Paper Title: Wodel-Test: a model-based framework for language-independent mutation testing

Paper doi: https://doi.org/10.1007/s10270-020-00827-0

Team members: Diaconu Ana-Maria, Duma Amalia-Diana, Drăghiciu Diana

Mutation testing (MT) is a software testing technique consisting in changing specific components of an application's source code to ensure that a software test suite will be able to detect the changes. This process is based on the hypothesis that if a test suite can distinguish between a program and its mutants, then it may be used to identify software flaws. Building a MT tool requires a significant effort which may limit the usage of this testing technique in emerging programming or domain-specific languages with a low number of users where, without automation, building the tool may not be cost-effective. Another problem arises when MT tool needs to handle heterogeneous technology which leads to further complexity in development. To tackle these problems, this paper describes a language independent approach for the automated construction of MT tools. For a specific language we use model-driven engineering (MDE) to generate a customized MT tool and rely on domain specific languages to define mutation operators over the model-representation of programs. The MDE approach distinguishes between two roles: the MT tool creator and the tester. The former provides a MT tool specification from which a MT tool is synthesized. The latter uses the generated tool to perform MT.

The aim of Wodel-Test is to provide a language-independent framework for mutation testing that can be used with different programming languages and software models. This allows developers to test their software more thoroughly and efficiently, regardless of what language or modeling tool they are using. The novelty of the tool consists in the chosen approach for the automated construction of MT tools, mainly, its focus on model-based mutation testing. This means that, rather than the code itself, software models are used to represent the behaviour of the tested software, model-driven engineering(MDE) being used to generate a custom MT tool for the language. The main advantages of this specific approach include the fact that models can be used to generate a large number of test cases quickly and efficiently and they are often easier to understand and analyze than the source code, this being especially useful for complex software systems. Wodel-Test has an additional innovative feature, that being its ability to handle multiple programming languages by using a domain-specific language (DSL) which allows developers to specify the software models and mutation operators in a language-independent way. Thus, it is easier for developers to use the toll with different programming language without having to learn a new testing framework for each language beforehand.

The evaluation was conducted from the perspective of the two user roles involved in the process: the tester and the MT tool creator. To assess the usefulness of the generated MT tools, the functionality of the MT tool for Java, WODEL - TEST/Java, developed using the approach provided in this research is compared with other existing Java MT tools. This comparison is divided into three parts: general features of the tool, extensibility and customizability of mutation operators and efficiency of the MT process. Regarding the general features, the most important criteria that was examined was: input to the MT process, mutant generation process and reporting. In terms of extensibility, WODEL - TEST/Java allows extending the set of predefined mutation operators by providing domain specific languages that enable the creation of arbitrary mutation operators. A MT process was executed on a Java library created by a third party to determine the efficiency (how many mutants per second can WODEL - TEST/Java produce) and efficacy (are the generated mutants appropriate for evaluating the test cases). The goal was to evaluate to what extend this approach was able to produce results similar to other MT tools. According to the results of this trial, WODEL - TEST/Java was the fastest in the per-mutant time category.