

# Ten years of software and systems modeling

## Reflecting on the past, looking to the future

Gregor Engels · Jon Whittle

Published online: 25 September 2012  
© Springer-Verlag 2012

### 1 Introduction

The first issue of the *Journal of Software and Systems Modeling* was published in September 2002. In the very first editorial, the journal's Editors-in-Chief noted that one of the key aims of the journal was “to provide researchers as well as tool vendors and standardization committees with insights that can lead to better modeling languages and techniques, and provide software and system developers with a deeper understanding [...] that can lead to more effective application.”

In the 10 years that have passed since that first SoSyM issue, the world of software and systems modeling has seen a lot of changes. In 2002, UML was still a relative new kid on the block; since then, we have seen a major revision in the form of UML 2.0 in 2005. At the time of writing, the latest version of the UML specification is 2.4. Just prior to the publication of the first SoSyM issue, a major development in modeling had just been introduced. In 2001, the Object Management Group brought out the first specification of Model Driven Architecture (MDA). This was a development that was to cause high levels of controversy in software engineering circles as software luminaries took turns to declare it “dead on arrival” [1] or to herald a new wave of productivity increases [2, 3]. The preceding 10 years have also seen enlivened debate on the merits of a general purpose language, such as UML, over bespoke domain-specific modeling

languages (cf. [4]). Thankfully, such arguments have now settled, with both parties accepting the benefits of each approach and, furthermore, recognizing that the modeling community is more powerful united<sup>1</sup> than divided.

Life is punctuated by milestones, especially as we get older. There is an increasing trend for us to celebrate the big birthdays, those that mark the passing of a decade, and to declare that “life begins at forty” or “fifty is the new forty”; to put it simply, we hope that the best times are yet to come. As with life, the pursuit of scholarship has a human need to recognize its achievements. It is with pleasure, therefore, that we, as guest editors, mark 10 years of SoSyM with a special issue that both reflects on developments in software and systems modeling over the last decade, but also looks to the future at what modeling could become 10 years from now.

This special issue contains 10 invited papers. Authors were invited based on a consultation with the SoSyM Editors-in-Chief and Editorial Board. The criteria for selection were that authors be well-respected experts in software and systems modeling with the stature and experience to comment authoritatively on the past, present and/or future of modeling. Each paper went through two rounds of reviewing—first by the editors to provide strategic feedback and to ensure a well-balanced special issue, and secondly by SoSyM reviewers who looked for value in each paper in terms of synthesizing knowledge in the field or setting a research agenda for the future. Rather than focusing on original scientific contributions, the papers review the state-of-practice, reflect on how we got to where we are, consider where alternative paths might have led us, or attempt to predict the unpredictable—the future.

---

G. Engels  
s-lab, Software Quality Lab, University of Paderborn,  
Paderborn, Germany  
e-mail: engels@upb.de

J. Whittle (✉)  
School of Computing and Communications, Lancaster University,  
Lancaster, UK  
e-mail: whittle@comp.lancs.ac.uk; j.n.whittle@lancaster.ac.uk

<sup>1</sup> Not necessarily unified!

## 2 What's in it for you?

The special issue is presented in three sections. In the first, “Retrospective”, Steve Cook presents a fascinating account of the history of UML. The paper lays out the key milestones in its development and cleverly revisits issues—both technical and political—that have negatively impacted the development of the language. It is not all bad news though; Cook titillates with his description of ongoing efforts to fix some of the problems of UML he has highlighted.

Also in the “Retrospective” section is an insiders’ guide to anyone looking to start a company selling modeling tools. At the height of the hype around modeling, Tony Clark and Pierre-Alain Muller were both key players in different model-driven engineering (MDE) start-ups. As it turns out, neither was wholly successful. Clark and Muller describe some of the challenges they faced along the way, dissect the reasons why their companies ultimately shifted business model or folded, and give important lessons for the rest of us thinking of making a living out of MDE.

The bulk of the special issue is an opportunity to reflect on some of the “Really Hard Questions” of software and systems modeling. This section discusses achievements and open issues in the field. When it comes to modeling, hard questions come no bigger than the notion of abstraction. And few would dare to tackle this concept head-on. Michael Jackson, however, is a seasoned hand at addressing the really hard problems and devotes his paper to the essence of abstraction. In it, he discusses various dimensions of abstraction, relates the notion of abstraction to different fields of computer science and discusses the complementary notion of refinement.

Stephen Mellor has famously said that modeling will be commonplace in “three years’ time.” Of course, as he himself points out, he has been saying this for the last two decades! Bran Selic asks in his paper, what will it take before MDE is mainstream in industrial software development? A key insight is that besides technical factors, cultural and social as well economic factors are crucial for the success of introducing MDE into industrial practice. Selic unravels some of the social factors affecting MDE adoption, which includes a discussion of improved modeling education, as well as some important economic factors, such as the cost of tools. Finally, Selic presents key technical achievements and open issues.

Having heard from Selic about the limiting cost factor of MDE tools, the next paper asks the really hard question “What does the ideal MDE tool look like?” This thorny issue is tackled by two researchers who are intimately familiar with tool issues, having led the development of a number of successful academic MDE tools. Richard Paige and Dániel Varró team up to bring their combined exper-

tise to bear and offer a number of insights for future tool developers.

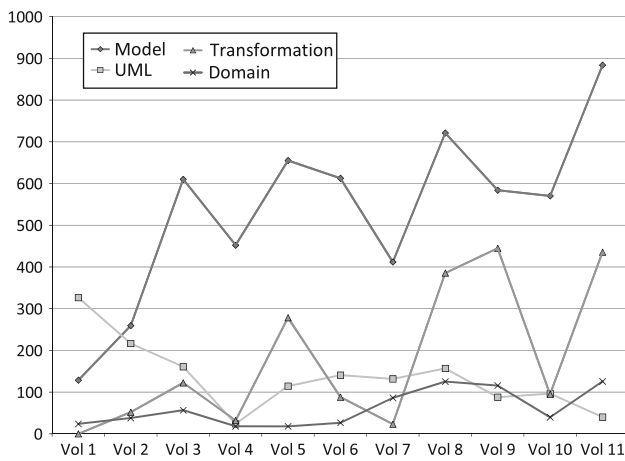
The next two papers reflect on one of the most fundamental modeling questions, that of the quality of models. In the first, Willem Visser, Matthew Dwyer and Michael Whalen provide a solid synthesis of ideas from various model checking paradigms, noting that “hidden” models are common in verification approaches. The notion of hidden models turns out to be crucial—as the authors state, model quality is directly related to how effectively the hidden models can be created.

The second paper on model quality looks specifically at process modeling, asking “What makes a good process model?” Process modeling is a key growth area and a domain where MDE has been effectively applied in recent years. One of the world experts on process modeling, Wil van der Aalst, shares his vast experience and looks back upon work in applying process mining tools and techniques in over 100 organizations. Based on these experiences, he proposes a set of seven problems of process modeling, which determine how process models should be developed and used in practice.

In the final paper in the “Really Hard Questions” section, Michel Chaudron, Werner Heijstek and Ariadi Nugroho round us off by summarizing some of the work that has been done to empirically evaluate software and systems modeling. Empirical assessment of MDE is an under-researched area; but if we are to avoid repeating the mistakes of the past, we must gain an in-depth understanding of how modelers think and how modeling is applied in practice.

No anniversary edition of a journal would be complete without an analysis of the future. The final section of the special issue is “A Look to the Future.” This section includes two such papers. First, Jean-Marc Jézéquel, Benoit Combemale, Steven Derrien, Clément Guy and Sanjay Rajopadhye look at the topic of software language engineering from two viewpoints: the compilation viewpoint with its long tradition in defining languages and creating compilation tools and the MDE viewpoint with its shorter tradition in defining and using modeling languages to represent complex systems at an abstract level. The paper is a serious attempt to unite these two sub-fields and offers important thoughts on future research directions.

Finally, the special issue is honoured to include a contribution from David Harel, the inventor of statecharts. Along with Assaf Marron, Harel re-examines the holy grails of modeling, namely, compositional, executable and intuitive models. The paper presents three powerful ideas. First, the idea that modeling in the future will be aligned with cognitive processes. Second, the comment that humans do not manipulate previous memories but “sculpt” their knowledge to develop their current model. And third, the future role of customization in software.



**Fig. 1** Log likelihood (y-axis) of terms by SoSyM volume (x-axis). See Table 1 for word groupings used

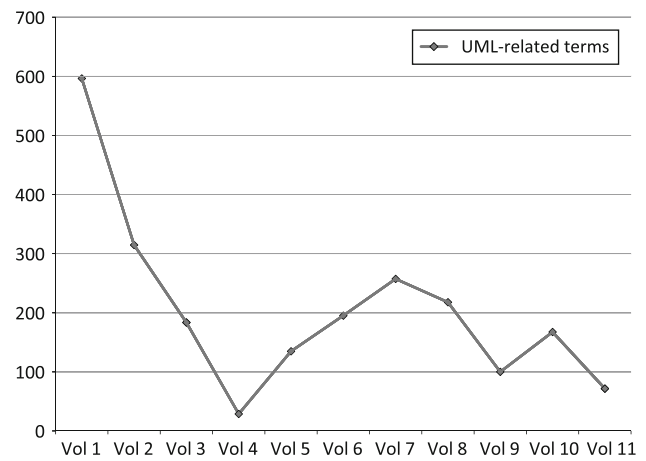
### 3 What's new?

We end this editorial by whetting the reader's appetite with some reflection on how SoSyM papers have changed over the last decade. John Hutchinson, a researcher at Lancaster University, very kindly agreed to carry out a linguistic analysis of the papers published in SoSyM since the inception of the journal. Using the Wmatrix corpus analysis and comparison tool [5], he analyzed the words and phrases used in 10 years' worth of SoSyM titles and abstracts.<sup>2</sup> The intention was to look for trends in terms of how subjects of discourse have changed with the times.

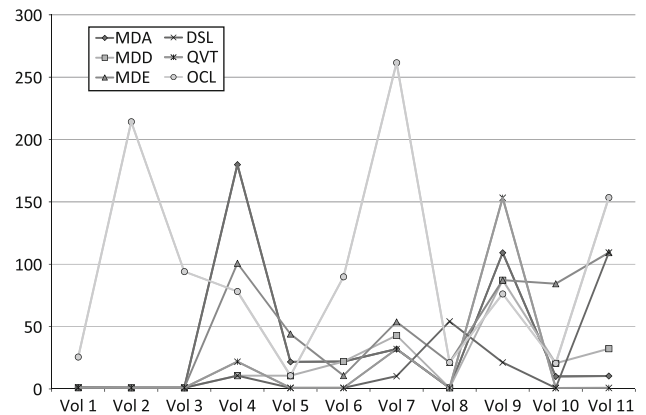
Wmatrix allows a data sample to be compared with a suitable reference corpus—in this case a million word sample of written data taken from the British National Corpus. The comparison highlights words and expressions that are over-represented in the data sample; that is, words that are relatively more frequently used in the sample when compared to average written usage. It is these over-represented words and expressions that indicate the prominent themes contained in the sample. The measure of over-representation is log likelihood (LL), which compares observed and expected frequencies of the word and expressions in question. By creating a separate data file for each volume of SoSyM, we can observe how the prominent themes have changed over the years.

A typical Wmatrix analysis will usually focus on individual words and expressions; however, for our analyses, words and expressions were grouped together. This was done to acknowledge that certain words used in modeling have similar meanings—such as DSL and domain-specific. Figures 1, 2 and 3 highlight certain trends over 11 volumes of SoSyM published since the inception of the journal.

<sup>2</sup> Editorials were excluded to avoid any potential bias from the Editors-in-Chief. Special Issues were included (but again not the editorials).



**Fig. 2** Log likelihood (y-axis) of UML-related words and expressions by SoSyM volume (x-axis). See Table 2 for word groupings used



**Fig. 3** Log likelihood (y-axis) of acronyms (and acronym expansions) by SoSyM volume (x-axis)

Figure 1 shows the LL value of key themes related to SoSyM. A number of key trends emerge from this Figure. First, there is a clear move away from articles that focus on UML and a move towards articles that focus more generally on modeling. This makes sense; indeed, 2005 saw the name of the UML Conference change to MODELS. Second, there is also a trend towards increasing talk of transformations, although the upward drift is somewhat more erratic. Third, and perhaps most surprisingly, there is only a very slight (and arguably only flat) rising trend of discourse on domain-specific concepts.

Figure 2 unpicks the move away from UML in a little more detail. Here, we include not only words directly mentioning UML but also words specifically associated with UML (see Table 1 for a list). The figure highlights more readily the decline of UML-related topics over the lifetime of the journal.

Finally, Fig. 3 is a glimpse of how SoSyM titles and abstracts have used acronyms over the years—here, we group together the acronym itself as well as its fully expanded version (so, for instance, the LL for OCL also includes the LL

**Table 1** Grouped words for linguistic analysis (term analysis)

Category	Included words and expressions
Model	Model, models, modeling, modelling, modeled, modelled, modelers, model_elements, modeller
UML	uml, uml-based unified
Transformation	Transformation, transformations
Domain	Domain, domains, dsl, domain-specific

The methodology for grouping was to first run the analysis without any grouping and then to group words that appeared with high LL and similar meaning

**Table 2** Grouped words for linguistic analysis (UML-related words and expressions)

Category	Included words and expressions
UML-related	uml (and all varieties, e.g. uml_books, uml_class, etc.) unified association, associations, multiplicity, multiplicities, n-ary, stereotype, stereotypes

The methodology for grouping was to first run the analysis without any grouping and then to group words that appeared with high LL and were specific to UML

for Object Constraint Language). Note the spikes in MDA (vol. 4) and QVT (vol. 9). MDE shows a somewhat upward trend over the lifetime of the journal. The real message here, however, is the staying power of OCL, which consistently gets mentioned, albeit with dips along the way.

Although what we have presented here is a very preliminary analysis of SoSyM language use—and a much more in-depth study could possibly uncover more interesting trends—we hope that we given the reader some indication of how SoSyM has changed over the years.

**Acknowledgments** First and foremost, the editors would like to thank the authors, all of whom are extremely busy people who somehow managed to find time to write interesting and insightful papers for this special issue. The editors would also like to thank the many reviewers who undertook the hard task of assessing drafts of papers and made important contributions that improved the papers. We are indebted to the Editors-In-Chief, Robert France and Bernhard Rumpe, who motivated us to continue when our energy was flagging and who helped immensely with organization. We thank Martin Schindler who handled the administrative process and lighted a pathway for us through the idiosyncracies of Manuscript Central. Finally, we thank John Hutchinson who conducted the linguistic analysis of 10 years of abstracts published in SoSyM.

## References

1. Zetie, C.: MDA is DOA, partly thanks to SOA. Forrester Research Trends Report, March (2006)
2. Frankel, D.S.: Model Driven Architecture: Applying MDA to Enterprise Computing. John Wiley and Sons, Hoboken (2003)

3. Mellor, S.J.: MDA Distilled, Principles of Model Driven Architecture. Addison-Wesley Professional, Boston (2004)
4. Frankel, D., Parodi, J. (eds.): The MDA Journal: Model Driven Architecture Straight from the Masters (2004)
5. Rayson, P.: From key words to key semantic domains. International Journal of Corpus Linguistics **13**(4), 519–549 (2008)

## Author Biographies



**Gregor Engels** received his PhD in computer science in 1986 from the University of Osnabrück, Germany. Between 1991 and 1997, he held the position of Chair of Software Engineering and Information Systems at the University of Leiden, the Netherlands. Since 1997, he has been Professor of Informatics at the University of Paderborn, Germany. Currently, he is also director of two technology transfer laboratories at the University of Paderborn, the

s-lab—Software Quality Lab and the C-LAB, where more than 50 PhD students do joint research with industrial partners. His research interests are in the area of model-driven software development, software architecture, and software quality assurance. He has been a member of the Steering Committee of the MODELS conference since 2006 and served as PC Chair in Nashville in 2007. He also serves on the editorial board of the Journal of Software and System Modeling.



**Jon Whittle** is full Professor and Chair of Software Engineering at Lancaster University. He is also a Royal Society Wolfson Merit Award Scholar. He has been working with Model-Driven Development for over ten years, including stints at NASA Ames Research Center (California), George Mason University (Virginia), IIT Kanpur (India), and more recently at Lancaster. He has been intimately involved with the MDD research community during this time, serving as

the Chair of the Steering Committee of the MODELS conference from 2006 to 2008 and as PC Chair in New Zealand in 2011. He also serves on the editorial board of the Journal of Software and System Modeling. Jon's current interests are in empirically investigating what factors lead to success or failure with MDD in industry.