Science of Computer Programming 107–108 (2015) 41–63

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Institution-based foundations for verification in the context of model-driven engineering

Daniel Calegari [a,*], Nora Szasz [b]

[a] Facultad de Ingeniería, Universidad de la República, Montevideo 11300, Uruguay
[b] Facultad de Ingeniería, Universidad ORT Uruguay, Montevideo 11100, Uruguay

ARTICLE INFO

Article history:
Received 4 May 2014
Received in revised form 20 February 2015
Accepted 20 February 2015
Available online 5 March 2015

Keywords:
Verification
Formal semantics
MOF
QVT-relations
Theory of institutions

ABSTRACT

Within the Model-Driven Engineering (MDE) paradigm, a separation of duties between software developers is usually proposed to cope with formal verification issues. MDE experts are responsible for the definition of models and model transformations, while formal verification experts conduct the verification process. This schema should be aided by (semi)automatic translations from the MDE elements to their formal representation in the potentially many semantic domains used for verification, and also by translations between these domains. Translations may be useful to perform a heterogeneous verification, i.e. using different domains for the verification of each part of the whole problem, and also to integrate MDE elements with the specification and verification of other traditional software artifacts. However, this schema requires formal foundations allowing the representation of the MDE elements in such a way that it is possible to ensure that translations are semantic-preserving. The aim of this paper is to present a formalization of the MDE elements using the Theory of Institutions. We provide institutions for the representation of MDE elements based on the MOF and QVT-Relations standards. We also show how the theory assists with these requirements for the definition of an environment for the formal verification of MDE elements using heterogeneous verification approaches.

1. Introduction

The Model-Driven Engineering paradigm (MDE, [1]) refers to the systematic use of models as primary engineering artifacts throughout the engineering process. In particular, the paradigm envisions a software development life-cycle driven by models representing different views of the system to be constructed. Its feasibility is based on the existence of a (semi)automatic construction process driven by model transformations, starting from abstract models of the system and transforming them until an executable model is generated. It also encompasses other engineering efforts of a complete software engineering process, such as maintenance and reverse engineering. This approach tends to improve efficiency on the construction process and the trustworthiness of the resulting products.

In the MDE ecosystem everything is a model, even the code is considered as a model. In this context, a model is an abstraction of the system or its environment. Models are defined from metamodels, i.e. models which introduce the syntax and semantics of certain domain-specific kind of models. When a model is structurally compliant with respect to its metamodel, their relation is called *conformance* [2]. In some cases, there are conditions (called invariants) that cannot be

* Corresponding author. Tel.: +598 27114244; fax: +598 27110469.
E-mail addresses: dcalegar@fing.edu.uy (D. Calegari), szasz@ort.edu.uy (N. Szasz).

http://dx.doi.org/10.1016/j.scico.2015.02.006
0167-6423/

captured by the structural rules of this language, in which case the language is supplemented with another language, e.g. the Object Constraint Language (OCL, [3]). A model transformation (or just transformation from now on) is basically the automatic generation of a target model from a source model, according to a transformation definition, i.e. a set of rules that together describe how a model in the source language can be transformed into a model in the target language [4]. The Object Management Group (OMG) conducted a standardization process of languages for MDE. They defined the MetaObject Facility (MOF, [5]) as the language for metamodeling, and three transformation languages with different transformation approaches. The Query/View/Transformation Relations (QVT-Relations, [6]) is one of those languages which follows a relational approach which consists on defining transformations as declarative relations between source and target elements.

The quality of the whole MDE process strongly depends on the quality of the models and model transformations. It is increased by verification of the generated models and of the model transformations, at early development stages. In some cases formal methods (i.e. mathematically based techniques) arise as a tool for strengthening verification results. The specification and verification of an MDE-built system has some parallelism with traditional software systems. The formal treatment of a problem requires some notation with formal semantics, along with a deductive system for reasoning. This is often considered difficult to apply because requires significant mathematical experience, which leads to a mismatch problem between software engineering expectations and formal methods possibilities.

To cope with this situation, a separation of duties between software developers is usually proposed, giving rise to different technological spaces [7], i.e. working contexts with a set of associated concepts, body of knowledge, tools, required skills, and possibilities. In general terms, MDE experts define models and transformations, while formal experts conduct the verification process, often aided by some (semi)automatic generation process which translates the MDE elements to their formal representation into the domain used for verification purposes. The formal representation is usually defined in a unified semantic domain (e.g. [8–10]), having tools for conducting the verification process and for retrieving some feedback to the MDE experts. However, the use of a mathematical formalism serving as a unique semantic basis for verification can be quite restrictive considering that there are several alternatives which depend on those properties that must be addressed in each specific case [11]. Depending on the problem, it can be useful to have a representation of MDE elements in different logical domains. To achieve this, one possibility is to generate partial or complete formal representations of elements from the MDE technological space in different formal domains, and then "connect" these specifications via translations between the logical domains. These connections may be useful to perform a heterogeneous verification, i.e. using different domains for the verification of each part of the whole problem [12], and also to integrate MDE elements with the specification and verification of other traditional software artifacts [13]. The currently used semantic domains are not well adapted for the definition of such schema, since it requires the formal representation of the MDE elements in such a way that it is possible to ensure that translations between these domains are semantic-preserving. If the semantics is not preserved then it is no possible to ensure that proofs in different domains (e.g. first-order or rewriting logics) are sound with respect to the original problem.

The aim of this paper is to present a formalization of the MDE elements (based on the MOF and QVT-Relations standards) using the Theory of Institutions [14]. More specifically, we focus on structural semantic aspects instead of dynamic semantics aspects of MDE elements. The theory provides ways of representing the syntax and semantics of the MDE elements: models, metamodels, the conformance relation between them, and model transformations as institutions, in some consistent and interdependent way. An institution abstracts the representation of any specification language, providing mechanisms for the definition of semantic-preserving translations between semantic domains (which are also represented as institutions).

This paper is a substantially extended and thoroughly revised version of [15]. Additional material includes:

- a detailed and improved formalization of the institutions, splitting them into standard versions for the representation of the problem and proof-theoretical extensions devised to be used within a proof environment;
- an extended formal treatment of the conformance relation, with a discussion about the inclusion of an institution for OCL and for model typing;
- a discussion about the definition of an institution-based environment for the formal verification of MDE elements using heterogeneous approaches [16], together with a detailed discussion with respect to related work.

The remainder of the paper is structured as follows. In Section 2 we present the basic notions of the Theory of Institutions which will be useful for the theoretical understanding of the following sections. For a more detailed introduction to the topic refer to [17,18]. After this, we provide an institution-based formalization of the MDE elements which is the main concern of this paper: in Section 3 we provide formal definitions of institutions for representing models, metamodels and the MOF-based conformance relation between them, in Section 4 we define an institution for QVT-Relations model transformations, and in Section 5 we define an extension of these institutions in order to be used within a proof environment. In Section 6 we introduce how these formal foundations can be used for the definition of an environment for the formal verification of MDE elements using heterogeneous verification approaches. In Section 7 we summarize related works. Finally, in Section 8 we present some conclusions and guidelines for future work.