A Framework for Evaluating the Ontological Quality of Languages in MDE Environments

Fáber D. Giraldo[1,2]

[1] System and Computer Engineering, University of Quindío, Colombia
[2] PROS Research Centre, Universitat Politècnica de València, Spain
email: fdgiraldo@uniquindio.edu.co, fdgiraldo@pros.upv.es

Abstract. Information system development methods that follow the model-driven engineering (MDE) approach commonly prescribe the use of multiple viewpoints. Viewpoints have associated languages and notations according with the MDE approach, and they often have differents abstraction and granularity levels. The enactment of such methods in projects entails risks that threaten its success. For instance, the set of modelling languages may have redundant and/or missing constructs, the modellers can attempt to use a language that is inappropriate for a given level of abstraction, if a language covers several abstraction levels then some of its constructs may have a semantic overload. This work proposes a framework to evaluate the ontological quality of a set of languages; that is, measuring to what extent the set of languages comply with principles and guidelines intended to minimise the above-mentioned risks and to facilitate their combined use within an MDE project. Also, the framework will allow to identify relations between language constructs and software concepts, so it is expected to aid MDE method engineers in the task of designing model transformations.

1 Introduction

One of the main challenges in the model-driven engineering (MDE) initiative is the management and the integration of languages and models formulated to support multiple views during the information systems (IS) development. Modelling languages create and use models that represent materialized views over *concerns* of an IS, according to rules defined by viewpoints. In this way, it is possible to mitigate the problems associated with the management of transversal features of an information system [6].

Generally, the considerations addressed by the languages and models contemplate: *i)* business concerns; *ii)* non-functional features derived from quality attributes; *iii)* new paradigms for software construction (e.g., aspects, collaboration or requirements characterization); *iv)* functional and logical concerns.

Now a proliferation of languages is evident (with their abstract and concrete syntax and their semantics), and proposals that emerge with the purpose of managing specific views or perspectives of an IS. There are proposals that define their wide set of symbols and concepts, and which have not been interesting by

the academical, researchers and industrial communities. Also, there are proposals based on excessively stereotyped UML, which limit the expressiveness or meaning of the models to the stereotyped classes, and/or modifications (or additions) of UML symbols. New UML based notations could not fully satisfy the meaning-meaningful relation associated with a specific domain. Therefore, people who designed a notation of this style should be able to transmit the meaning of the concept to express.

This paper presents a proposal for defining the foundations of an ontological evaluation framework to be applied over languages used in MDE projects, with the purpose of validating the *quality* of these elements in the management and technical implementation of an IS according with the views (stakeholders) involved, and features of the MDE itself. This paper is organized as follows: the Section 2 introduces the problem statement that promotes our idea. Section 3 presents the goals and the methodology of our research. Section 4 briefly presents an overview of our proposed solution; and finally, Section 5 reviews the state of the art about quality of models and modelling languages.

2 Problem statement

MDE proposes modelling languages as the new abstraction units, hence, the introduction of a new language in an MDE enviroment should be as easy as creating a new class in a Java project [21]. In the MDE projects one can often find several proposals of languages, models, notations and tools that manage specific concerns belonging to multiple views of an IS. But in practice, several of these proposals are not applied due to problems detected in their integration with a previous set of IS models. Also, there are some MDE initiatives where the domains have associated metamodels, but their representation is made by UML stereotypes or by functionalities in traditional use-case scenarios.

The adoption of the MDE approaches have guided the development of a large number of initiatives; although it emphasizes the use of models as primary artifacts of a software construction process, ***it causes a conceptual divergence in the support of specific views and/or concerns belonging to an IS***. This phenomenon is strengthened by the lack of (semantic) support offered by UML or other traditional notations.

In spite of the development of metamodels, reference architectural frameworks, and ontological frameworks, it has been recognized and widely reported the inability for consistently modelling all related and inherent views in an IS using a single metamodel or a single notation. In [18] is shown how a single metamodel can only be feasible if the granularity and abstraction level of the viewpoints are similar, which is impossible to guarantee in a typical MDE scenario, taking into account that the viewpoints often have.

Due to the increasing collection of modelling languages and notations, several methods to assess the quality of modelling languages have been proposed. Some proposals provide guidelines for designing languages based on principles drawn from semiotics and cognitive theory (see the state of the art in Section 5). The

rationale behind such proposals is that models are a means to express conceptions about some phenomena, to reason about such conceptions, and to communicate them to others.

Although these methods emphasize the importance of the relationship between the concepts of the modeled concern with respect to the used notation, the required effort to formalize semantic definitions become a high cognitive load for those involved in an MDE process. Also, these frameworks do not consider the most relevant features of the MDE itself into their formulation. It can be explained as a natural consequence of the several (divergents) interpretations of MDE that results from attempts for new notations and languages framed in MDE without a rationale support (particular interpretations on MDE). There are so many ways to adopt an MDE approach that it is not possible to establish general conclusions about MDE itself [4].

Also, the identified guidelines and frameworks do not evaluate the quality of models from dimensions such as the *mapping* or translation between models (even models that belong to the same viewpoint of an IS), neither successful experiences originated from massive application of a modelling technique in a particular MDE environment.

3 Research methodology

The main goal of this work is to ***formulate a method for the evaluation of the ontological quality of a set of languages jointly used within an MDE project***. This work aims to verify whether it is possible to generate a framework for the evaluation of languages, so that it can determinate how one language from the MDE viewpoint is structured. It means, if the language supports views, abstraction levels, integration capabilities, and if it is possible to generate full functional software from the language(s)/model(s) under review. The framework must indicate what is missing or what is not neccesary for using a language in an MDE environment correctly.

3.1 Research questions

The research will focus on resolving the following questions:

- *(RQ1)* What problems are evidenced in model-driven projects related to the selection of languages?
- *(RQ2)* What is the set of concepts that are required to model when we are in a model-driven project?
- *(RQ3)* When a set of modelling languages is selected to be used in combination in an MDE project, are there methods for evaluating the suitability such set of languages?
- *(RQ4)* Propose a method for the evaluation of the ontological quality of a set of languages used jointly within a model-driven project.
- *(RQ5)* What advantages/disadvantages are obtained by the application of the proposed method?

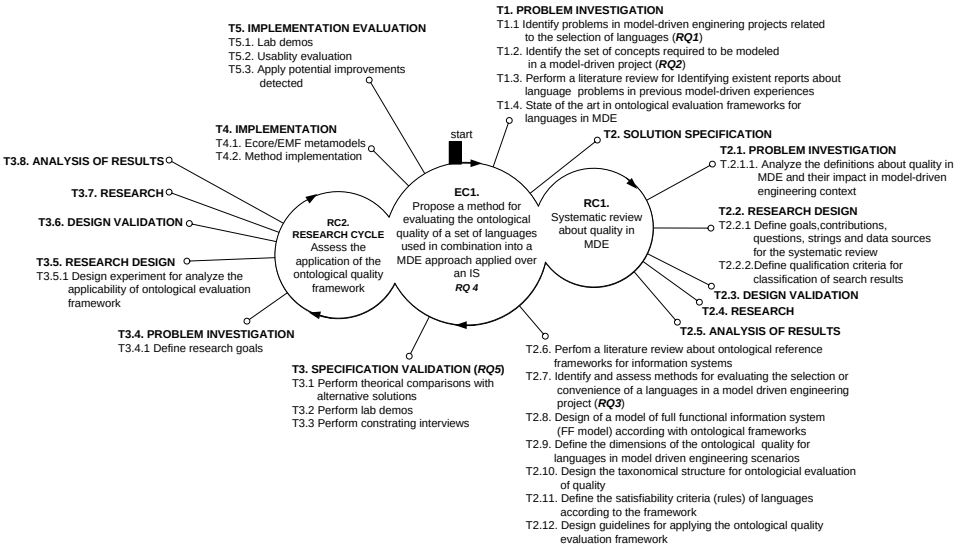

Fig. 1. Design Science scheme applied to the research project

In a practical way, through the proposed evaluation framework, the language designer or language engineer[9] can determinate/evaluate if a given language and/or model (and its associated artifacts) has the capability for creating models from the metamodel basis, generating concrete syntaxis, managing views, viewpoints and perspectives, and the potentiality (capabilities) of integration offered by the language with respect to other languages used in MDE environments that support domains according with the existing perspectives (in similar or differents abstraction levels exposed in these environments).

3.2 Methodology

This work will use the *Design Science* guidelines proposed in [22] with the purpose of defining, managing and differentiating the *practical* and *knowledge* problems along the project. For the context of this research, the knowledge problems (covered by the *research cycles* - *RC* of Figure 1) are related to the contrast of existing knowledge about IS construction under the model-driven paradigm, and the scope and applicability of our proposed ontological framework in MDE contexts. A practical problem (*engineering cycle* - *EC*) is related to the formulation of the ontological quality evaluation framework, from MDE basis, enterprise architecture and ontological IS foundations.

Our engineering cycle starts with the identification of the involved stakeholders through **survey-based research** and **ethnography research**. For this

case, we consider people involved in model-driven projects, such as languages users and method engineers. This identification includes the (further) expectations about the use of combination of languages for developing IS under model-driven principles. To answer *RQ1* we will perform a **literature review**. *RQ2* will be answered by means of a **literature review** of the most representative IS ontological reference frameworks contrasting their conceptions about elements of an IS with MDE features. To respond *RQ3*, a **literature review** must be performed in order to identify similar evaluation frameworks in ontological levels. *RQ4* will be answered by the **design** of the model of the ontological framework for the evaluation of languages in MDE context, with its respective use specification. *RQ4* implies a **systematic review** about the concept of quality in model-driven engineering to identify the relevance and scope of our proposal with respect to representative trends of quality in MDE. To answer *RQ5*, we will perform **theoretical comparisons** and **cognitive analysis** in which we assess the effort needed to apply the proposal, a **lab demo** in which we apply our proposal in a small but realistic case, and a **controlled experiment** in which we evaluate our proposal in a rigorous way.

4 Overview of the solution

The quality evaluation framework proposed in this research is conceived as a conceptual, methodology and technology tool for the evaluation of language proposals and models (as productions of languages), whose purpose is to assess one set of languages/models regarding its incorporation and adoption capabilities in a MDE enviroment. Also, this framework must establish the capacity of languages to support automation and software generation. Figure 2 exposes the initial version of a metamodel that conforms the quality evaluation framework. This metamodel will be updated according with the refinement of the *quality* term as a consecuence of advances produced by the introduction of the most applicable philosophical background.

Figure 3 presents a initial collection of metaclasses which considers some typical elements existing into a MDE environment, to be assesed through the application of the evaluation framework proposed in this work.

The existence of several languages in a IS model-driven project could derivate evidences about those languages that overlap and model IS aspects in a redundant way, or conversely, some of the IS aspects could not be covered by any language. Both situations supose a risk for MDE projects. We think it influences in the adoption of model-driven methods and tools. Therefore, when the languages and tools are established accordly, it will favor the adoption of model-driven intiatives. When the framework is used, it will be possible to optimize the selection of languages; and therefore, when it is applied in a model-driven project we expect that the development time is reduced and the optimization of resources used with respect to non-use of the framework.

When our framework can be applied, we expect that our framework allows answering, among others, the following questions:

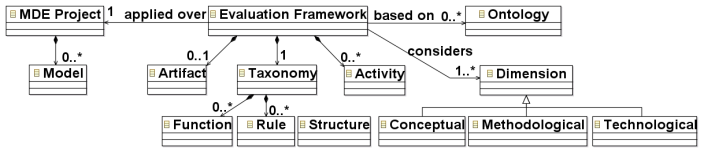

Fig. 2. Overview of the solution as MOF compliant metamodel

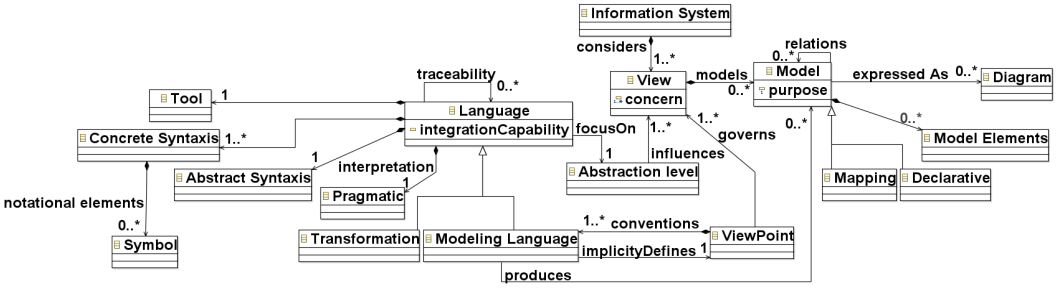

Fig. 3. Concepts involved into an IS development from MDE viewpoint

- Does the model describe more information than it is really needed?
- Is it possible to evidence whether the identified models are for declarative or mapping purposes?
- Are the language and notation according to the MDE?
- Do the models allow to perform traceability?
- Is there any aspect of the IS not covered by the identified models?
- Can the models generate fully functional software?
- Does the model cover a specific view of the IS?

5 State of the Art

The *quality* term in MDE context is reduced to two representative trends: one trend is about the cognitive and semiotic evaluation of notations, usability of modelling languages and modelling process. As a consecuence, this trend has several frameworks and guidelines that focus on recomendations for building models. It is evident frameworks such as SEQUAL[10][11], guidelines for modelling frames into a *"quality model"* concept, such as [12][19][1][7], application guideline reports such as [8], and specific guidelines for working with notations such as [15][16], among others. In [17] is reported at least three frameworks are empirically evaluated, from a static view focusing on the resultant model process but not on the act of modelling itself.

On the other hand, the *quality models* term is used to justify metamodels that relate to the conceptual set of software quality assurance, according with ISO 9126 or ISO 25000 standards. In [13][14] an inductivist judgment is applied in order to justify MDE as a quality engineering, so that models are the basis for development tasks, and therefore, the incorporation of software quality assurance concepts at MDA levels (M1 level) improving the quality of artifacts derived from it. Other metamodelling works such as [3], proposes the definition of specific metamodels for formalizing the software quality concept, so it is possible to model the information about quality. In [2], authors present an ISO/IEC 9126 quality model adaptation for dealing with the specific properties of metamodels. The main goal of this kind of work is the formalization of software quality assurance concepts as MOF-compliant metamodels.

Most of the above works do not cover the quality of languages and models from a MDE viewpoint, i.e., they do not explain how multiple proposals for managing multiple views in a MDE scenario can co-exist. In [5] authors highlight that the *quality* term in models have not a consistent definition, and it is differently defined, conceptualized and operationalized according with the discourse of each previous research proposals. Works like [20] propose an integration method for multiple languages supported by a reference framework (RM-ODP), but it does not specify how to evaluate the sufficiency, convecience or deficiency of these languages as such in a model-driven scenario.

Acknowledgments

The author thanks Prs Oscar Pastor and Sergio España (PROS Research Centre, Universitat Politècnica de València, Spain) for their collaborations and their valuable feedback. The author thanks to Colciencias (Colombia) for funding of this work through Colciencias Grant call 512-2010, and Spanish Ministry of Science and Innovation project PROS-Req (TIN2010-19130-C02-02).

References

1. Jörg Becker, Michael Rosemann, and Christoph von Uthmann. Guidelines of business process modeling, 2000.
2. Manuel Bertoa and Antonio Vallecillo. Quality attributes for software metamodels. In *13th TOOLS Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2010)*.
3. Xavier Burgus, Xavier Franch, and Josep Rib. *A MOF-Compliant Approach to Software Quality Modeling Conceptual Modeling ER 2005*, volume 3716 of *Lecture Notes in Computer Science*, pages 176–191. Springer Berlin / Heidelberg, 2005.
4. Jordi Cabot. Is transitioning to MDE revolutionary (for companies adopting it)?
5. Peter Fettke, Constantin Houy, Armella-Lucia Vella, and Peter Loos. *Towards the Reconstruction and Evaluation of Conceptual Model Quality Discourses Methodical Framework and Application in the Context of Model Understandability*, volume 113 of *Lecture Notes in Business Information Processing*, chapter 28, pages 406–421. Springer Berlin Heidelberg, 2012.

6. Robert France and Bernhard Rumpe. Model-driven development of complex software: A research roadmap, 2007.
7. Andrew Gemino and Yair Wand. Evaluating modeling techniques based on models of learning. *Commun. ACM*, 46(10):79–84, 2003.
8. Brian Henderson-Sellers, Graham Low, and Cesar Gonzalez-Perez. *Semiotic Considerations for the Design of an Agent-Oriented Modelling Language*, volume 113 of *Lecture Notes in Business Information Processing*, chapter 29, pages 422–434. Springer Berlin Heidelberg, 2012.
9. Anneke Kleppe. *Software Language Engineering: Creating Domain-Specific Languages Using Metamodels*. Addison-Wesley Professional, 1 edition, 2008. ISBN 0321553454, 9780321553454.
10. J. Krogstie. *Model-Based Development and Evolution of Information Systems: A Quality Approach*. Springer, 2012.
11. John Krogstie. *Evaluating UML Using a Generic Quality Framework*, pages 1459–464. IGI Global, 2009.
12. J. Mendling, H. A. Reijers, and W. M. P. van der Aalst. Seven process modeling guidelines (7pmg). *Inf. Softw. Technol.*, 52(2):127–136, 2010.
13. Parastoo Mohagheghi and Vegard Dehlen. An overview of quality frameworks in model-driven engineering and observations on transformation quality. *Quality in Modeling Workshop materials*, page 3, 2007.
14. Parastoo Mohagheghi, Vegard Dehlen, and Tor Neple. A metamodel and supporting process and tool for specifying quality models in model-based software development. *Nordic J. of Computing*, 14(4):301–320, 2008.
15. D. Moody. The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. *Software Engineering, IEEE Transactions on*, 35(6):756–779, 2009.
16. Daniel Moody, Patrick Heymans, and Raimundas Matuleviius. Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation. *Requirements Engineering*, 15(2):141–175, 2010.
17. Jakob Pinggera, Pnina Soffer, Stefan Zugal, Barbara Weber, Matthias Weidlich, Dirk Fahland, HajoA Reijers, and Jan Mendling. *Modeling Styles in Business Process Modeling*, volume 113 of *Lecture Notes in Business Information Processing*, chapter 11, pages 151–166. Springer Berlin Heidelberg, 2012.
18. Jose Raúl Romero, Juan Ignacio Jaén, and Antonio Vallecillo. Realizing correspondences in multi-viewpoint specifications. In *2009 IEEE International Enterprise Distributed Object Computing Conference*, pages 163–172.
19. Matthias Schrepfer. *Modeling Guidelines for Business Process Models*. Master thesis, 2010.
20. Antonio Vallecillo. *On the Combination of Domain Specific Modeling Languages Modelling Foundations and Applications*, volume 6138 of *Lecture Notes in Computer Science*, pages 305–320. Springer Berlin / Heidelberg, 2010.
21. Van Deursen Arie Visser, Eelco and Warmer Jos. Model-driven software evolution: A research agenda. In *In Proc. Int. Ws on Model-Driven Software Evolution held with the ECSMR07*, 2007.
22. Roel Wieringa. Design science as nested problem solving. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology*, DESRIST '09, pages 8:1–8:12, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-408-9.