Models, More Models, and Then A Lot More

Önder Babur, Loek Cleophas
and Mark van den Brand
Eindhoven University of Technology
Eindhoven, The Netherlands

Bedir Tekinerdogan
Wageningen University & Research
Wageningen, The Netherlands

Mehmet Aksit
University of Twente
Entschede, The Netherlands

***Abstract*—With increased adoption of Model-Driven Engineering, the number of related artefacts in use, such as models, greatly increase. To be able to tackle this dimension of scalability in MDE, we propose to treat the artefacts as data, and apply various techniques ranging from information retrieval to machine learning to analyse and manage them in a scalable and efficient way.**

I. Introduction

Model-Driven Engineering (MDE) promotes the use of models, metamodels and model transformations as first-class citizens to tackle the complexity of software systems. As MDE is applied for larger problems, the complexity, size and variety of those artefacts increase. With respect to model size and complexity, for instance, the issue of scalability has been pointed out by Kolovos et al. [1]. Regarding this aspect, a good amount of research has been done for handling a small number of (possibly very big and complex) models, e.g. in terms of comparison, merging, splitting, persistence or transformation. However, scalability with respect to model variety and multiplicity (i.e. dealing with a large number of possibly heterogeneous models) has so far remained mostly under the radar.

Before elaborating on this aspect of scalability in MDE, it should be mentioned that other artefacts within software engineering, notably source code, have had a longer history of widespread and large-scale use. This has naturally led to earlier adoption of techniques e.g. for searching in large codebases and data mining. Those techniques however, may not be directly translatable to the MDE domain due to the inherent differences of MDE artefacts from source code.

II. The Expanding Universe of MDE

This new scalability issue emerges partly due to some recent developments in the MDE community. Firstly, there have been efforts to (a) initiate public repositories to store and manage large numbers of models and other artefacts [2], [3]. Further efforts include mining repositories for public models to be used for MDE research, e.g. mining UML models from GitHub in [4] (the Lindholmen dataset), or mining Ecore metamodels by us [1]. In the former case, the number of UML models can go up to 90k. The sheer amount of models inevitably calls for techniques for searching, preprocessing (e.g. filtering), analysing and visualising the data in a holistic and efficient manner.

[1]to be published

Even within a single industry or organisation, a similar situation emerges with larger adoption of MDE. We have been collaborating with high tech companies in the Netherlands. In one of those companies, just one of the MDE ecosystems currently contains tens of metamodels, model transformations and thousands of models. With the complete revision history, the total number of artefacts staggeringly goes up to multiple tens of thousands. Similar stories in terms of scale hold for our other industrial partners with growing heterogeneous sets of models involving multiple domains. Note that besides the *versions*, for systems with implicit or explicit (e.g. as a Software Product Line) variability, *variants* can be considered another amplifying factor for the total number of MDE artefacts to manage.

A final observation is that our industrial partners increasingly use model-driven or model-based practices. This not only happens through manual migration from code-based development to MDE, but also automatically via process mining and automata learning. This confirms the statement by Brambilla et al. [5] that MDE adoption in (at least some parts of) the industry grows quite rapidly, and we conclude that tackling scalability will be increasingly important in the future.

III. Treating MDE Artefacts as Data

Based on the observations above, we advocate a perspective where MDE artefacts are treated holistically as data, processed and analysed with various scalable and efficient techniques from various disciplines listed as follows. While there is related MDE research on some of the items on the list, we believe a conscious and integrated mindset would mitigate the future challenges for scalable MDE.

a) Information Retrieval: Techniques from information retrieval (IR) can facilitate indexing and searching of models, and thus their management and reuse. The adoption of IR techniques on source code dates back to early 2000s, and within the MDE community there has been some recent effort in this direction (e.g. by Bislimovska et al. [6]). Further IR-based techniques (though mainly developed for model comparison/clustering) can be found in [7], [8] involving repository management and model searching scenarios.

b) Natural Language Processing: Accurate natural language processing (NLP) tools are needed to handle realistic models with noisy text content, compound words, and synonymy/polysemy. In our experience, it is very problematic to blindly use NLP tools on models, especially the *"Let's find*

synonyms using WordNet!" approach without proper part-of-speech tagging and word sense disambiguation. More research is needed on (1) finding the right chain of NLP tools applicable for models (in contrast with source code and documentation) and (2) reporting accuracies and disagreements between various tools (along the lines of the recent report in [9] for repository mining).

c) Data Mining: Following the perspective of approaching MDE artefacts as data, we need scalable techniques to extract relevant units of information from models (*features* in data mining (DM) jargon), and to discover patterns including domain clusters, outliers/noise and clones (see example applications in [7], [8], [10]). To be able to analyse, explore and eventually make sense of the large datasets in MDE (e.g. the Lindholmen dataset), we can investigate what can be borrowed from comparable approaches in DM for structured/graph data.

d) Machine Learning: The increasing availability of large amounts of MDE data can be exploited via machine learning to automatically infer certain qualities and functions. There has been a thrust of research in this direction for source code (e.g. for performance prediction, defect classification), and it would be noteworthy to investigate the emerging needs of the MDE communities and feasibility of the learning techniques for MDE. The approach in [11] for learning model transformations by examples is one of the few pieces of such work in MDE.

e) Visualization: We propose visualization and visual analytics techniques to inspect a whole dataset of artefacts (e.g. cluster visualizations in [8], in contrast with visualizing a single big model in [1]) using various features such as metrics and cross-artefact relationships. The goals could range from exploring a repository to analysing an MDE ecosystem holistically and even studying the (co-)evolution of MDE artefacts.

f) Distributed/Parallel Computing: With the growing amount of data to be processed, employing distributed and parallel algorithms in MDE is very relevant. While there are conceptually related approaches in MDE worthwhile investigating, e.g. distributed model transformations for very large models [12], [13] or model-driven data analytics [14], we wish to draw attention here to performing computationally heavy data mining or machine learning tasks for large MDE datasets in an efficient way.

We propose this non-exhaustive list as a preliminary exploitation guideline to help tackling scalability in MDE. Although the areas themselves are quite mature on their own, it should be investigated to what extent results and approaches can be transferred into the MDE technical space.

IV. Conclusion

We observe a rapid increase in the size of the MDE universe, which leads to scalability issues to be addressed by the community. To overcome this new and relatively overlooked challenge, we propose a holistic research perspective with several components ranging from information retrieval to machine learning.

Acknowledgments

This work is supported by the 4TU.NIRICT Research Community Funding in the Netherlands.

References

[1] D. S. Kolovos, L. M. Rose, N. Matragkas, R. F. Paige, E. Guerra, J. S. Cuadrado, J. De Lara, I. Ráth, D. Varró, M. Tisi, and J. Cabot, "A research roadmap towards achieving scalability in model driven engineering," in *Proceedings of the Workshop on Scalability in Model Driven Engineering*, ser. BigMDE '13. New York, NY, USA: ACM, 2013, pp. 2:1–2:10. [Online]. Available: http://doi.acm.org/10.1145/2487766.2487768

[2] F. Basciani, J. D. Rocco, D. D. Ruscio, A. D. Salle, L. Iovino, and A. Pierantonio, "Mdeforge: an extensible web-based modeling platform," in *Proceedings of the 2nd International Workshop on Model-Driven Engineering on and for the Cloud co-located with the 17th International Conference on Model Driven Engineering Languages and Systems, CloudMDE@MoDELS 2014, Valencia, Spain, September 30, 2014.*, 2014, pp. 66–75. [Online]. Available: http://ceur-ws.org/Vol-1242/paper10.pdf

[3] H. Störrle, R. Hebig, and A. Knapp, "An index for software engineering models," in *Joint Proceedings of MODELS 2014 Poster Session and the ACM Student Research Competition (SRC) co-located with the 17th International Conference on Model Driven Engineering Languages and Systems (MODELS 2014), Valencia, Spain, September 28 - October 3, 2014.*, 2014, pp. 36–40. [Online]. Available: http://ceur-ws.org/Vol-1258/poster8.pdf

[4] R. Hebig, T. H. Quang, M. R. Chaudron, G. Robles, and M. A. Fernandez, "The quest for open source projects that use uml: mining github," in *Proc. of MODELS 19*. ACM, 2016, pp. 173–183.

[5] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*, 1st ed. Morgan & Claypool Publishers, 2012.

[6] B. Bislimovska, A. Bozzon, M. Brambilla, and P. Fraternali, "Textual and content-based search in repositories of web application models," *ACM Transactions on the Web (TWEB)*, vol. 8, no. 2, p. 11, 2014.

[7] Ö. Babur, L. Cleophas, and M. van den Brand, "Hierarchical clustering of metamodels for comparative analysis and visualization," in *Proc. of the 12th European Conf. on Modelling Foundations and Applications, 2016*, 2016, pp. 2–18.

[8] F. Basciani, J. Di Rocco, D. Di Ruscio, L. Iovino, and A. Pierantonio, "Automated clustering of metamodel repositories," in *International Conference on Advanced Information Systems Engineering*. Springer, 2016, pp. 342–358.

[9] F. N. A. Al Omran and C. Treude, "Choosing an nlp library for analyzing software documentation: A systematic literature review and a series of experiments," in *14th International Conference on Mining Software Repositories*, 2017.

[10] Ö. Babur, "Statistical analysis of large sets of models," in *31th IEEE/ACM International Conference on Automated Software Engineering (ASE 2016), Singapore, Singapore, September 3-7, 2016*, 2016.

[11] I. Baki and H. A. Sahraoui, "Multi-step learning and adaptive search for learning complex model transformations from examples," *ACM Trans. Softw. Eng. Methodol.*, vol. 25, no. 3, pp. 20:1–20:37, 2016. [Online]. Available: http://doi.acm.org/10.1145/2904904

[12] A. Benelallam, A. Gómez, M. Tisi, and J. Cabot, "Distributed model-to-model transformation with atl on mapreduce," in *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering*. ACM, 2015, pp. 37–48.

[13] L. Burgueño, M. Wimmer, and A. Vallecillo, "Towards distributed model transformations with lintra," 2016.

[14] T. Hartmann, A. Moawad, F. Fouquet, G. Nain, J. Klein, Y. L. Traon, and J.-M. Jezequel, "Model-driven analytics: Connecting data, domain knowledge, and learning," *arXiv preprint arXiv:1704.01320*, 2017.