A Methodological Approach for the Coupled Evolution of Metamodels and ATL Transformations

Davide Di Ruscio, Ludovico Iovino, and Alfonso Pierantonio

Department of Information Engineering, Computer Science and Mathematics University of L’Aquila
{davide.diruscio,ludovico iovino, alfonso.pierantonio}@univaq.it

Abstract. Model-Driven Engineering is a software discipline that relies on (meta) models as first class entities and that aims to develop, maintain and evolve software by exploiting model transformations. Analogously to software, metamodels are subject to evolutionary pressures which might compromise a wide range of artefacts including transformations. In contrast with the problem of metamodel/model co-evolution, the problem of adapting model transformations according to the changes operated on the corresponding metamodels is to a great extent unexplored. This is largely due to its intricacy but also to the difficulty in having a mature process which on one hand is able to evaluate the cost and benefits of adaptations, and on the other hand ensures that consistent methods are used to maintain quality and design integrity during the adaptation. This paper proposes a methodological approach to the coupled evolution of ATL transformations aiming at evaluating its sustainability prior to any adaptation step based on the assessment of change impact significance.

1 Introduction

Model-driven engineering (MDE) is a software discipline that employs models for describing problems in an application domain by means of metamodels. Different abstraction levels are bridged together by automated transformations which permit source models to be mapped to target models. These artifacts and the interrelationships among them constitute an ecosystem at whose core there are metamodels [4]. Since evolution in software is anything but a rare occurrence [12], it can affect metamodels as well [19] causing a ripple effect over the rest of the ecosystem. However, whenever a metamodel undergoes modifications, it is of vital relevance that the impact of such changes is fully understood prior initiating their propagation: regardless how urgent the motivations for changing a metamodel are, underestimating the difficulties in restoring the consistency in the ecosystem can lead to an impasse, in which no progress can be made [6].

The problem of metamodel/model coupled evolution[6] has been already extensively investigated (e.g., see [6][2][16][8][10]). The existing approaches provide tools and techniques to define and apply migration strategies able to take models conforming to the original metamodel and to produce models conforming to the evolved metamodel. On the contrary, despite its relevance the metamodel/transformation co-evolution problem

1 Throughout this paper we will use the terms coupled evolution, co-evolution and co-adaptation as synonyms whenever it does not give place to misinterpretations.
A Methodological Approach for the Coupled Evolution

is still open and requires further investigations. In fact, adapting transformations does not only take into account the domain conformance \[14\] between the definition of a transformation and its metamodels but must consider also the intelligence used by the transformation for generating the target model elements. Very few attempts have been made so far and generally they tend to re-apply the same techniques used for the metamodel/model co-evolution, as in \[13\] where higher-order transformations (HOTs) are used to migrate, whenever possible, existing transformations according to occurred metamodel changes. Thus, only the most obvious cases, such as renamings and deletions, are covered leaving the responsibility of managing the most complex ones to the modeler who typically face the problem with individual and spontaneous skills. This is largely due to the intricacies of the problem but also to the lack of a mature process which on one hand is able to evaluate the cost and benefits of adaptations, and on the other hand ensures that consistent methods are used to maintain quality and design integrity during the adaptation.

This paper proposes a comprehensive and methodological approach to the coupled evolution of ATL transformations. As with many engineering activity, measurement is crucial in order to assess at early stages of a process the sustainability of the costs versus the benefits. Therefore, a process is proposed for the systematic co-evolution of artifacts and which includes the following activities: i) establishing the dependencies between a transformation and its (source) metamodel; ii) evaluating the cost of the adaptation; iii) deciding whether it is sustainable or not by eventually reconsidering certain decisions; and finally iv) if the assessment has a positive outcome the impacted transformation is adapted. The main contribution of the paper is to define a methodology in which an early assessment of the impact cost and significance is conducted and which can provide the modeler with the right tools and techniques for addressing a complex problem in a more disciplined way.

The structure of the paper is as follows: In Section 2, we discuss an example which motivates the metamodel/transformation coupled evolution problem. In Section 3 we discuss a classification of metamodel changes according to their impact on the existing transformations. The proposed process for the systematic co-evolution of metamodels and ATL transformations is described in Section 4. Related work is described in Section 5, and the paper is concluded in Section 6.

2 Motivating Scenario

In MDE model transformations play a key role since they are able to generate target models starting from source ones according to transformation rules, which are defined with respect to source and target metamodels. For instance, Listing 1.1 shows an ATL transformation able to transform models conforming to the PetriNet metamodel reported in Figure 1.a, and to generate Petri Net Markup Language (PNML) \[1\] models conforming to the metamodel in Figure 2.

According to the metamodel in Figure 1.a a PetriNetModel mainly consists of Places and Transitions which are contained in the Net element. Concerning the metamodel in Figure 2 the metaclass PNMLDocument represents the root element which is composed of Petri nets specified by means of NetElement instances. A Petri net