"

[19] Shanthi, A. V. K., and G. MohanKumar. "A Novel Approach for Automated Test Path Generation using TABU Search Algorithm." International Journal of Computer Applications 48.13 (2012): 28-34.

[20] Preeti Gulia, R. S. Chillar, "A new approach to generate and optimize test cases for UML state diagram using genetic algorithm", ACM SIGSOFT Software Engineering Notes archive, Volume 37 Issue 3, May 2012 Pages 1-5.