

{tag} International Journal of Computer Applications
Foundation of Computer Science (FCS), NY, USA

[Volume 176](#)

-
[Number 7](#)

Year of Publication: 2017

Authors:

Afrina Khatun, Naushin Nower, Kazi Sakib

10.5120/ijca2017915633

{bibtex}2017915633.bib{/bibtex}

Abstract

Automated test regeneration intends to ensure high coverage of system model from an existing test suite. While regenerating test suite, most of the existing techniques ignore coverage achieved by existing test suite. As a result, these techniques leave important model elements untested. Thus, an automatic test regeneration technique to achieve high state model coverage is proposed. In the proposed technique, Input Parser module processes inputted UML diagrams, source code and test suite as XML elements, source class and test steps respectively. The Coverage Computational module measures model coverage result by executing the existing test suite. Finally, Test Regeneration module regenerates executable test cases considering coverage result, UML and source information. The experimental results on four projects show that the proposed technique improves transition and state coverage of existing test suite on average by 61.26% and 52.95% respectively. Moreover, the technique has also successfully regenerated 98% executable test cases.

References

1. Shahid, M., Ibrahim, S. "An evaluation of test coverage tools in software testing." in Proc. of the 6th International Conference on Telecommunication Technology and Applications (CSIT), Vol. 5, pp.217–222, 2011.
2. Utting, M., Legeard, B. "Practical Model-Based testing A tools Approach", Morgan Kaufmann Publishers, 2007
3. Fraser, G., Arcuri, A. "Evosuite: automatic test suite generation for object-oriented software", in Proc. of the 19th ACM SIGSOFT Symposium and the 13th European conference on Foundations of Software Engineering, pp.416–419, ACM, 2011
4. Nahar, N., Sakib, K. "SSTF: A novel automated test generation framework using software semantics and syntax", in Proc. of the 17th International Conference on Computer and Information Technology (ICCIT), pp.69–74, IEEE, 2014
5. Najumudheen, E. S. F., Mall, R., Samanta, D. "A dependence graph-based test coverage analysis technique for objectoriented programs", in Proc. of the 6th International Conference on Information Technology: New Generations (ITNG), pp.763–768, IEEE, 2009
6. Heckeler, P., Behrend, J., Kropf, T., Ruf, J., Rosenstiel, W., Weiss, R. "State-based coverage analysis and UML-driven equivalence checking for C++ state machines" FM+ AM, Vol. P-179, pp. 49-62, 2010
7. Ferreira, R. D., Faria, J. P., Paiva, A. C. "Test coverage analysis of UML state machines", in Proc. of the 3rd International Conference on Software Testing, Verification, and Validation Workshops (ICSTW), pp.284–289, IEEE, 2010
8. He, W., Zhao, R. "Sequential Pattern Mining Based Test Case Regeneration", in IEEE Journal of Software, Vol. 8, no. 12, pp.3105–3113, 2013.
9. Alshahwan, N., Harman, M. "State aware test case regeneration for improving web application test suite coverage and fault detection", in Proc. of the International Symposium on Software Testing and Analysis (ISSTA), pp.1–5, IEEE, 2012.
10. Khatun, A., Sakib, K. "An automatic test suite regeneration technique ensuring state model coverage using UML diagrams and source syntax" in Proc. of the 5th International Conference on Informatics, Electronics and Vision (ICIEV), pp.88- 93, IEEE, 2016.
11. Sarma, M., Mall, R. "Automatic test case generation from UML models", in Proc. of the 10th International Conference on Information Technology,(ICIT), pp.196–201, IEEE, 2007.
12. Mingsong, C., Xiaokang, Q., Xuandong, L. "Automatic test case generation for UML activity diagrams", in Proc. of the International Workshop on Automation of Software Test (AST), pp.2–8, ACM, 2006
13. Eclipse.org - Juno Simultaneous Release, <https://eclipse.org/juno/>, Online accessed: 19 December 2016
14. JEP - Java Math Expression Parser <http://www.cse.msu.edu/SENS/Software/jep-2.23/doc/website/index.html> Online accessed: 3 September 2017.
15. javaparser 1.0.9 - Maven Repository <http://mvnrepository.com/artifact/com.google.code.javaparser/javaparser/1.0.9>, Online accessed: 19 May, 2016.
16. Enterprise Architect, <http://www.sparxsystems.com/>, Online accessed: 25 August, 2016.
17. JUnit - About, <http://junit.org/> Online accessed: 18 December, 2016.
18. Experimental Projects <https://github.com/Afrina/Projects> Online accessed: 5 September, 2016.
19. Observer Pattern Object Oriented Design <http://www.>

oodesign.com/observer-pattern.html Online accessed: 19 December, 2016.

Index Terms

Computer Science

Information Sciences

Keywords

Software Testing; Automatic Test Regeneration; Model Coverage Analysis; Unit Testing; Integration Testing