Automatic Benchmark Generation for OCL

Ankit Jha
Department of Computer Science
Maynooth University, Ireland
Supervisor: Hao Wu & Rosemary Monahan
Ankit.Jha.2021@mumail.ie

Abstract

The Unified Modeling Language (UML) is widely used in building different types of software models. It is the central element for Model Driven Engineering (MDE) software development methodology. The graphical notations of UML not only allow software engineers to model different aspects of software but also enhances the collaboration among team members. On other hand, the Object Constraint Language (OCL), is a declarative language that enables software engineers to use formal methods to specify rules or constraints that cannot be expressed by using UML graphical notations. Hence, using both UML and OCL to build software models makes current software development practices more rigorous.OCL as a specification language offers many features that allow users to write formal rules for a UML model. For example, using pre/post conditions to specify rules of an operation call in a UML class diagram. Recently, numerous approaches and techniques have been proposed for analyzing or verifying UML models annotated with OCL.These approaches and tools either provide some level of case studies or (semi-)automatic tools support for verifying OCL constraints. However, a large research gap still remains. That is no comprehensive OCL benchmarks exist for evaluating OCL analysis tools and verifiers. This immediately imposes three challenges on the UML and OCL communities. (1) It is infeasible to gain an overview of how better existing tools and techniques perform in verifying OCL constraints. (2) It is difficult to compare individual tools for others to identify the strength and limitations of each technique. (3) It is time-consuming to design a customized benchmark for testing a particular aspect of an OCL tool such as string data type-based verification. To close this gap and tackle these challenges, we propose an approach to automatically generate OCL benchmarks. In particular, our proposed research focuses on the following two research questions.

- **RQ1:** How to efficiently generate a large set of OCL benchmarks.
- **RQ2:** How to effectively evaluate generated OCL benchmarks.