A Posteriori Typing for Model-Driven Engineering: Concepts, Analysis, and Applications

Juan de Lara
Modelling and Software Engineering Group
http://miso.es
Universidad Autónoma de Madrid
Spain
Juan.deLara@uam.es

Esther Guerra
Modelling and Software Engineering Group
http://miso.es
Universidad Autónoma de Madrid
Spain
Esther.Guerra@uam.es

ABSTRACT

Model-Driven Engineering (MDE) is a software engineering paradigm where models are actively used to specify, test, simulate, analyse and maintain the systems to be built, among other activities. Models can be defined using general-purpose modelling languages like the UML, but for particular domains, the use of domain-specific languages is pervasive. Either way, models must conform to a meta-model which defines their abstract syntax.

In MDE, the definition of model management operations – often typed over project-specific meta-models – is recurrent. However, even if two operations are similar, they must be developed from scratch whenever they are applied to instances of different meta-models. This is so as operations defined (i.e., typed) over a meta-model cannot be directly reused for another. Part of this difficulty of reuse is because classes in meta-models are used in two ways: as templates to create objects and as static classifiers for them. These two aspects are inherently tied in most meta-modelling approaches, which results in unnecessarily rigid systems and hinders reusability of MDE artefacts.

To enhance flexibility and reuse in MDE, we propose an approach to decouple object creation from typing [1]. The approach relies on standard mechanisms for object creation, and proposes the notion of *a posteriori* typing as a means to retype objects and enable multiple, partial, dynamic typings.

A posteriori typing enhances flexibility because it allows models to be retyped with respect to other meta-models. Hence, we distinguish between *creation* meta-models used to construct models, and *role* meta-models into which models are retyped. This permits unanticipated reuse, as a model management operation defined for a role meta-model can be reused *as-is* with models built using a different creation meta-model, once such models are reclassified. Moreover, our approach permits expressing some types of bidirectional model transformations by reclassification. The transformations defined as reclassifications have better performance than the equivalent ones defined with traditional transformation languages, because reclassification does not require creating new objects.

In [1], we propose two mechanisms to define *a posteriori* typings: type-level (mappings between meta-models) and instance-level (set of model queries). The paper presents the underlying theory and type correctness criteria of both mechanisms, defines some analysis methods, identifies practical restrictions for retyping specifications, and demonstrates the feasibility of the approach by an implementation atop our meta-modelling tool MetaDepth. We also explore application scenarios of *a posteriori* typing (to define transformations, for model transformation reuse, and to improve transformation expressiveness by dynamic type change), and present an experiment showing the potential performance gains when expressing transformations as retypings.

The tool is available at http://metadepth.org/. A catalogue of transformations expressed as retypings, and retypings bridging recurring meta-model heterogeneities, are available at http://miso.es/aposteriori/.

CCS CONCEPTS

• **Software and its engineering → Software design engineering**; *System modeling languages*; *Reusability*;

KEYWORDS

Model-driven engineering, reuse, model typing, partial typing, dynamic typing, model transformations, bidirectionality, MetaDepth

ACM Reference Format:
Juan de Lara and Esther Guerra. 2018. *A Posteriori* Typing for Model-Driven Engineering: Concepts, Analysis, and Applications. In *ICSE '18: 40th International Conference on Software Engineering , May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, Article 4, 1 page. https://doi.org/10.1145/3180155.3182545

ACKNOWLEDGMENTS

Work funded by the Spanish MINECO (TIN2014-52129-R) and the R&D programme of Madrid (S2013/ICE-3006).

REFERENCES

[1] Juan de Lara and Esther Guerra. 2017. *A posteriori* typing for model-driven engineering: Concepts, analysis, and applications. *ACM Trans. Softw. Eng. Methodol.* 25, 4 (2017), 31:1–31:60. https://doi.org/10.1145/3063384

ICSE '18, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5638-1/18/05…$15.00
https://doi.org/10.1145/3180155.3182545