

Initiating a Benchmark for UML and OCL Analysis Tools

Martin Gogolla¹, Fabian Büttner^{2,*}, and Jordi Cabot²

¹ University of Bremen, Germany

² AtlanMod, École des Mines de Nantes - INRIA, LINA, France

Abstract. The Object Constraint Language (OCL) is becoming more and more popular for model-based engineering, in particular for the development of models and model transformations. OCL is supported by a variety of analysis tools having different scopes, aims and technological corner stones. The spectrum ranges from treating issues concerning formal proof techniques to testing approaches, from validation to verification, and from logic programming and rewriting to SAT-based technologies. This paper is a first step towards a well-founded benchmark for assessing validation and verification techniques on UML and OCL models. The paper puts forward a set of UML and OCL models together with particular questions for these models roughly characterized by the notions consistency, independence, consequences, and reachability. The paper sketches how these questions are handled by two OCL tools, USE and EMFtoCSP. The claim of the paper is not to present a complete benchmark right now. The paper is intended to initiate the development of further UML and OCL models and accompanying questions within the UML and OCL community. The OCL community is invited to check the presented UML and OCL models with their approaches and tools and to contribute further models and questions which emphasize the possibilities offered by their own tools.

1 Introduction

Model-driven engineering (MDE) as a paradigm for software development is gaining more and more importance. Models and model transformations are central notions in modeling languages like UML, SysML, or EMF and transformation languages like QVT or ATL. In these approaches, the Object Constraint Language (OCL) can be employed for expressing constraints and operations, thus OCL plays a central role in MDE. A variety of OCL tools is currently available, but it is an open issue how to compare these tools and how to support developers in choosing the OCL tool appropriate for their project. This paper puts forward a set of UML and OCL models together with particular questions for these models. This set of models is intended to be a first version of an OCL analysis tool benchmark to be developed within the OCL and UML community.

* This research was partially funded by the Nouvelles Équipes program of the Pays de la Loire region (France).

The current benchmark consists of four UML and OCL models: CivilStatus (CS), WritesReviews (WR), DisjointSubclasses (DS), and ObjectsAsIntegers (OAI). These models employ and emphasize different UML and OCL language features and pose different computational challenges for the analysis tools and their underlying technologies like provers, solvers, or finders: Plain invariants and enumerations in CS, association multiplicities in WR, classifier generalization in DS, and recursive operation definitions with inherited association ends and constraints in OAI. The accompanying questions can be roughly characterized by the partly overlapping notions consistency, independence, consequences, and reachability: under the label ‘consistency’ we discuss whether there exist object diagrams for the model at all, ‘independence’ concentrates on whether the invariants are non-redundant, ‘consequences’ studies how to formally deduce new properties from the explicitly stated ones, and ‘reachability’ focuses on how to characterize all object diagrams of a model and how to construct an object diagram with stated properties. The benchmark does not expect that all questions can be fully answered by a considered tool, but it expects that it is discussed to what extent and in which direction an approach or tool can help to answer the question.

The structure of the rest of this paper is as follows. The next section gives a short introduction to OCL. Section 3 introduces the first version of our benchmark. Four example models with accompanying questions are introduced. As a proof of concept for the applicability of the models, Sect. 4 and Sect. 5 show how these models and questions are handled by two concrete tools and how the models must be fine-tuned to become processable by the respective tool, if needed. These two tools have been selected to illustrate how the models can be used to evaluate tools. Section 6 puts forward a list of topics that could be addressed in future work. Section 7 discusses related work and some (not all) approaches suitable to be subject to an OCL analysis tool benchmark. The paper is finished in Sect. 8 with concluding remarks. Furthermore, the paper is extended by an additional document [14] in which all models are detailed in the formats `.use` and `.ecore` and all details of the benchmark examples for the tools USE and EMFtoCSP are made available.

2 OCL in 5 Minutes

The Object Constrains Language (OCL) is a textual, descriptive expression language. OCL is side effect free and is mainly used for phrasing constraints and queries in object-oriented models. Most OCL expressions rely on a class model which is expressed in a graphical modeling language like UML, MOF or EMF. The central concepts in OCL are objects, object navigation, collections, collection operations and boolean-valued expressions, i.e., formulas. Let us consider these concepts in connection with the object diagram in Fig. 1 which belongs to the class diagram in Fig. 3. This class diagram captures part of the submission and reviewing process of conference papers. A more detailed description of the class diagram and the corresponding constraints is given later in Section 2.2. The class diagram defines classes with attributes (and operations, not used in