Thesis Overview:

A Formal Foundation for Transformations in Model Driven Engineering

Roxana S. Giandini
Universidad Nacional de La Plata, Facultad de Informática, February 2008
PhD Thesis in Computer Science
Advisor: Claudia Pons
[giandini,cpons]@lifia.info.unlp.edu.ar

In the MDE (Model Driven Engineering) paradigm, models are thought to be the primary conductors in all software development aspects. A PIM (Platform Independent Model) is transformed into one or more PSMs (Platform Specific Model); hence, a specific PSM is generated for each specific technological platform. In turn these output models may be transformed into another set of models until finally the output consists of program code that can be executed. Model transformation is the MDE engine; models are no longer mere contemplative entities and become productive entities. The MDE initiative covers a broad spectrum of research areas: modeling languages, definition of languages for model transformation, construction of support tools for the different tasks involved, application of concepts to development methods and specific domains, etc. Currently, some of these aspects are well-based, and are being applied with some success; however, other aspects are still undergoing their definition process. In this context, it is necessary to make every effort to convert MDE and its concepts and related techniques into a coherent proposal, based on open standards, and supported by mature tools and techniques. Model transformations require specific languages for their definition; these languages should have a formal base, for example, a metamodel that supports them and allows for an automated treatment.
Also, transformations can be specified or implemented using different tools and different languages and they can be manipulated as black-box entities. The ability to orchestrate different transformations in a flexible and reliable manner in order to produce the required output is a major challenge in MDE. There are various approaches for model transformation that offer forms of compositionality, either based on internal or external composition of transformations. For instance, the QVT (Query\View\Transformation) standard specifies a language in which one is able to express transformation definitions that consist of a number of mapping rules. The mapping rules may be combined by internal (or fine-grained) composition of transformations. On the other hand, the combination of transformations as black box entities is called the external (or coarse-grained) composition of transformations. The QVT operational formalism provides a set of elementary programmatic constructs to express external chaining of transformations. It offers the possibility to write loops, if-then-else controls, to pass parameters to the transformations and the possibility to retrieve the output of a transformation and to pass as input to the consequent transformation. Despite the fact that the QVT offers two modeling perspectives - it allows us to specify what the transformation does (declarative QVT) as well as how the transformation is accomplished (operational QVT) - most of the current work on model transformations seems essentially operational and executable in nature. Executable descriptions are necessary from the point of view of implementations; but from a conceptual point of view, transformations can also be viewed as descriptive models by stating only the properties a transformation has to fulfill and by omitting execution details. In particular, regarding the composition problem, most approaches are focused only on the operational aspects of the composition, neglecting its descriptive side. Such partial visualization of the composition problem is useful to offer a reasonable solution to a wide range of practical needs. However it does not cover the entire composition problem spectrum. Therefore the formal definition of transformation languages must include a holistic foundation for mechanisms of composition.

For the exposed motives, this work has the following general goal:

To define a formal foundation for model transformation languages. This goal is achieved by means of the following sub-targets:

- To present a language to express transformations among models whose initial metamodel is inspired in the QVT language, which is the OMG (Object Management Group) standard specification for transformations. The language we propose aims at being minimal for expressing queries and transformations among models. To maintain simplicity and to reduce user training time are the main advantages of this minimal approach. Furthermore, the proposal maximizes the use of OMG existing standard specifications, making much easier its implementation.

- To formally define (applying the theory of problems formalism) the syntax and the semantics of the above mentioned minimal language, allowing consistency validation between a (declarative) specification written in the language and its (operational) implementation.

- To describe how the algebraic theory of problems can be applied as a basis to build a mathematical foundation for the transformation composition problem embracing both dimensions (descriptive and operational). In particular, we intuitively and formally define the three commonly know combinatorial operators in the history of computing: union (no-determinism), sequence and *fork* (parallelism) operations. In addition, inverse operation on transformations is defined.
  Having the composition machinery accurately synchronized at both QVT levels would allow us to fully exploit the divide-and conquer paradigm in the development of model transformations. That is to say, we are able to break a declarative transformation into sub transformations whose implementations might be recombined in an implementation for the original transformation. Breaking a declarative transformation means to state the given transformation in terms of operations on composing declarative transformations, while the recombination to get the solution is done by means of the corresponding operations on the solution of each component.

Fundamentally, the principal contribution of this thesis to the MDE methodology consists in to contribute to the maturity of the model transformation languages. Specifically, this contribution consists in:

- To propose mechanisms of consistency validation between descriptions of transformation, written at different levels of abstraction (declarative and operational).

- To provide a mathematical characterization of the transformation composition problem that results in a clear picture of the composition machinery at both QVT levels. It can be used as a foundation to improve the QVT specification towards the construction of simpler and more powerful transformation composition machinery, which will serve as a ground for the construction of precise and solid model transformation languages and tools.

- The proposed mechanisms propitiate and facilitate the application and use of the model driven development methodology.

Roxana Giandini
giandini@lifia.info.unlp.edu.ar