

Automated JUnit Test Generation with EvoSuite

Seminar Proposal

José Miguel Rojas
Department of Computer Science
The University of Sheffield, Sheffield, UK

1 Title

Automated JUnit Test Generation with EvoSuite.

1.1 Lecturer

José Miguel Rojas (j.rojas@sheffield.ac.uk)
Department of Computer Science, The University of Sheffield, Sheffield, UK

1.2 Coordinator

Prof. Sira Vegas (svegas@fi.upm.es)

2 Summary

The EVOSUITE tool automatically generates JUnit tests for Java software. Given a class under test, EVOSUITE creates sequences of calls that maximise testing criteria such as line and branch coverage, while at the same time generating JUnit assertions to capture the current behaviour of the class under test. This is an interactive hands-on seminar where participants learn the basic notions of software unit testing and how to use EVOSUITE for automated JUnit test generation during software development. The key message is that automated unit test generation with EVOSUITE is ready for use during real software development. We will cover basic usage on the different EVOSUITE options (e.g., different coverage criteria, seeding) and interfaces (command line, Eclipse, IntelliJ and Maven plugins). EVOSUITE is open source; we will also outline ways in which students can contribute in research projects involving EVOSUITE.

2.1 Intended Audience

The main target audience for this seminar are:

- Students using Java for software development.
- Students interested in learning about software unit testing.
- Students interested in automation of unit test generation techniques.
- Students interested in automated software engineering research.

2.2 Dates and Duration

15–18 March; 16 hours (4 hours per day).

2.3 Outline

- Software testing
- Unit test generation
- The JUnit test automation framework for Java
- Unit test generation tools for Java
- EVOSUITE [1–4]
 - Search-based testing
 - Mutation-based assertion generation
 - Main options
 - The command-line interface
 - The Eclipse and IntelliJ plugins
 - The Maven plugin
 - Open lines of research

3 Methodology

The seminar will start with motivation to software unit testing and its automation. The JUnit framework will be explored as the ad-hoc standard for unit testing of Java software. Participants will be presented with a pre-defined small Java project with seeded faults and will be asked to write JUnit tests with assertions that detect those faults.

An overview will follow on the state of the art of automated unit test generation techniques and tools for Java. EVOSUITE will then be introduced and the main underlying techniques will be explained. Participants will be familiarised with the basic workflow of EVOSUITE. They will be asked to download

an archive that contains some example classes and the EVOSUITE jar file. They will then be shown how to run EVOSUITE on the example classes, what are the main parameters to consider and how to set them (e.g., search budget, search algorithm, minimisation, assertion generation), and what is the output.

Once participants have familiarised themselves with EVOSUITE, we will demonstrate how one would use the tool as a software developer. Participants will work again on the previously given a small Java project, this time using the Apache Maven build infrastructure and the EVOSUITE's Maven plugin. They will be asked to use Eclipse or IntelliJ IDEA to perform a small implementation task, and use EVOSUITE while doing so.

To conclude, we will provide an overview on how students can contribute to improve and extend EVOSUITE. We will start by checking out the source code from GitHub (<https://github.com/EvoSuite/evosuite>), and explaining how to build EVOSUITE. We will then walk the participants through the main modules. To demonstrate how to extend EVOSUITE we will let participants do a small exercise where they implement a small modification to the underlying genetic algorithm.

4 Assessment Method

Assessment will be based on classroom attendance, classroom participation and the solution of programming and testing exercises.

5 Prerequisites

Knowledge of object-oriented programming, basic programming skills in Java, and familiarity with the Eclipse IDE.

6 Language

English.

7 Short Biography

José Miguel Rojas is a Research Associate in Software Testing at The University of Sheffield, United Kingdom. He received a Computer Science Degree from the Universidad Autónoma Gabriel René Moreno (Santa Cruz, Bolivia 2007) and a PhD in Computer Science from the Technical University of Madrid (Madrid, Spain 2013). His research interests include automated software engineering and software testing. His research is mainly conducted through the development of tools and techniques and through empirical methods such as controlled experiments with human participants.

References

- [1] G. Fraser and A. Arcuri, “EvoSuite: Automatic test suite generation for object-oriented software.” in *ACM Symposium on the Foundations of Software Engineering (FSE)*, 2011, pp. 416–419.
- [2] —, “EvoSuite: On the challenges of test case generation in the real world (tool paper),” in *IEEE Int. Conference on Software Testing, Verification and Validation (ICST)*, 2013.
- [3] —, “Whole test suite generation,” *IEEE Transactions on Software Engineering*, vol. 39, no. 2, pp. 276–291, 2013.
- [4] —, “1600 faults in 100 projects: Automatically finding faults while achieving high coverage with evosuite,” *Empirical Software Engineering (EMSE)*, 2013.
- [5] —, “A large-scale evaluation of automated unit test generation using evosuite,” *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 24, no. 2, p. 8, 2014.
- [6] J. M. Rojas, G. Fraser, and A. Arcuri, “Automated unit test generation during software development: A controlled experiment and think-aloud observations,” in *ACM Int. Symposium on Software Testing and Analysis (ISSTA)*, 2015.
- [7] G. Fraser, M. Staats, P. McMinn, A. Arcuri, and F. Padberg, “Does automated unit test generation really help software testers? a controlled empirical study,” *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 2015.