On the Scalability Criterion for Comparing Modeling Approaches

Philippe Collet, and Philippe Lahire

Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, France

Abstract. The Comparing Modeling Approaches (CMA) workshop series aims at providing a common relevant case study and a growing collection of criteria to compare different modeling approaches, from aspect-oriented to object-oriented and beyond. Among criteria to be defined and integrated, scalability is particularly relevant as the Model-Driven Engineering community identifies it as one of the main challenges ahead. In this paper, we make a first attempt at refining the scalability criterion for both approaches and their tool support, and we provide a first set of questions for this criterion.

Keywords: Model-driven development, Scalability, Assessment, Analysis.

1 Introduction

The aim of the *Comparing Modeling Approaches* (CMA) workshop series is to provide a common relevant case study and a growing collection of criteria to compare different modeling approaches, from aspect-oriented to object-oriented and beyond. As stated in the most up-to-date report on the criteria [1], *"the purpose of the criteria is to provide a basis for understanding, analyzing, and comparing various modeling approaches"*. This should lead to the organization of some sort of map of existing approaches, while providing well identified aspects to be considered when a new approach is developed.

While previous editions have seen important progress in the description of modeling and requirement approaches, and in the refinement of the comparison criteria, several of them have been identified as relevant but not refined and integrated yet with the others [1]. All these criteria are complex to define, being subject to differences of interpretation, with different viewpoints to apprehend them, different dimensions to quantify them, and consequently important difficulties to evaluate them. Among these criteria, we see *scalability* as particularly relevant to the Model-Driven Engineering (MDE) community, as it is identified as one of its main challenges ahead [2].

In this paper, we tackle the problem of defining the *scalability* criterion. We first discuss the term scalability and some specificities of its interpretation in a MDE settings. We then give a first description of the questions that could define this criterion in the CMA assessment. Thanks to feedback obtained during the workshop, the questions are also organized according to their location, i.e., whether they are going to be put in a separate *scalability* tab or in an existing tab.

2 On Scalability

2.1 Definitions

The concept of scalability is generally related to the capability of handling a large, and growing, number of entities, forming an increasing workload.

A concise definition is given in [3]:

> *"Scalability is a desirable property of a system, a network, or a process, which indicates its ability to either handle growing amounts of work in a graceful manner, or to be readily enlarged"*

The term is recognized as being intrinsically hard to define [4], and two interpretations are usually found in the literature [5]:

1. the ability to handle increased workload, *without adding resources to a system*;
2. the ability to handle increased workload *by repeatedly applying a cost-effective strategy for extending a system's capacity.*

In distributed systems, the notion is also split into *horizontal scalability*, the ability to connect entities to form a single logical unit, and *vertical scalability*, the ability to increase capacity by adding resources to an existing entity [6].

2.2 Scalability in MDE

Recently, as MDE techniques and tools are now applied to more complex, and obviously, larger software systems, the question of scalability in MDE is recognized as one of the main challenges for its community [2, 7]. A recent research roadmap identifies some issues related to scalability [8]:

- *Working with large metamodels and DSLs*, i.e. providing modular and/or compositional support at the metamodeling and syntax levels.
- *Querying and transforming on a large scale*, i.e. using lazy techniques, optimized engines and moves of computation to the cloud, especially as transformations is at the core of MDE.
- *Supporting collaborative modelling*, i.e. handling large models with several users through operations for editing, versioning, differencing, etc.
- *Efficient model persistence*, i.e. storing large models and indexing them.

Some of the concerns expressed over the scalability of the MDE approach can be reused to serve as assessment for the modeling approaches and their supporting tools. Quite naturally, by the nature of the scalability property, the issues above can be directly related to the tool support rather than to the approach itself. Nevertheless, we observe that for the first issue on working with large metamodels, the support for composition or modular decomposition can also be related to the composition operators (or rules) that are currently studied in the CMA assessment.

3 Scalability as a Criterion

We structure the study of scalability according to whether it relates to the approach or its supporting tools.

3.1 Questions on the Approach

As for approaches, we propose to ask questions on size and number of frequently used first-class entities/units of encapsulation. This would relate to entities already identified by current questions in the assessment. Questions are concerned with the most used entities in the application of the approach, whether they are supposed to be large entities, a very large number of them, or both. Details can also be requested on what kind of applications (examples, case studies, simulated or generated models, real-life software systems).

Then some questions should concern the capabilities of entities, or of some of their provided operations to be modularized in some way, with a form of module, support for separation of concerns, etc. It must be noted that this is partially covered by the current description of key modeling concepts. Nonetheless the focus here is on whether the entity (first class entity, unit of encapsulation) is known to facilitate scalability. As an example, in UML class diagrams, packages aim at structuring large designs (question C), while classes are more the atomic unit of encapsulation and the most used (question A). Similarly, in UML sequence diagrams, fragments and their referencing forms a modularization mechanism (question C). In our recent study on feature modeling with the Familiar DSL [9], we also observe the need for question D, with an explicit slice operator that allows to decompose an existing feature model. More examples could be added to the future comparison criteria document, i.e., the future version extended from [1], so to help in understanding the questions.

Additionally we propose to ask whether readability or understandability problems have been observed in known applications.

Questions (*in a new "Scalability" tab*):

A. What first class entities, units of encapsulation or other concepts are likely to be the most used when applying the approach (describe briefly the new concept if needed)?

B. What are the known applications or case studies with the largest entities (give the number)?

C. What are the known applications or case studies with the largest number of entities (give the number)?

D. Does the approach provide mechanisms to decompose large operations over the models (operations such as transformation or analysis)?

Questions (*in the "Modeling Dimensions" tab*):

– Does the approach provide modularization or SoC mechanisms in the aim of supporting large models (given the list of mechanisms)?

– Does the approach provide decomposition or view facilities over the entities (give the list of mechanisms)?

– Have readibility or undestandibility problems been reported (give if possible the limiting factor, i.e. number or size of entities)?

☐ examples

☐ case studies

☐ simulated/generated models

☐ real-life applications

3.2 Questions on the Tool Support

Concerning tool support, we propose to first focus on questions related to the maximum number and size of entities that the tool has been demonstrated, or is likely, to support. For question A on the largest entity, we expect to have relevant answers for cases like UML sequence diagrams, where the whole diagram in the entity. These first questions are similar to the ones of the approach related set. This seems important to us to explicitly separate the usage of the approach with and without its tooling support. We couple these questions with ones related to the growth of in-memory representation and computation time (questions C and E). Then issues related to the known limitations of the tool are to be tackled: limits of in-memory representation and/or persistent representation. To complement this, we propose to question over similar limitations in the the composition operators/rules (in-memory representation issue during application of the operators, computation time issue), and to extend the questions to other operations, such as transformations, to enable a better understanding of the known or potential bottlenecks in the tooling support.

Finally, we propose to question on the support in the tool for large models visualization, view management and collaborative design (diff, merge).

Questions (*in a new "Scalability" tab*):

A. What is the largest *entity* that the tool is demonstrated, or likely, to support?

☐ demonstrated

☐ likely

B. What is the largest *number of entities* that the tool is demonstrated, or likely, to support?

☐ demonstrated

☐ likely

C. Is the growth of the *in-memory* representation a function of the number of entities (if so, is it sublinear, linear, exponential or another) ?

D. Do *in-memory* representation limitations appear with the tool (give details on the limiting factor)?

E. Do *persistent* representation limitations appear with the tool (give details on the limiting factor)?

F. Is the growth of the *computation* time for composition operators a function of the number of entities (if so, is it sublinear, linear, exponential or another) ?

G. Does the usage of other operations than composition on entities (e.g. transformation, analysis) in the tool lead to limitations (give details on the operation)?

☐ in-memory representation issues

☐ computation time issues

Questions (*in a new "Composition" tab*):

– Does the usage of the composition operator (or rule) in the tool lead to limitations?

☐ in-memory representation issues (when applying the composition)

☐ computation time issues (when applying the composition)

Questions (*in the "Tool support" part of the "Key Modeling Concepts" tab*):

– What is the tool support for

☐ large models visualization (zoom,. . .)?

☐ view management (over the same entity,. . .)?

☐ differencing?

☐ merging?

– What is the collaborative support of the tool for the previous operations?

---

4 Conclusion

Scalability is one of the complex critera that are still in the *parking lot* of the CMA workshop assessment. In this paper we have proposed a first set of questions to characterize it. During the workshop this first step has fostered discussions and the presented set of questions has been refined.

Among open issues, the way to integrate the proposed questions in the current version of the assessment was discussed. This led to the organization of questions between a new *scalability* tab and integration in existing tabs. In the short term, we plan to integrate these proposals in the assessment questionnaire and to update previous model descriptions accordingly.

Acknowledgements

The work reported in this paper is partly funded by the ANR YourCast project under contract ANR-2011-EMMA-013-01. The authors would like to thank all participants of CMA@RE'13 and Barbados'13 workshops for their valuable discussions on the approaches, the bCMS case study and the comparison criteria, as well as the reviewers and participants of CMA@MODELS'13 for their feedback to extend and structure the proposed questions.

References

1. Geri Georg et. al.: Modeling Approach Comparison Criteria for the CMA Workshop at MODELS 2013. Technical report (April 2013)
2. Kolovos, D.S., Paige, R.F., Polack, F.: The grand challenge of scalability for model driven engineering. In Chaudron, M.R.V., ed.: MoDELS Workshops. Volume 5421 of Lecture Notes in Computer Science., Springer (2008) 48–53
3. Bondi, A.B.: Characteristics of scalability and their impact on performance. In: Proceedings of the 2nd international workshop on Software and performance. WOSP '00, New York, NY, USA, ACM (2000) 195–203
4. Hill, M.D.: What is scalability? SIGARCH Comput. Archit. News **18**(4) (December 1990) 18–21
5. Weinstock, C., Goodenough, J.: On system scalability. Technical Note SEI (CMU/SEI-2006-TN-012) (March 2006)
6. Michael, M., Moreira, J.E., Shiloach, D., Wisniewski, R.W.: Scale-up x scale-out: A case study using nutch/lucene. Parallel and Distributed Processing Symposium, International **0** (2007) 441
7. Di Ruscio, D., Kolovos, D.S., Matragkas, N., eds.: BigMDE '13: Proceedings of the Workshop on Scalability in Model Driven Engineering, New York, NY, USA, ACM (2013)
8. Kolovos, D.S., Rose, L.M., Matragkas, N., Paige, R.F., Guerra, E., Cuadrado, J.S., De Lara, J., Ráth, I., Varró, D., Tisi, M., Cabot, J.: A research roadmap towards achieving scalability in model driven engineering. In: Proceedings of the Workshop on Scalability in Model Driven Engineering. BigMDE '13, New York, NY, USA, ACM (2013) 2:1–2:10
9. Collet, P., Lahire, P.: Feature Modeling and Separation of Concerns with FAMILIAR. In: Comparing Requirements Modeling Approaches, International Workshop at RE'13 (CMA@RE'13). (2013) 6