An Empirical Study on the Perception of Metamodel Quality

Georg Hinkel[1], Max Kramer[2], Erik Burger[2], Misha Strittmatter[2] and Lucia Happe[2]
[1] *Forschungszentrum Informatik (FZI), Haid-und-Neu-Straße 10-14, Karlsruhe, Germany*
[2] *Karlsruhe Institute of Technology (KIT), Am Fasanengarten 5, Karlsruhe, Germany*

Keywords: Metamodel, Perception, Empirical Study.

Abstract: Despite the crucial importance of metamodeling for Model- Driven Engineering (MDE), there is still little discussion about the quality of metamodel design and its consequences in model-driven development processes. Presumably, the quality of metamodel design strongly affects the models and transformations that conform to these metamodels. However, so far surprisingly few work has been done to validate the characterization of metamodel quality. A proper characterization is essential to automate quality improvements for metamodels such as metamodel refactorings. In this paper, we present an empirical study to sharpen the understanding of the perception of metamodel quality. In the study, 24 participants created metamodels of two different domains and evaluated the metamodels in a peer review process according to an evaluation sheet. The results show that the perceived quality was mainly driven by the metamodels completeness, correctness and modularity while other quality attributes could be neglected.

1 INTRODUCTION

In Model-Driven Engineering (MDE), models are not only used to document the design of a system, but also to perform analyses, to run simulations, or to generate implementation code. To support such automated transformations, individual model instances have to conform to a metamodel that is expressed using a meta-modeling language. All textual and visual editors for models and all transformations to other models or code depend directly or indirectly on this metamodel. As a result, all development steps that involve the usage or development of an editor or transformation are affected by the metamodel. Thus, metamodels are central artifacts of many MDE processes and tools.

The influence of metamodels on MDE processes and artifacts, as well as the quality of metamodels, has, however, not been studied and discussed enough in the community. There is no clear notion of metamodel quality in literature. Software quality standards, such as ISO/IEC 25010, are only partly applicable since these standards are intended for complete software products and systems, but not for development artifacts such as metamodels. The quality characteristics of these standards can only be used to clarify which quality aspects of a metamodel are considered, but they do not explain how. Furthermore, they cannot be used to determine how a general notion of metamodel quality is assembled from these quality characteristics. In order to agree on a common notion of quality for metamodels, and in order to eventually measure it, it is an important step to characterize how the quality of metamodels is perceived.

In literature, however, existing characterizations of metamodel quality, such as those from Bertoa et al. (Bertoa and Vallecillo, 2010), have not been validated on an empirical basis. Further tools exist that correct metamodel design flaws (López-Fernández et al., 2014), but they are only based on personal experience and expertise, without a proper validation. In particular, it is not clear whether the improvement of some quality attributes will have an impact on the overall quality of a metamodel at all. Therefore, a validated characterization of metamodel quality is essential for automated quality improvements, such as refactorings.

The idea in this paper is that developers assess the metamodel quality based on all their experience with model-driven tools. Thus, the quality of the metamodels is not restricted to a particular application domain, such as particular analysis methods, model transformation, or instance creation. Rather, we try to assess the quality of metamodels as a whole.

This approach is beneficial especially in platform projects where concrete applications are not entirely clear at the beginning of the project. For example, the Neurorobotics-platform of the *Human Brain Project*

Hinkel, G., Kramer, M., Burger, E., Strittmatter, M. and Happe, L.
An Empirical Study on the Perception of Metamodel Quality.
DOI: 10.5220/0005632001450152
In *Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2016)*, pages 145-152
ISBN: 978-989-758-168-7
Copyright © 2016 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

(HBP) tries to support neuroscientists in integrating neuronal networks into robot controllers to provide validation methods for neuronal network models and models of the neurophysiology. First neurophysiology models have been created with the pragmatics to simulate them in the platform (Hinkel et al., 2015). However, we also want to encourage neuroscientists to analyze or automatically optimize these neurophysiology models and are thus looking for criteria that the (meta-)models have to fulfill. Moreover, the HBP is designed for a total duration of ten years. During this period of time, it is likely that the metamodel will degrade unless extra effort is spent for its refactorings (Lehman, 1974; Lehman et al., 1997). For such refactorings, it is important to understand the facets of metamodel quality.

In this paper, we present the design and results of an empirical study on the characterization of metamodel quality perception: 24 professional developers and students assessed the overall quality and individual quality characterstics by completing 89 copies of a questionnaire. We performed correlation tests and show that out of 10 individual characteristics only modularity, completeness, and correctness had a significant influence on the perception of overall metamodel quality within our study. Our empirical study provides insights on how developers perceive the quality of metamodels. All results and supporting material such as the questionnaires are available online[1].

The remainder of this paper is structured as follows: Section 2 explains the experiment setup. Section 3 presents the results from our empirical study. Section 4 discusses threats to validity. Section 5 explains our plans for future work. Finally, Section 6 discusses related work before Section 7 concludes the paper.

2 EXPERIMENT SETUP

We performed a controlled experiment to obtain manual assessments of metamodel quality. In order to eliminate unwanted effects of metamodel properties that may influence the quality assessment but that are not important when measuring metamodel quality, the participants also created the metamodels to be judged in this controlled setting. The 24 participants created metamodels for two domains. Each domain was described in a text and the participants were asked to design a metamodel according to it. The participants consisted of professional researchers as well as students from a practical course on MDE. They were randomly assigned to the domains, ensuring a balance between the domains.

The first domain concerned user interfaces of mobile applications. Participants were asked to create a metamodel that would be able to capture designs of the user interface of mobile applications so that these user interface descriptions could later be used platform-independently. The participants created the metamodel according to a domain description in natural language from scratch. We refer to creating the metamodel of this mobile applications domain as the Mobiles scenario.

The second domain was business process modeling. Here, the participants were given a truncated metamodel of the Business Process Modeling Language and Notation (BPMN) (The Object Management Group, 2011) where the packages containing conversations and collaborations had been removed. The original metamodel can be found in the BPMN packages of the Eclipse project. The task for the participants was to reproduce the missing part of the metamodel according to a textual description of the requirements for the collaborations and conversations. We refer to the task of completing this metamodel as the BPMN scenario.

After creating the metamodel, all participants were given several metamodels for the same domain that were created by other participants and rated their quality according to a questionnaire. In a next step, the participants were given the textual description of the domain that they were not creating a metamodel for and evaluated the quality of several metamodels for that scenario.

Given the limited time-box of the experiment, not all of the participants returned an equal amount of filled questionnaires.

The questionnaire asked the participants to evaluate the perceived overall quality of the metamodels as well as evaluations on the following quality attributes: complexity, understandability, modularity, conciseness, completeness, correctness, changability, consistency, instance creation and transformation creation, adapted from Bertoa et al. (Bertoa and Vallecillo, 2010). The participants were asked to rate the quality attributes on a six level Likert-scale.

3 RESULTS

We have received 89 responses for our evaluation sheet out of which 46 are assessing the quality of 11 metamodels in the Mobiles domain and 43 are assess-

[1] https://sdqweb.ipd.kit.edu/wiki/Metamodel_Quality sdqweb.ipd.kit.edu/wiki/Metamodel_Quality

ing the quality of 12 metamodels in the BPMN domain. All evaluation sheets assess the overall quality of the metamodels as well as the quality attributes we mentioned in Section 2. The assessment could be done in the grades very bad (-5), bad (-3), slightly bad (-1), slightly good (+1), good (+3) and very good (+5). For Complexity, a good Complexity means that the metamodel is simple, i.e. perceived as not complex. The quality attributes were briefly described in the questionnaire, though understandability was introduced as the degree in which a metamodel is self-describing.

We analyzed the correlations between the quality attributes and performed an analysis of variance (ANOVA) on the questionnaire responses. We did this for each of the scenarios separately and once for both of them combined, adding the scenario itself as another influence factor. Unfortunately, not all of the questionnaires were complete, so we had to omit about half of the questionnaires for the ANOVA due to some missing entries.

In the remainder of this section, we present the results for the correlation analysis in order to reason about the validity of our results and on future study designs in Section 3.1. Afterwards, we present our findings from the Mobiles scenario in Section 3.2, from the BPMN scenario in Section 3.3 and present the results of an ANOVA spanning over both domains for a generalization of the results in Section 3.4.

3.1 Correlation between Quality Attributes

The responses that we got from our experiment allow us to analyze how well the quality attributes describe the metamodel quality in our experiment in general. Therefore, we printed the correlations of the quality attributes to each other in Table 1. The table shows Pearson correlation coefficients, but as the Likert scale is equidistant, this coincides with the Spearman correlation coefficient. The analysis of these correlations allow us to reason on the design of future empirical studies with regard to the quality attributes that we are asking from participants. Ideally, the quality attributes should be orthogonal to each other, each describing different aspects of the metamodel with no correlation to other aspects. Thus, Table 1 should show ones on the diagonal and zeroes elsewhere. However, some correlations may always be introduced by random. But if the correlations between the quality attributes get too big, they invalidate the ANOVA in the subsequent sections. To highlight strong correlations ($\rho > 0.5$) between different quality attributes, we have printed them in bold.

The strongest correlation by far can be observed between completeness and correctness. This seems reasonable since completeness can be seen as a form of correctness. In fact, some studies treat completeness and correctness as one quality attribute. We can support this methodology for metamodels since these quality attributes have a very strong correlation ($\rho = 0.73$). On the other hand, understandability and conciseness also often go together but only showed a weaker correlation ($\rho = 0.56$). Here, we argue that these quality attributes are perceived significantly different and thus should not be recorded together.

Only three other strong correlations have been found within our experiment. Instance creation strongly correlates with both transformations and complexity. However, since this correlation is not as strong as the correlation between completeness and correctness and even more importantly, the creation of transformations does not correlate with the complexity, we argue that it is better to treat these two quality attributes separately. Furthermore, there is a correlation between complexity and understandability. This is also reasonable and supports the intuition that complex metamodels are hard to understand. However, this correlation is by far not as strong as the correlation between completeness and correctness and therefore have their right as separated quality attributes.

As a result, we argue that the quality attributes that we used for our experiment to evaluate aspects of the metamodel quality are valid except that completeness and correctness should have been treated as a single quality attribute. We will use this insight in the design of future experiments.

3.2 Important Quality Attributes in the Mobiles Scenario

In the Mobiles scenario, the second most important quality attribute was the completeness of the metamodel ($p = 0.00013$). This may be caused by the fact that the Mobile domain as presented to the participants apparently seemed very complex and thus many metamodels were perceived as incomplete. It seems surprising that the influence of a metamodels correctness is not even significant on the 10%-level but this may be due to the fact that completeness and correctness of the metamodels were strongly correlated ($\rho = 0.83$) making completeness and correctness to some extend interchangeable.

A remarkable result is that the modularity has had a more significant influence on the overall perceived quality of the Mobiles metamodels in our experiment ($p = 8.25 \cdot 10^{-5}$). Conciseness, understandability and complexity also showed significant influences at the

Table 1: Correlations of Quality Attributes, strong correlations ($|\rho| > 0.5$) in bold.

	Complexity	**Understandability**	**Conciseness**	**Modularity**	**Consistency**	**Completeness**	**Correctness**	**Changability**	**Instance Creation**	**Transformations**
Complexity	(1.00)	0.39	**0.55**	-0.09	0.14	-0.14	-0.17	0.25	**0.57**	0.29
Understandability	0.39	(1.00)	**0.56**	0.34	0.24	0.13	0.19	0.19	0.45	0.10
Conciseness	**0.55**	**0.56**	(1.00)	0.15	0.24	0.14	0.20	0.24	0.36	0.25
Modularity	-0.09	0.34	0.15	(1.00)	0.32	0.29	0.30	0.21	0.07	0.08
Consistency	0.14	0.24	0.24	0.32	(1.00)	0.46	0.37	0.19	0.23	0.15
Completeness	-0.14	0.13	0.14	0.29	0.46	(1.00)	**0.73**	0.08	0.11	0.16
Correctness	-0.17	0.19	0.20	0.30	0.37	**0.73**	(1.00)	0.10	0.07	0.08
Changability	0.25	0.19	0.24	0.21	0.19	0.08	0.10	(1.00)	0.28	0.34
Instance Creation	**0.57**	0.45	0.36	0.07	0.23	0.11	0.07	0.28	(1.00)	**0.55**
Transformations	0.29	0.10	0.25	0.08	0.15	0.16	0.08	0.34	**0.55**	(1.00)

5%-level ($p = 0.02, 0.04, 0.04$) but these results do not withstand a Holm correction for multiple tests. We believe this is due to the fact that we had few valid questionnaires after having to cancel roughly half of them (22) because of missing values.

3.3 Important Quality Attributes in the BPMN Scenario

Unlike the Mobiles scenario, in the BPMN scenario the participants had to extend a metamodel for a given feature request. This task apparently was very difficult for many participants, as the correctness has by far the strongest significance ($p = 0.012$) of all quality attributes, followed by modularity ($p = 0.022$) and completeness ($p = 0.298$). None of these significances withstands a Holm-correction, again partially due to the low number of responses.

Comparing the results with the Mobiles scenario, we can see much less significances and higher p-values, indicating that the perception of the quality attributes is less clear. This may be due to the complexity of the original BPMN metamodel.

3.4 Important Quality Attributes in both Scenarios Combined

Taking the questionnaires of both experiment scenarios together, we can get results that are to some extend independent from the scenario. In total, we had to omit 44 responses in the ANOVA due to missing entries. The resulting ANOVA table is depicted in Table 2. The table shows the F statistic and the p-value for the test that the underlying linear model is not degraded when the quality attribute is omitted. As we perform multiple tests, we apply a Holm correction on the resulting p-values. We used the usual significance codes, so $***$ stands for a significance $p < 0.001$, $**$ for $0.001 < p < 0.01$, $*$ for $0.01 < p < 0.05$ and $\cdot$ for $0.05 < p < 0.10$ but apply them to the already corrected p-values.

We have taken into account the scenario as an additional factor in order to analyze whether the results can be generalized from the scenario. The most important result from this analysis is that the influence of the scenario is insignificant ($p = 0.70$). This means, although the scenario of both of the experiments were different in both domain and type (metamodel from the scratch or extension to an existing one), the influence factors of the quality attributes are much stronger for the perception of the overall quality than the scenario. This indicates that we can generalize the findings from this analysis of variance in some degree to metamodels of any domain.

According to these results, the most important quality attributes of a metamodel are its completeness and correctness. These attributes also have a strong correlation ($\rho = 0.73$) indicating that complete metamodels are often also correct and vice versa. However, this result is hardly surprising.

Despite it is the metamodels most common application, the instance creation is not among the most significant quality attributes but does not appear as significant. We believe this is a consequence from the fact that instance creation was not correlated to completeness or correctness in our experiment. Thus, the instance creation is inflated by the fact that it is very easy to create an instance of an overly simplistic metamodel.

An interesting result is the strong influence of

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Table 2: Results of the ANOVA for both scenarios combined.

	F-statistic	p-value	p-value corrected	
Complexity	2.711	0.11	0.74	
Understandability	8.21	0.0072	0.058	$\cdot$
Modularity	34.82	$1.29 \cdot 10^{-6}$	$1.29 \cdot 10^{-5}$	$***$
Conciseness	1.040	0.32	1.00	
Completeness	36.39	$8.78 \cdot 10^{-7}$	$9.66 \cdot 10^{-6}$	$***$
Correctness	11.60	0.0018	0.016	$*$
Changability	1.77	0.19	1.00	
Consistency	0.60	0.45	1.00	
Instance Creation	0.09	0.77	1.00	
Transformations	0.27	0.61	1.00	
Scenario	0.15	0.70	1.00	

modularity on a very high significance level ($p < 0.0001$ after the HOLM correction). This is particularly interesting since many large metamodels like the UML metamodel used by Eclipse (Version 2.1) or many component models like Kevoree (Fouquet and Daubert, 2012) or SOFA 2 (Bureš et al., 2006) do not employ any modularization, i.e. all metaclasses are direct elements of a single root package. Here, our results indicate that these metamodels could be perceived as much better if they were introducing a package structure.

4 THREATS TO VALIDITY

The threats to the validity of our experiment are divided as usual into threats to internal and external validity. The internal validity refers to the systematic bias we might have introduced in the experiment design and is discussed in Section 4.1. The external validity refers to the degree in which the findings from our experiment can be generalized and are discussed in Section 4.2.

4.1 Internal Validity

Our experiment only consisted of one appointment in which the participants had to evaluate the metamodels. Thus, we can exclude an influence of histories, maturation or mortality. We have chosen the groups creating metamodels for the Mobiles case and the BPMN case randomly so that we can exclude an influence of selection. The evaluations of the metamodels were made by the participants so that we can exclude a subject effect from ourselves.

Not all metamodels have, however, been evaluated by all of the participants. Most metamodels have been evaluated by three participants out of the 24 participants of our experiment. Having the participants evaluate more metamodels would have borne the risk of getting less participants and of suffering from a strong sequencing effect. We have chosen the metamodels to be evaluated by the participants by random in order to minimize this effect.

Even so, we may have faced an instrumentation or sequencing effect as the participants evaluated the multiple metamodels of the same domain where subsequent evaluations may be biased depending on the first metamodel which they had to evaluate. To minimize this, participants were not allowed to change their opinion on previous metamodel evaluations. However, we assume that this effect is very small since the order was chosen by random and statistically, most metamodels were rated by some participants first and rated last by others. Furthermore, the participants only evaluated at maximum two metamodels per scenario.

4.2 External Validity

The mobile metamodels and the BPMN metamodel extensions were relatively small whereas in practice, much larger metamodels are used. Therefore, we cannot exclude the possibility that the correlations that we observed are not present in larger metamodels.

Furthermore, the participants of our experiment were academic professionals and students but we did not have participants from industry. Therefore, we cannot exclude the risk that the observed correlations would not be observed outside an academic setting.

When we asked the participants to assess the quality of the metamodels, we did not provide a specific purpose or usage scenario for the metamodels. Rather, we asked to assess the quality in a broader perspective. The participants evaluated the metamodels directly after they created some metamodels themselves. This sequence of tasks may have introduced an implicit tendency to assess the quality from the perspective of metamodel design and not, for example, from the perspective of transformation engineering.

5 FUTURE WORK

In the future, we want to use the results from the empirical study to evaluate the goodness of fit of metrics to quantitatively measure metamodel quality. Such a set of metrics would allow developers to detect possible shortcomings of their metamodels in a very early stage of the development. Here, the results of this paper guide us in the selection of metrics to evaluate. In particular, the results imply the necessity of metrics to measure the modularity of metamodels as well as their completeness and correctness. Unfortunately, the latter two quality attributes are very hard to measure. After all, a metamodel is a formalization of the domain and to avoid duplicate concepts, it is often the only one. This makes it difficult to choose artifacts that can be used to validate the completeness or correctness. Further, metamodels should abstract from the real world so that completeness is always relative to the application scenario of the models conforming to a metamodel.

This situation is different for modularity. Approaches already exist to improve the modularity of object-oriented design. We look forward to validate metrics to measure the modularity of metamodels in order to automatically improve their quality. Because of the high influence of modularity to the overall quality, the experiment metamodels of the Mobiles scenario are suitable to validate modularization metrics.

The combination of metamodels and a perception of their quality also yields other possibilities to sharpen the understanding of quality in an MDE context. In a first step, we want to validate the usage of metrics to measure metamodel quality. In a second step, we want to analyze how the design of different metamodels of the same domain affect subsequent artifacts such as model transformations and analyses. With such artifacts, the metamodel quality can be validated in terms of quality attributes of depending artifacts. Finally, the results can be combined to validate whether the perception of metamodel quality is actually correct in the sense that the expected positive effects of e.g. modularity persist in artifacts like model transformations and analyses.

Summing up, empirical experiments such as we have performed for this paper yield a large amount of valuable data that can be used for a series of hopefully meaningful results.

6 RELATED WORK

Related work in the context of metamodel quality mostly exists for the quantitative measurement of metamodel quality in metrics by adoption of metrics for UML class diagrams and more generally object-oriented design. However, to the best of our knowledge, the characterization of metamodel quality has not yet been approached through the perception of modeling experts.

Bertoa et al. (Bertoa and Vallecillo, 2010) define a rich framework of quality attributes for metamodels. However, the quality attributes are not validated in terms of how the quality attributes are perceived and no analysis has been done on the correlations between the quality attributes.

López et al. propose a tool and language to check for properties of metamodels (López-Fernández et al., 2014). In their paper, they also provide a catalog of negative properties, which they categorize in: design flaws, best practices, naming conventions and metrics. They check for breaches of fixed thresholds for the following metrics: number of attributes per class, degree of fan-in and -out, DIT and the number of direct subclasses. However, their catalog stems from conventions and experience and is not empirically evaluated. Further, the negative properties are not related to quality attributes.

Vépa et al. present a repository for metamodels, models, and transformations (Vépa et al., 2006). The authors apply metrics that were originally designed for class diagrams onto metamodels from the repository. The applied metrics are: several size metrics (as a basis for other metrics), DIT, several number of features per class metrics, number of inherited attributes and attribute inheritance factor. For some of the metrics, Vépa et al. provide a rationale how they relate to metamodel quality but no validation is given.

Williams et al. applied a variety of size metrics onto a big collection of metamodels (Williams et al., 2013). However, they did not draw any conclusions with regards to quality.

Di Rocco et al. also applied metrics onto a large set of metamodels (Di Rocco et al., 2014). Besides the usual size metrics, they also feature the number of isolated metaclasses and the number of concrete immediately featureless metaclasses. Further, they searched for correlations of the metrics among each other. E.g., they found that the number of metaclasses with super class is positively correlated with the number of metaclasses without features. Based on the characteristics they draw conclusions about general characteristics of metamodels. Their long-term goal is to draw conclusions from metamodel characteristics with regards to the impact onto tools and transformations which are based on the metamodel. However, in this work, they did not correlate the metric results to any quality attributes.

Gomez et al. propose an approach which aims at evaluating the correctness and expressiveness of a metamodel (Gómez et al., 2012). A metamodel is considered correct, if it only allows valid instances. Expressiveness is the degree in which it is able to express the instances it is supposed to. Their approach automatically generates a (preferably small) set of instances to evaluate these two criteria.

Garcia et al. developed a set of domain specific metamodel quality metrics for multi-agent systems modelling languages (García-Magariño et al., 2009). They propose three metrics: availability, specificity and expressiveness. These metrics take domain knowledge into account, e.g., the "number of necessary concepts" or the "number of model elements necessary for modelling the system of the problem domain".

Leitner et al. propose complexity metrics for domain models of the software product line field as well as feature models (Leitner et al., 2012). A feature model (Czarnecki and Eisenecker, 2000) is used to express variability. In its simplest form it is a tree with mandatory and optional nodes. More complex constraints are also possible using excludes or feature sets. A domain model is a DSL which also describes variability. However, domain models are not as constrained by their metamodels as it is the case with feature models. The authors argue, that the complexity of both, feature and domain models, influences the overall quality of the model, but especially usability and maintainability. They show the applicability of their metrics, but do not validate the influence between the metrics and quality.

Vanderfeesten et al. published work on quality and quality metrics for business process models (Vanderfeesten et al., 2007). They present a great number of metrics in their work. Some of them are so specific they can only be applied to business quality models, others are quite general and can be applied to metamodels or even graphs in general. They assess the relation metric results and error occurrences (Mendling and Neumann, 2007; Mendling et al., 2007a; Mendling et al., 2007b), between metric results and manual quality assessments (Sánchez-González et al., 2010) or to both (Vanderfeesten et al., 2008). However, transferring the relation between metrics and quality attributes to metamodels is similarly problematic as it is with metrics for object-oriented design. Business process models are used for very specific purposes. They describe business processes in the real world. They are used for documentation, communication, can be analyzed and simulated. The usage of metamodels is primarily instantiation into models. Thus, if some of the metrics for business process models can also be applied to metamodels, it cannot be assumed that their correlations to quality attributes still hold.

7 CONCLUSION

In this paper, we have described an empirical study to enhance the understanding of metamodel quality perception. We had a total of 24 participants, both students and professionals. The results from analyzing 89 questionnaires evaluating 23 metamodels of two domains shows that the perception of metamodel quality was mainly depending on completeness, correctness and modularity where other quality attributes like the consistency of the metamodel did not show a significant influence. The chosen quality attributes have shown only few correlations and are therefore a good starting point for the design of future experiments. As the influence of the domain was not significant for the perception of metamodel quality, we argue that our results are independent from the domains modeled in the scope of our experiment and thus can be generalized to other domains as well.

The significance of modularity for the quality perception of a metamodel is a signal to many metamodels used today in industry and academia that often ignore this quality attribute. Furthermore, we want to use the results to push the development and evaluation of metrics to measure the metamodel quality automatically.

ACKNOWLEDGEMENTS

This research has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 604102 (Human Brain Project) and the Helmholtz Association of German Research Centers.

REFERENCES

Bertoa, M. F. and Vallecillo, A. (2010). Quality attributes for software metamodels. In *Proceedings of the 13th TOOLS Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2010)*.

Bureš, T., Hnetynka, P., and Plášil, F. (2006). Sofa 2.0: Balancing advanced features in a hierarchical component model. In *Proceedings of the fourth International Conference on Software Engineering Research, Management and Applications*, pages 40–48. IEEE.

Czarnecki, K. and Eisenecker, U. W. (2000). *Generative Programming*. Addison-Wesley, Reading, MA, USA.

Di Rocco, J., Di Ruscio, D., Iovino, L., and Pierantonio, A. (2014). Mining metrics for understanding metamodel characteristics. In *Proceedings of the 6th International Workshop on Modeling in Software Engineering*, MiSE 2014, pages 55–60, New York, NY, USA. ACM.

Fouquet, F. and Daubert, E. (2012). Kevoree project. http://www.kevoree.org/. [Online; accessed 10-October-2013].

García-Magariño, I., Gómez-Sanz, J., and Fuentes-Fernández, R. (2009). An evaluation framework for mas modeling languages based on metamodel metrics. *Agent-Oriented Software Engineering IX*, pages 101–115.

Gómez, J. J. C., Baudry, B., and Sahraoui, H. (2012). Searching the boundaries of a modeling space to test metamodels. *Software Testing, Verification, and Validation, 2008 International Conference on*, 0:131–140.

Hinkel, G., Groenda, H., Vannucci, L., Denninger, O., Cauli, N., and Ulbrich, S. (2015). A Domain-Specific Language (DSL) for Integrating Neuronal Networks in Robot Control. In *2015 Joint MORSE/VAO Workshop on Model-Driven Robot Software Engineering and View-based Software-Engineering*.

Lehman, M., Ramil, J., Wernick, P., Perry, D., and Turski, W. (1997). Metrics and laws of software evolution-the nineties view. In *Software Metrics Symposium, 1997. Proceedings., Fourth International*, pages 20–32.

Lehman, M. M. (1974). *Programs, cities, students: Limits to growth? (Inaugural lecture - Imperial College of Science and Technology ; 1974)*. Imperial College of Science and Technology, University of London.

Leitner, A., Weiß, R., and Kreiner, C. (2012). Analyzing the complexity of domain model representations. In *Proceedings of the 19th International Conference and Workshops on Engineering of Computer Based Systems (ECBS)*, pages 242–248.

López-Fernández, J. J., Guerra, E., and de Lara, J. (2014). Assessing the quality of meta-models. In *Proceedings of the 11th Workshop on Model Driven Engineering, Verification and Validation (MoDeVVa)*, page 3.

Mendling, J. and Neumann, G. (2007). Error metrics for business process models. In *Proceedings of the 19th International Conference on Advanced Information Systems Engineering*, pages 53–56.

Mendling, J., Neumann, G., and van der Aalst, W. (2007a). On the correlation between process model metrics and errors. In *Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual modeling-Volume 83*, pages 173–178. Australian Computer Society, Inc.

Mendling, J., Neumann, G., and Van Der Aalst, W. (2007b). Understanding the occurrence of errors in process models based on metrics. In *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, pages 113–130. Springer.

Sánchez-González, L., García, F., Mendling, J., Ruiz, F., and Piattini, M. (2010). Prediction of business process model quality based on structural metrics. In *Conceptual Modeling–ER 2010*, pages 458–463. Springer.

The Object Management Group (2011). Business process model and notation 2.0. http://www.bpmn.org/.

Vanderfeesten, I., Cardoso, J., Mendling, J., Reijers, H. A., and van der Aalst, W. (2007). Quality metrics for business process models. *BPM and Workflow handbook*, 144.

Vanderfeesten, I., Reijers, H. A., Mendling, J., van der Aalst, W. M., and Cardoso, J. (2008). On a quest for good process models: the cross-connectivity metric. In *Advanced Information Systems Engineering*, pages 480–494. Springer.

Vépa, E., Bézivin, J., Brunelière, H., and Jouault, F. (2006). Measuring model repositories. In *Proceedings of the 1st Workshop on Model Size Metrics*.

Williams, J. R., Zolotas, A., Matragkas, N. D., Rose, L. M., Kolovos, D. S., Paige, R. F., and Polack, F. A. (2013). What do metamodels really look like? In *Proceedings of the first international Workshop on Experiences and Empirical Studies in Software Modelling (EESSMod)*, pages 55–60.

SCIENCE AND TECHNOLOGY PUBLICATIONS