Describing the correlations between metamodels and transformations aspects

Juri Di Rocco, Davide Di Ruscio, Ludovico Iovino, Alfonso Pierantonio

Department of Information Engineering Computer Science and Mathematics
University of LAquila, Italy,
`name.lastname{@univaq.it}`

Abstract. Metamodels are a key concept in Model-Driven Engineering. Any artifact in a modeling ecosystem has to be defined in accordance to a metamodel prescribing its main qualities. One of the most important artifact is model transformation that are considered to be the heart and soul of MDE and as such advanced techniques and tools are needed for supporting the development, quality assurance, maintenance, and evolution of model transformations. Several works propose the adoption of metrics to measure quality attributes of transformation without considering any metamodel aspects. In this paper, we present an approach to understand structural characteristics of metamodels and how the model transformations depend on corresponding input and target metamodels.

Keywords: Model Driven Engineering, metamodeling, metamodel metrics, transformation metrics

1 Introduction

Metamodels are a key concept in Model-Driven Engineering [22]. Almost any artifact in a modeling ecosystem [13] has to be defined in accordance to a metamodel, which represents an ontological description of application domains [10]. Metamodels are important because they formally define the modeling primitives used in modeling activities and represent the *trait-d'union* among all constituent components. One of this components are model transformations (MT), in fact MT play a key role since they permit to bridge different abstraction levels by automatically mapping source models to target ones. In [23] model transformations are considered to be the "heart" and "soul" of MDE and as such they require to be treated in a similar way as traditional software artifacts [2]. Understanding common characteristics of metamodels, how they evolve over time, and what is the impact of metamodel changes throughout the modeling ecosystem is key to success. Several approaches have been already proposed to analyse models [20] and transformations [3,28] with the aim of assessing quality attributes, such as understandability, reusability, and extendibility [7]. Similarly, there is the need for techniques to analyse metamodels as well in order to evalutate their structural characteristics and the impact they might have during the whole metamodel lifecycle especially in case of metamodel evolutions. To this end, some works propose the adoption of metrics for analysing metamodels [17,19] and transformation [28] as typically done in software development by means of object-oriented measurements [16]. Starting from our previous work [11], we are interested in better understanding metamodel characteristics and how metamodels and transformations are correlated by investigating the correlations of different metrics applied on a corpus of more than 450 metamodels and 90 transformations. On one hand we propose an approach for *a)* measuring certain metamodeling aspects (e.g., abstraction, inheritance, and composition) that modelers typically use; and *b)* for revealing what are the common characteristics in metamodeling that can increase the complexity of metamodels hampering their adoption and evolution in modeling ecosystems [13]. On other hand we propose an approach for identifying how the transformations are correlated to metamodels. The identified correlations permit to draw interesting considerations e.g. how a model transformation is typically structured depending on the considered metamodels, and how does the complexity of metamodels has an impact on the overall model transformations development. Such considerations can be preparatory to further analysis that are very common in software development [9], e.g., estimating the effort required to develop model transformations by considering the structural characteristics of the source and target metamodels.

The paper is structured as follows: Section 2 describes the process we have conceived and applied to analyze metamodels. Interesting correlations are discussed in Section 3. Section 4 discusses related work and Section 5 concludes the paper and draws some research perspectives.

2 The correlation among metamodels and transformations

Software metrics have been proposed to assess and predict software effort and quality [15] and recent research has proposed the adoption of metrics to measure transformations. In particular, metrics on transformations have been investigated [28,3] to support the measurements of model transformations with the aim of understanding transformations via quantitative evaluations. For instance, in [28] specific metrics have been conceived to measure ATL transformations, and in [4] authors define the meaning of several quality attributes in the context of model transformations and align them to a set of metrics.

The adoption of metrics to measure metamodels has been recently proposed in [17,19,12]. In particular, in [17] authors apply object-oriented measurements to understand common structural characteristics of metamodels, whereas [19] proposes a measuring mechanism for assessing the quality of metamodels. To the best of our knowledge, none of the existing approaches calculate transformation metrics with the aim of correlating them.

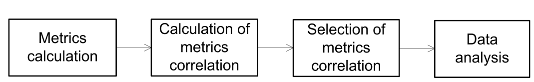

Fig. 1. Overview of the process for metamodel analysis

Since it is reasonable to claim that the complexity of model transformations is somehow related to that of the source and target metamodels, in our opinion in order to have a complete measurement of model transformations, it is necessary to identify also possible correlations between transformation and metamodel metrics e.g., to figure out at what extent the number of matched rules of given ATL transformation depends on the number of metaclasses in the source and/or target metamodels.

To this end, in this section the measurement process shown in Fig. 1 is presented. In particular, the first step of the process consists in applying a number of metrics on a representative corpus of transformations and corresponding metamodels. Afterwards the calculated metamodel and transformation metrics are correlated among them by using statistical tools. Finally, the collected data are analysed in order to cross/link structural characteristics of transformations and metamodels, e.g., how the different kinds of ATL rules (i.e., matched, lazy, and called) are typically used. It is important to remark that in the analysis step, metamodel metrics are also considered in order to identify possible correlations among transformation and metamodel metrics (e.g., how the number of metaclasses in the target metamodel impacts the structural characteristics of transformations in terms of number of matched rules, helpers, etc.). In [12], we describe the process, shown in 1, we have applied to identify linked structural characteristics and to understand how they might change depending on the nature of metamodels. In this work we have extended this process in order to calculate different set of metrics from different artifacts (metamodels and transformations) and to understand how the model transformations are dependent from corresponding input and target metamodels.

2.1 The proposed measurement process

The first step of the proposed process consists of the application of metrics on a data set of metamodels and transformations. Concerning the applied metrics on metamodels we borrowed those in [17] and added new ones by leading to a set of 28 metrics. Due to space limitations, in the rest of the paper we consider only the metrics shown in Tab. 1 for metamodels and shown in Tab. 2 form transformation. The corpus of the analyzed metamodels and transformations has been obtained by retrieving artifacts from different repositories, i.e., EMFText Zoo [6], ATLZoo [5], Github, and GoogleCode. To perform such analysis we have automatize the process for metrics calculation using a eterogenous repository called MDEForge presented in [8]. The calculated data are exported in CSV files encoding the values of all the calculated metrics. Generating CSV files enables the adoption of statistical tools like IBM SPSS, Microsoft Excel, R and Libreoffice Calc for subsequent analysis of the generated data.

2.2 Calculation and selection of metrics correlations

Correlation is probably the most widely used statistical method to detect cross-links and assess potential relationships among observed data. There are different techniques and indexes to discover and measure correlations. In the following we overview the Pearson's and Spearman's coefficients that we have considered in this paper to measure the correlations among calculated metamamodel metrics.

The *Pearson's correlation coefficient* [18] was developed by Karl Pearson from a related idea introduced by Francis Galton in the 1880s. It is widely used in the sciences as a measure of the degree of linear dependence between two variables. In particular, the Pearson correlation coefficient is appropriate when it is possible to draw a regression line between the points of the available data (e.g., see the diagrams A and B in Fig. 2).

The *Spearman's correlation coefficient* [24] was used by Charles Spearman in the 1900s in the psychology domain. This coefficient is better than Pearson to manage situations when there is a monotonic relationship between the considered variables. For instance, in the cases shown in the diagrams C and D in Fig. 2, the Pearson coefficient would wrongly identify a very low correlations among the considered data. This is due to the fact that the assumption of linear relationships required by Pearson is not satisfied. Contrariwise, Spearman's correlation index would perform better in cases of monotonic relationships as in the diagrams C and D in Fig. 2

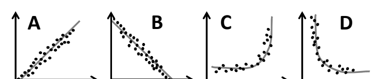

Fig. 2. Examples of scattered plots

It is also important to note that the assumption of a monotonic relationship is less restrictive than a linear relationship (an assumption that has to be met by the Pearson correlation). For this reason, we use Spearman only for highlighting curvilinear correlations. Finally, both Pearson's and Spearman's correlation indexes assume values in the range of -1.00 (perfect negative correlation) and +1.00 (perfect positive correlation). A correlation with value 0 indicates that there is no correlation between two variables. In order to assess the strength of correlations it is possible to consider the guide that Evans [14] suggests for the *absolute value* of the correlation indexes, i.e., [*0.0*,*0.19*] very weak, [*0.20*,*0.39*] weak, [*0.40*,*0.59*] moderate, [*0.60*,*0.79*] strong, and [*0.80*,*1.0*] very strong.

Metamodel metrics correlations Once the metamodel metrics have been calculated, the most correlated ones are identified and selected. In particular, we have calculated the Pearson's correlation indexes for all the values of the metamodel metrics. The outcome of this operation is a correlation matrix as the one shown in Fig 3. The discussion is based on the correlation matrix shown in Fig 3 and by considering the most interesting correlations having value greater than 0.60 (thus strong or even very strong). Because of lack of space it is not possible to discuss all the identified correlations that include the metrics shown in Table 1 and 2. However, interested readers can refer to the spreadsheet available online[1] containing all the obtained results. For instance, the number of MC[2] (number of metaclasses) is strongly correlated with the number of CMC

[1] http://www.di.univaq.it/ludovico.iovino/data-mise2015.html
[2] For the complete list of acronyms in the table we refer to [11]

(number of concrete metaclasses) as testified by their Pearson's correlation index having value 0.997.

	#MC	#AMC	#CMC	#IFLMC	#SF	#ASF	#TCWS	#MGHL	#MHS	LNS
#MC										
#AMC	0.451									
#CMC	0.997	0.377								
#IFLMC	0.874	0.139	0.894							
#SF	0.831	0.574	0.810	0.488						
#ASF	-0.102	-0.064	-0.100	-0.176	0.155					
#TCWS	0.993	0.451	0.990	0.890	0.797	-0.131				
#MGHL	0.666	0.637	0.633	0.534	0.558	-0.216	0.678			
#MHS	0.704	0.463	0.688	0.562	0.620	-0.164	0.704	0.561		
LNS	-0.082	-0.055	-0.080	-0.030	-0.108	-0,181	-0.072	-0.100	-0.094	

Fig. 3. Pearson Correlation values related to metamodel metrics

Model transformation and metamodel metric correlations The interesting part of our analysis relies on correlating model transformation and metamodel metrics. To this end a correlation matrix based on the Spearman's index has been calculated and a fragment is shown in Fig 4. The matrix relates model transformation metrics with metrics calculated on the corresponding source and target metamodels. For instance, according to the calculated matrix, the number of output patterns (OP) of a model transformation is strongly related with the number of metaclasses (MC) contained in the output metamodel.

	B	IP	OP	TR	MR	LR	CR	RWF	RWD	H	HWC	HNC	CRT	
MC	0.450	**0.690**	0.467	0.452	0.402	0.295	0.248	0.267	0.329	-0.002	-0.082	0.168	0.088	INPUT
AMC	0.340	0.463	0.339	0.412	0.374	0.264	0.228	0.390	0.306	0.083	-0.019	0.229	-0.003	
CMC	0.478	0.504	0.496	0.468	0.412	0.290	0.289	0.260	0.360	0.036	-0.040	0.178	0.098	
SF	0.503	0.394	0.467	0.363	0.334	0.208	0.282	0.126	0.315	-0.037	-0.138	0.139	0.051	
MC	0.520	0.542	**0.783**	**0.746**	0.500	0.223	0.369	0.480	0.399	0.180	0.168	0.204	0.131	OUTPUT
AMC	0.478	0.504	0.496	0.468	0.412	0.290	0.289	0.260	0.360	0.036	-0.040	0.178	0.098	
CMC	0.503	0.394	0.467	0.363	0.334	0.208	0.282	0.126	0.315	-0.037	-0.138	0.139	0.051	
SF	**0.808**	0.506	0.505	0.481	0.451	0.202	0.266	0.375	0.284	-0.008	-0.075	0.100	-0.014	

Fig. 4. Spearman Correlation values related to transformation and metamodel metrics

3 Data analysis

In this section we discuss some relevant correlations we have identified as described in the previous section. In particular, by considering some of the identified transformation metrics, it is possible to draw interesting considerations about how the constructs of the ATL language are typically used by developers.

Moreover, by considering the correlations of both transformation and metamodel metrics (see Section 3.2), further considerations can be drawn about how structural characteristics of metamodels affect the structure of the corresponding model transformations.

3.1 Metamodels correlation analysis

In this section we briefly present the most representative metrics and correlations we have discovered in this process. We present the metrics correlation discussing the meaning and highlighting the results in the graphical representation.

How the number of metaclasses is related to the adoption of abstraction constructs In this section we discuss how the size of metamodels expressed in terms of number of metaclasses is related to the adoption of abstraction constructs, i.e., abstract metaclasses, and supertypes.

In particular, as shown in Fig. 5 the number of metaclasses (MC) and the number of those with super types (MCWS) are strongly correlated (with Pearson index 0.99). More specifically, when the number of metaclasses grows, typically also the number of classes with supertypes increases. In other words, as expected, the adoption of inheritance is proportional to the size of metamodels expressed in terms of number of metaclasses. Interestingly, metamodel designers prefer to add siblings in hierarchies instead of adding new hierarchy levels. This is testified by Fig. 5 that shows the values of the *MHS* (Max Hierarchy Sibling) and *MGHL* (Max generalization hierarchical level) metrics. Such conclusions are confirmed by the Pearson correlation indexes between *MC* and *MHS* (0.70) and the one between *MC* and *MGHL* (0.66). Finally, Fig. 5 reveals that in metamodels with at most 50 metaclasses, *i)* the number of supertypes in hierarchy is in between 0 and 20, *ii)* the number of siblings in a hierarchy is in between 0 and 10, and *iii)* the maximum height of a hierarchy is in between 0 and 5. These data represent a pattern charactering the typical typical metamodel definition.

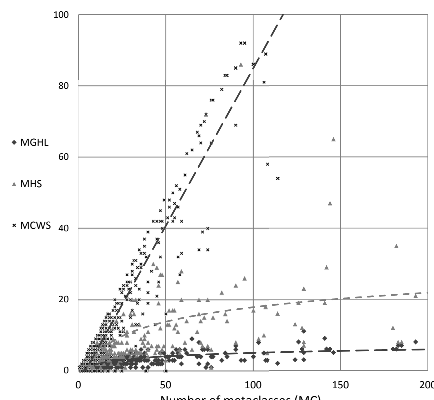

Fig. 5. Analyzing metamodel abstraction level

How structural features are used with hierarchies This section aims at comprehend how structural features are used in presence of class hierarchies. To this end, we can consider the average number of features (ASF) and the total number of metaclasses with supertypes (MCWS) metrics. Even though the correlation index of these two metrics is low, according to the matrix in Fig 3, the Spearman approach permits to identify a greater correlation index. As shown in Fig. 6 it is evident that increasing the number of metaclasses with supertypes, the average number of structural features in a metaclass decreases. Moreover, an interesting statistical result obtained by considering the correlation between the MC and ASF metrics is that by considering metamodels having the number of metaclasses in the range between 1 and 50 , the average number of features (excluding the inherited ones) of a metaclass ranges between 1 and 5.

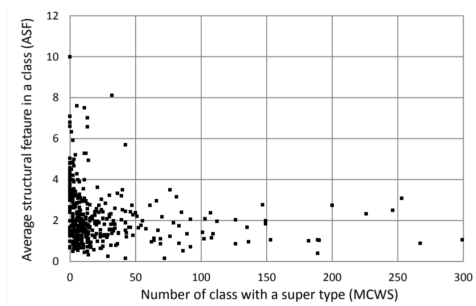

Fig. 6. Analyzing structural features introduction in hierarchies

How the number of featureless metaclasses is related to hierarchies height The correlation between the number of metaclasses with supertypes (MCWS) and the number of concrete metaclasses without features (IFLMC) is interesting for understanding how specializations of metaclasses can introduce or reduce structural features in metamodels. To this end, MCWS and IFLMC are strongly correlated as supported by the Pearson's index having value 0.890. The effect of such correlation is shown in Fig. 7[3]. In particular, by increasing the number of metaclasses with super types, the number of metaclasses without attributes or references increases too. This means that when hierarchies are introduced, usually existing features are subject to refactoring operations. Usually, what is done is to move them to super classes and to create leaves in the hierarchies inheriting features from the super types. This is in line with the typical usage of hierarchies for factorizing common aspects in superclasses.

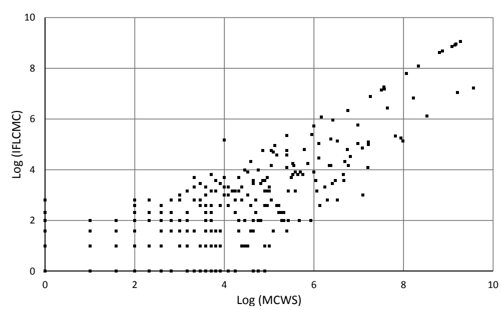

Fig. 7. Analyzing hierarchical height and featureless metaclasses

3.2 How metamodel characteristics affect model transformations

By exploiting the matrix obtained by correlating transformation and metamodel metrics, in this section we discuss how metamodels affect the development of

[3] This scattered plot diagram use date logarithmic scale for empathize the correlation

model transformations. The discussion is based on the correlation matrix shown in Fig. 4 and by considering the most interesting correlations having value greater than 0.65.

How transformation rules are influenced by target metamodels This aspect can be investigated by considering the correlation between the number of metaclasses in the target metamodel (OUT MC) and the number of TR (Transformation Rules). Such two values are correlated because of the Spearman's index having value 0.746. The graph in Fig 8[4] represents how these two values are influenced by each other in our corpus. According to the graph it is evident that increasing the number of the MC in the target metamodel the number of TR increases too. This is generally true, since the transformation writing is output-driven when the developer tries to cover all the metaclasses of the target metamodel. We can also state that the common concentration in the corpus is in the range between 1 and 20 metaclasses and 1 and 15 transformation rules, again confirming the declarative style of transformation as common choice of the developers.

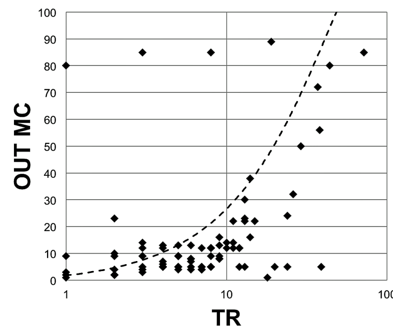

Fig. 8. How TR are influence by number of MC in target metamodel

How the structural features in the target metamodel influence the number of bindings According to the calculated Spearman correlation, the structural features (SF) of the target metamodel can influence the number of bindings (B) written in the rules of the transformations. The plot in Fig 9[4] shows that increasing the value of SF in the output metamodel (OUT SF), the number of binding grows too. The distribution is common for the number of SF between 0 and 20 distributed for the value of B that goes from 1 to about 75.

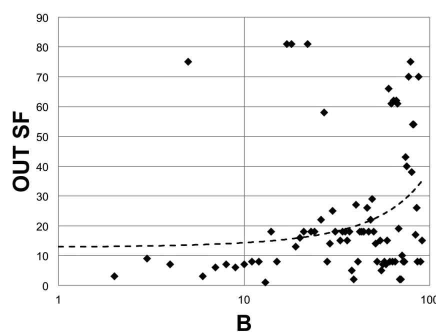

Fig. 9. How the SF in the target metamodel influence the number of B

[4] The scattered plot diagram use date logarithmic scale for empathize the correlation

How the total number of output patterns are influenced by the target metamodels According to the calculated matrix the Spearman's correlationindex between the value of OP (Output Patterns) in the rules and the number of metaclasses in the target metamodels has value 0.783. This correlation occurring in our corpus is depicted in Fig 10[4] where the value of OP in the rules of our transformations increases at the raising of the value of MC in the target metamodels. The most dense concentration is in the range of 1-10 output patterns and 1-10 metaclasses in output.

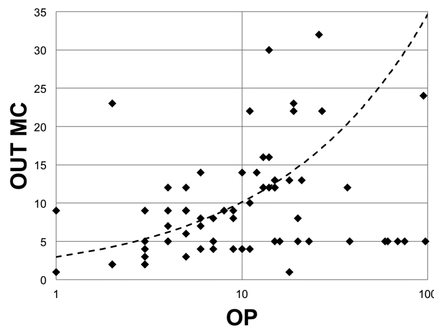

Fig. 10. How OP are influenced by the target metamodels

How the total number of input pattern are influenced by the source metamodels As anticipated in the previous sections the IP (Input Pattern) of the transformations are related to the value of MC in the source metamodel (IN MC). This is confirmed by the Spearman's correlation that results 0.692. In the graph in Fig 11[4] the distribution is less clear than the previous case but the trend is similar: increasing the value of MC in input, the value of IP increases too. This again confirms the use of declarative style as the preferred one in our corpus.

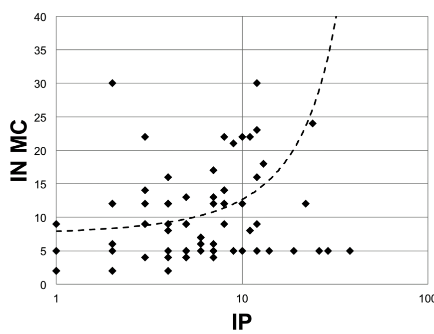

Fig. 11. How IP are influenced by the source metamodels

4 Related works

In [28] the authors introduces metrics to measure ATL transformations and the adoption of metrics to measure quality attributes of transformation without considering any metamodel aspects. In other approaches the main topic is the quality attribute driven by the metric [4], for example making the quality of model transformations measurable. In [25] the authors have focused on transformation model measurements in order to better understand transformations via a quantitative evaluation, like the declarative factor of modules and rules. In [27] an analogous approach for measuring model repositories is shown, simply considering models in the evaluation. The authors in [26] investigate factors having impact on the execution performance of model transformations and they extracted metrics for the analysis. Van Amstel et al. propose a set of six quality attributes to evaluate the quality of model transformations [1]. All cited works propose the adoption of metrics to measure quality attributes of transformation without considering any metamodel aspects. The authors of [21] worked on how model transformations can improve the quality of models using metrics. A similar approach for understanding structural characteristics of metamodels and their relationships has been presented in [11]. Williams et al. in [17] is the first one to discuss metrics related to a large metamodel collection exposing how metamodels are commonly structured, and how they evolve over time.

5 Conclusions and future work

In this paper, we proposed a number of metrics which can be used to acquire objective, transparent, and reproducible measurements of metamodels and transformations. The first goal is to better understand the main characteristic of metamodels, how they are coupled, and how they change depending on the metamodel structure. We have also proposed an approach to analyze model transformations by considering also the corresponding metamodels. The approach relies on the correlation of different metrics and has been applied on a corpus of 450 metamodels and 90 transformations and permitted to draw interesting considerations that we intend to extend in the future.

References

1. van Amstel, M.F., Lange, C.F., van den Brand, M.G.: Using metrics for assessing the quality of asf+ sdf model transformations. In: Theory and Practice of Model Transformations, pp. 239–248. Springer (2009)
2. van Amstel, M., van den Brand, M.: Model transformation analysis: Staying ahead of the maintenance nightmare. In: ICMT 2011, LNCS, vol. 6707, pp. 108–122. Springer (2011)
3. van Amstel, M., van den Brand, M.: Quality assessment of atl model transformations using metrics. Proceedings of the 2nd International Workshop on Model Transformation with ATL (MtATL 2010), Malaga, Spain (June 2010) (2010)
4. van Amstel, M., Lange, C., van den Brand, M.: Metrics for analyzing the quality of model transformations. Proceedings of the 12th ECOOP Workshop on Quantitative Approaches on Object Oriented Software Engineering pp. 41–51 (2008)
5. ATLAS Group: ATL Transformations Zoo. http://www.eclipse.org/m2m /atl/atlTransformations/
6. ATLAS Group: EMFTEXT Concrete Syntaxes Zoo. http://emftext.org/index.php/EMFText_Concrete_Syntax_Zoo$
7. Bansiya, J., Davis, C.G.: A hierarchical model for object-oriented design quality assessment. Software Engineering, IEEE Transactions on 28(1), 4–17 (2002)
8. Basciani, F., Di Rocco, J., Di Ruscio, D., Di Salle, A., Iovino, L., Pierantonio, A.: Mdeforge: an extensible web-based modeling platform. pp. 66–75 (2014), http://ceur-ws.org/Vol-1242/paper10.pdf
9. Boehm, B., Abts, C., Chulani, S.: Software development cost estimation approaches a survey. Annals of Software Engineering 10(1-4), 177–205 (2000), http://dx.doi.org/10.1023/A%3A1018991717352

10. Chandrasekaran, B., Josephson, J., Benjamins, V.: What are ontologies, and why do we need them? Intelligent Systems and their Applications, IEEE 14(1), 20–26 (1999)
11. Di Rocco, J., Di Ruscio, D., Iovino, L., Pierantonio, A.: Mining metrics for understanding metamodel characteristics. In: Proceedings of the 6th International Workshop on Modeling in Software Engineering. pp. 55–60. MiSE 2014, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2593770.2593774
12. Di Rocco, J., Di Ruscio, D., Iovino, L., Pierantonio, A.: Mining metrics for understanding metamodel characteristics. In: MiSE. pp. 55–60 (2014)
13. Di Ruscio, D., Iovino, L., Pierantonio, A.: Evolutionary togetherness: how to manage coupled evolution in metamodeling ecosystems. In: Intl. Conf. on Graph Transformations (ICGT 2012). LNCS, vol. 7562. Springer (2012)
14. Evans, J.S.B., Over, D.E.: Rationality and reasoning. Psychology Press (2013)
15. Fenton, N.E., Pfleeger, S.L.: Software Metrics: A Rigorous and Practical Approach. PWS Publishing Co., Boston, MA, USA, 2nd edn. (1998)
16. Harrison, R., Counsell, S., Nithi, R.: An evaluation of the mood set of object-oriented software metrics. IEEE Transactions on Software Engineering 24, 491–496 (1998)
17. James, W., Athansios, Z., Nicholas, M., Louis, R., Dimitios, K., Richard, P., Fiona, P.: What do metamodels really look like? Frontiers of Computer Science (2013)
18. Lee Rodgers, J., Nicewander, W.A.: Thirteen ways to look at the correlation coefficient. The American Statistician 42(1), 59–66 (1988)
19. Ma, Z., He, X., Liu, C.: Assessing the quality of metamodels. Frontiers of Computer Science 7(4), 558–570 (2013), http://dx.doi.org/10.1007/s11704-013-1151-5
20. Monperrus, M., Jezequel, J.M., Champeau, J., Hoeltzener, B.: Model-Driven Software Development. IGI Global (Aug 2008), http://www.igi-global.com/chapter/measuring-models/26829/
21. Saeki, M., Kaiya, H.: Measuring model transformation in model driven development. In: CAiSE Forum. vol. 247 (2007)
22. Schmidt, D.: Guest Editor's Introduction: Model-Driven Engineering. Computer 39(2), 25–31 (2006)
23. Sendall, S., Kozaczynski, W.: Model transformation: The heart and soul of model-driven software development. IEEE Softw. 20(5), 42–45 (Sep 2003), http://dx.doi.org/10.1109/MS.2003.1231150
24. Spearman, C.: The proof and measurement of association between two things. The American journal of psychology 15(1), 72–101 (1904)
25. Tolosa, J.B., Sanjuán-Martínez, O., García-Díaz, V., Lovelle, J.M.C., et al.: Towards the systematic measurement of atl transformation models. Software: Practice and Experience 41(7), 789–815 (2011)
26. Van Amstel, M., Bosems, S., Kurtev, I., Pires, L.F.: Performance in model transformations: experiments with atl and qvt. In: Procs. ICMT2011, pp. 198–212. Springer (2011)
27. Vépa, E., Bézivin, J., Brunelière, H., Jouault, F.: Measuring model repositories. In: Proceedings of MSM06 (2006)
28. Vignaga, A.: Metrics for measuring atl model transformations. MaTE, Department of Computer Science, Universidad de Chile, Tech. Rep (2009)

6 Appendix

Acronym	Name	Description
AMC	Number of abstract MetaClass	Number of metaclasses that cannot be instantiated in models
ASF	Average Structural Features	Average number of attributes and references in a metaclass
CMC	Number of concrete MetaClass	Number of metaclasses that can be directly instantiated
IFLMC	Number of concrete Immediately Featureless MetaClass	The number of concrete metaclasses that have no attributes or references, but may inherit features from a superclass
LNS	Isolated metaclasses	It is the percentage of metaclasses that are not connected with any other one
MC	Number of total MetaClass	Number of metaclasses in the metamodel (MC = AMC + CMC)
MCWS	Number of class with a super type	Number of metaclasses having at least one super type
MGHL	Maximum generalization hierarchical level	Maximum hierarchical depth in the metamodel
MHS	Max Hierarchy Sibling	Maximun number of classes inheriting from a generic superclass
SF	Number of structural features	Number of attributes and references in the metamodel

Table 1. Some of the used metrics for measuring metamodels

Acronym	Name	Description
B	Number of bindings	Number of bindings in all output pattern
IP	Number of Input Pattern	The metric number of input pattern elements measure the size of the input pattern of rules. Note that since called rules do not have an input pattern, the metric number of input model elements does not include called rules.
OP	Number of Output Pattern	The metric number of output pattern elements measure the size of the output pattern of rules.
TR	Number of Transformation Rules	A measure for the size of a model transformation is the number of transformation rules it encompasses. In ATL, there are different types of rules, viz., matched rules, lazy matched rules, unique lazy matched rules, and called rules.
MR	Number of Matched Rules (Excluding Lazy Matched Rules)	Number of matchad rule exzcluding lazy matched rule. If this matrics are equals to number of transformation rule the transformation are defined *completely declarative*
LR	Number of Lazy Matched Rules (Including Unique)	Number of lazy rule including unique
CR	Number of Called Rules	Number of Called Rules
RWF	Number of Rules with a Filter Condition on the Input Pattern	Number of rules with a filter condition on the input pattern. The input pattern has a condition. This implies that not all model elements in the source model may be transformed.
RWD	Number of Rules with a do Section	ATL allows the definition of imperative code in rules in a do block. This can be used to perform calculations that do not fit the preferred declarative style of programming. To measure the use of imperative code in a transformation, we defined number of rules with a do section
RWU	Number of Rules with a using clause	ATL allows the definition of local variable in a rule. This can be used to perform calculations that do not fit the preferred declarative style of programming. To measure the use of imperative code in a transformation, we defined number of rules with a using clause
H	Number of Helper	Number of total helper in the transformation
HWC	Number of Helpers with Context	Number of helper with context in the transformation
HNC	Number of Helpers without Context	Number of helper without context in the transformation
CRT	Number of Calls to resolveTemp()	The resolveTemp() function is used to look-up references to non-default output elements of other rules. Therefore, it is to be expected that model transformations with a large number of calls to the resolveTemp() function are harder to understand.

Table 2. Some of the used metrics for measuring transformations