Abstract

Model-driven engineering (MDE) is a software development methodology that focuses on use of models in software development process. Later on, these models can be transformed into code in order to save time and efforts of the developers.

One of the widely used models for transformation is Unified Modeling Language (UML) class diagram along with its Object Constraint Language (OCL) constrains. Before transformation of UML/OCL model into code, it is a significant concern to detect and trace the errors in the model, as model transformations and code generation may spread errors to other notations where they are harder to detect and trace.

The formal verification of UML models is a time consuming process and there are several formal verification tools which can check the correctness of UML/OCL models, but their high computational complexity is a common drawback. It means that the current verification methods have problems with scalability for complex UML models. If we try to apply the existing methods on larger models, the formal verification tools and techniques take longer time to verify the model.

To address these problems, we propose disjoint and non-disjoint slicing techniques that will break the UML/OCL class diagram into several independent submodels in order to reduce the complexity of UML/OCL class diagram. Afterwards, these submodels can be verified separately through any verification engine. The disjoint slicing technique is very conservative in several ways, for example it only considers disjoint slices. However, the non-disjoint slicing is based on a more aggressive slicing criterion that can still preserve the satisfiability in case of non-disjoint slices.

Furthermore, a novel feedback technique is proposed which highlights any unsatisfiable submodels with their integrity constraints from the complex hierarchy of UML/OCL class diagram. The developers can benefit by correcting those unsatisfiable submodels while ignoring the rest of the model. The slicing procedures and feedback technique are implemented in UMLtoCSP(UOST). Several complex UML/OCL class diagrams are verified in UMLtoCSP(UOST) and their verification time is computed.