

Employing Classifying Terms for Testing Model Transformations

Martin Gogolla,¹ Antonio Vallecillo,² Loli Burgueno,³ Frank Hilken⁴

Keywords: UML and OCL model, Model testing, Model validation and verification.

1 Context

As the complexity of models and model transformations grows, there is an increasing need to count on powerful and precise testing techniques. One essential aspect of model transformation testing is the selection of effective test cases.

Our contribution proposes a new technique for developing test cases for UML and OCL models, based on an approach that automatically constructs object models for class models enriched by OCL constraints. By guiding the construction process through so-called classifying terms, the built test cases in form of object models are classified into equivalence classes. Classifying terms are general OCL terms on a class model that calculate a characteristic value for each object model. Each equivalence class is then defined by the set of object models with identical characteristic values and with one canonical representative object model. By inspecting these object models, a developer can explore properties of the class model and its constraints. Details can be found in [Go15] and [Hi17].

2 Basic Idea of Classifying Terms

In order to explain classifying terms, let us consider the following model exploration task: for a given class model, the developer wants to scroll through all valid object models, i.e., she wants to consider not only a single object model but the collection of all valid object models. This is realized in our current approach through a validation option ‘scrolling’ that spans up all object models.

The general difficulty appearing now is that many very similar, isomorphic object models will be taken into account now. The developer might expect to be shown interesting, structurally different object models. However, a development approach could offer the option to prevent that isomorphic object models are presented as distinct object models, when scrolling through the collection of valid object models

¹ University of Bremen, FB3, Bibliothekstr. 1, 28334 Bremen, gogolla@tzi.de

² University of Malaga, ETSI Informatica, Bulevar Louis Pasteur, 35, 29071 Malaga, av@lcc.uma.es

³ University of Malaga, ETSI Informatica, Bulevar Louis Pasteur, 35, 29071 Malaga, loli@lcc.uma.es

⁴ University of Bremen, FB3, Bibliothekstr. 1, 28334 Bremen, fhilken@tzi.de

As an answer, our approach gives the developer an explicit option to formulate her understanding of two object models being different. The technical realization is as follows: the developer specifies a closed OCL query term, i.e., a term without free variables, that can be evaluated in an object model and returns an integer as a characteristic value; in our approach, this term is called ‘classifying term’; each newly constructed object model has to show a different characteristic value. The classifying term determines an equivalence relationship on all object models. Two object models with the same characteristic value belong into the same equivalence class. The approach decides to choose only one representative from each equivalence class. The restriction that only one classifying term of type integer is considered can be lifted. In general, more than one classifying term (of type integer or boolean) can be handled.

Classifying terms can be employed for exploring the class model in order to see few diverse object models instead of many similar ones. The focus of exploration is determined by the modeler through the terms. By inspecting the constructed object models and checking their properties, the modeler gains insight into the characteristics of the class model including the OCL constraints and makes them alive. Using boolean classifying terms, one can draw conclusions which model properties (expressed as classifying terms) are allowed simultaneously in an object model. Thus one can analyze dependencies between requirements similar to invariant independence which checks whether a given invariant is a logical consequence from other invariants.

3 Conclusion

We have introduced classifying terms, an instrument for exploring object models in the context of a UML class model and accompanying OCL constraints. Classifying terms allow the developer to construct relevant test cases in form of object models in a goal-oriented way. Classifying terms determine equivalence classes of test cases, selection of representatives and exploration of model properties. The full paper also shows how classifying terms can be applied for model transformations. The work can be continued in various directions. It would be interesting to consider more than one equivalence class representative by distinguishing between first and second level classifying terms, where second level terms are only applied for non-empty first level equivalence classes. Larger case studies should give more feedback on the features and scalability of the approach.

References

- [Go15] Gogolla, Martin; Vallecillo, Antonio; Burgueno, Loli; Hilken, Frank: Employing Classifying Terms for Testing Model Transformations. In (Cabot, Jordi; Egyed, Alexander, eds): Proc. 18th Int. Conf. Model Driven Engineering Languages and Systems (MoDELS’2015). ACM, pp. 312–321, 2015.
- [Hi17] Hilken, Frank; Gogolla, Martin; Vallecillo, Antonio; Burgueno, Loli: Testing Models and Model Transformations using Classifying Terms. *Software and Systems Modeling*, 2017. DOI 10.1007/s10270-016-0568-3.