Optimization algorithms for constraint handling

Gergely Mezei, Tihamér Levendovszky, and Hassan Charaf

Model-based development is one of the most focused research fields. Domain-specific languages can describe the target domain in a flexible and highly customizable way, but they require a proper domain specification: the abstract syntax. Model-based development also requires efficient model transformation techniques to transform the models to source code, or to another model according to the aspects to model. Metamodeling is a means to avoid coding the language definitions manually and to create DSMLs in a visual environment, but the information represented by metamodels has a tendency to be incomplete, and sometimes inconsistent. Besides other issues, there is a need to describe additional constraints about the objects in the model. Transformation steps are also often imprecise without the ability to create constraints in the transformation steps. Therefore, constraint specification and validation lies at the heart of modeling and model transformation.

One of the most wide-spread approaches to constraint handling is the Object Constraint Language (OCL). OCL is a formal language that remains easy to read and write. OCL was originally created to extend the capabilities of UML, and define constraints for the model items. OCL can be used also in generic metamodeling environments to validate the models, or to define constraints in the model transformations. There are several interpreters and compilers that handle OCL constraints in modeling. These tools can extend the metamodel definitions, but they do not support optimization, or the constraint handling in model transformations, therefore, they are not always efficient enough.

Visual Modeling and Transformation System (VMTS) is an n-layer metamodeling and model transformation tool that grants full transparency between the layers (each layer is handled with the same methods). VMTS uses OCL constraints in both model validation and in the graph rewriting-based model transformation. This paper presents algorithms used in VMTS to optimize OCL constraint handling. Since the selection of the model items and their attributes referenced in the constraints has the most serious computational complexity, thus, the optimization algorithms focus on minimizing the number of navigation steps, and attribute queries required to check the constraint. The paper presents an efficient method to minimize the navigation steps of the constraints, and a caching mechanism to optimize the number of database queries during the validation process. Proofs are also provided to show that the optimized and the unoptimized code are functionally equivalent, and that the optimized code is never less efficient than the original. The presented algorithms do not use system-specific features, thus they can be used in any other modeling or model transformation environment.