MOMENT-OCL: A tool for OCL validation and OCL query evaluation

Contributed by Artur Boronat
Monday, 11 June 2007
Last Updated Monday, 11 June 2007

MOMENT-OCL is a tool for OCL constraint validation and OCL query evaluation that is integrated into the EMF. An EMF model definition conforms to a metamodel definition that is defined as an Ecore model definition, where Ecore is the meta-metamodel of the EMF. MOMENT-OCL uses the OCL library of the Kent Modeling Framework as front-end to parse OCL expressions, and the EMF as the front-end for the metamodeling framework. MOMENT-OCL uses the term rewriting engine Maude to execute OCL expressions.MOMENT-OCL represents OCL expressions as terms in an algebraic theory. This theory is generated from a pair (metamodel, ocl-expression), where metamodel is a metamodel definition in the EMF, and ocl-expression is either an OCL constraint or an OCL query that is given in textual form. MOMENT-OCL provides the following functionality:
 - Definition of OCL expressions, allowing the definition of OCL invariants and OCL queries. OCL expressions are defined for a specific metamodel definition in EMF. Fig. 1 shows the tree editor that constitutes the MOMENT-OCL tool interface. In the editor, we can add model nodes, representing specific metamodel definitions. A model node has properties, by means of which the metamodel definition and the model definition to be queried are referred to. These properties are shown in the common EMF Properties view. Within a model node, the user can define context nodes, representing the context of OCL expressions. In a context node, the user can add OCL invariants or OCL queries. All the OCL expressions that are defined within a context node have the same contextual type, indicated in a property of the corresponding context node.
Fig. 1. Tree editor of the MOMENT-OCL tool.

 - Syntactic and semantic analysis, indicating whether an OCL expression is well-formed or not, and whether an OCL constraint is meaningful for a specific metamodel definition or not. These tasks are performed by the OCL library of the Kent Modeling Framework.

 - Generation of the algebraic representation of OCL expressions, providing the Maude code that represents the operators and terms that are generated for a given OCL expression.

 - Execution of OCL invariants and OCL queries. In the tree view, when an OCL invariant is evaluated over a specific model definition, referred to in the corresponding model node, the icon of the corresponding OCL invariant node in the tree view changes its color indicating the result, as shown in Fig. 2: red if the invariant has failed, green if the invariant has succeeded, and yellow if there has been an error during the evaluation of the OCL constraint. As explained in Section \ref{sec:OCL}, an OCL expression is represented by a term, whose equational simplification, by using the equations that are generated for the pair (metamodel, ocl-expression) results in a canonical form that represents the resulting value of the OCL expression. The result that is obtained by the evaluation of an OCL invariant or an OCL query, i.e., the resulting term, is shown in the \emph{console view} of the tool, shown in Fig. 3. Despite the algebraic representation of the resulting term, it is shown in the concrete syntax format of the OCL language, thanks to the mixfix notation that is supported by Maude.

 - Persistence of OCL expressions, in textual format.   Fig. 2. Evaluation of OCL expressions in the MOMENT-OCL tool. Fig. 3. Console view of the MOMENT-OCL tool.For more information about the installation process of MOMENT-OCL and the instructions of use, see the tutorial and obtain the tool from the downloads section.