Master Thesis Topic

Grammar-based Fuzzing for Model-Driven Software Engineering (MDSE) Tools

Motivation and Background

Fuzzing for fuzz testing [1] is an established technique that aims to discover unexpected program behavior (e.g., bugs, security vulnerabilities, or crashes) by feeding automatically generated data into a program under test. However, the application of fuzzing to test Model-Driven Software Engineering (MDSE) tools is still limited due to the difficulty of existing fuzzers to provide syntactically and semantically valid input instance models that conform to a given meta-model. That is, although input models are serialized in XMI, the internal structure must form an abstract syntax graph (ASG) that must at least be properly typed and adhere to additional validity constraints. Otherwise, the system under test may reject the input and as a result, the core functionality of the program remains untested. While grammar-based fuzzing techniques [2,3,4] are able to efficiently produce syntactically valid XML files, it is unclear if they can be leveraged to effectively produce input files that reproduce the internal structure of valid instance models. In particular, it is unknown how well grammar-based approaches perform against existing MDSE fuzzing tools, e.g., MoFuzz [5].

Goals

The goal of this thesis is to explore grammar-based fuzzing techniques to fuzz MDSE tools and compare against existing MDSE fuzzing approaches.

Description of the Task

The specific tasks are:

- Getting familiar with grammar-based fuzzing techniques and the Eclipse Modeling Framework (EMF) [6]
- Develop a method to derive a grammar from a given EMF meta-model and integrate it into an existing grammar-based fuzzer
- Perform an experimental evaluation of the implemented approach and compare against existing MDSE fuzzing tools

Research Type

Theoretical Aspects:	*****
Industrial Relevance:	*****
Implementation	*****

Prerequisite

The student should be enrolled in the master of computer science program, and has completed the required course modules to start a master thesis.

Skills required

Programming skills in Java, understanding of, or willingness to learn, the software engineering methods (like fuzz testing) and tools (e.g., the Eclipse Modeling Framework) needed for the project.

Contacts

Hoang Lam Nguyen (nguyehoa@informatik.hu-berlin.de)
Software Engineering Group, Institut für Informatik, Humboldt-Universität zu Berlin

References

[1] Manès, Valentin Jean Marie, et al. "The art, science, and engineering of fuzzing: A survey." IEEE Transactions on Software Engineering (2019).
[2] Godefroid, Patrice, Adam Kiezun, and Michael Y. Levin. "Grammar-based whitebox fuzzing." Proceedings of the 29th ACM SIGPLAN Conference on Programming Language Design and Implementation. 2008.

[3] Soremekun, Ezekiel, et al. “Inputs from Hell: Learning Input Distributions for Grammar-Based Test Generation.” IEEE Transactions on Software Engineering (2020).
[4] Aschermann, Cornelius, et al. "NAUTILUS: Fishing for Deep Bugs with Grammars." NDSS. 2019.
[5] Nguyen, Hoang Lam, Nebras Nassar, Timo Kehrer, and Lars Grunske. “MoFuzz: A Fuzzer Suite for Testing Model-Driven Software Engineering Tools”. 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE). 2020.
[6] Eclipse Foundation. 2020. Eclipse Modeling Framework. https://www.eclipse.org/modeling/emf/