

# ModComp 2017: 4<sup>th</sup> International Workshop on Interplay of Model-Driven and Component-Based Software Engineering

Federico Ciccozzi\*, Ivano Malavolta<sup>†</sup>

\* Mälardalen University – IDT, Västerås (Sweden)

federico.ciccozzi@mdh.se

<sup>†</sup> Vrije Universiteit Amsterdam, Amsterdam (The Netherlands)

i.malavolta@vu.nl

**Abstract**—Model-Driven Engineering (MDE) and Component-Based Software Engineering (CBSE) have been proven to effectively reduce software development complexity by (i) shifting the focus from source code to models and (ii) breaking down the set of desired features and their intricacy into smaller sub-modules, respectively. Moreover, the interplay of MDE and CBSE approaches is gaining recognition as a very promising means to boost the development of software systems by reducing costs and risks and shorten time-to-market. While several attempts to effectively combine MDE and CBSE have been documented, there are still unsolved clashes raising when exploiting interplay of MDE and CBSE, mostly due to mismatches in the related terminology as well as to differences in their basic essence.

The goal of ModComp is to gather researchers and practitioners to share opinions, propose solutions to open challenges and generally explore the frontiers of interweaving between MDE and CBSE.

**Index Terms**—Model-driven engineering, component-based software engineering, MDE, CBSE, interplay, workshop, MODELS.

## I. ABOUT MODCOMP

The design of modern software systems requires support capable of properly dealing with their ever-increasing complexity. In order to account for such a complexity, the whole software engineering process needs to be rethought and, in particular, the traditional division among development phases to be revisited, hence moving some activities from design time to deployment and runtime. Model-Driven Engineering (MDE) [1] and Component-Based Software Engineering (CBSE) [2] can be considered as two orthogonal ways of reducing development complexity: the former shifts the focus of application development from source code to models in order to bring system reasoning closer to domain-specific concepts; the latter aims to organize software into encapsulated independent components with well-defined interfaces, from which complex applications can be built and incrementally enhanced.

When exploiting these development approaches, numerous different modelling notations and consequently several software models are involved during the software life cycle. On the one hand, effectively dealing with all the involved models and heterogeneous modelling notations that describe software

systems needs to bring component-based principles at the level of the software model landscape hence supporting, e.g., the specification of model interdependencies, and their retrieval, as well as enabling interoperability between the different notations used for specifying the software. On the other hand, MDE techniques must become part of the CBSE process to enable the effective reuse of third-party software entities and their integration as well as, generally, to boost automation in the development process.

An effective interplay of CBSE and MDE approaches could help in handling the intricacy of modern software systems and thus reducing costs and risks by: (i) enabling efficient modelling and analysis of extra-functional properties, (ii) improving reusability through the definition and implementation of components loosely coupled into assemblies, (iii) providing automation where applicable (and favourable) in the development process. In the last fifteen years, such a cooperation has been recognized as extremely promising; tools and frameworks have been developed for supporting this kind of integrated development process. In the last few, the application of MDE and CBSE to tackle great challenges related to high heterogeneity and variability of complex systems, like the Internet-of-Things, has been recognised too [3]. Nevertheless, when exploiting interplay of MDE and CBSE, clashes arise due to misalignments in the related terminology but also, and more importantly, due to differences in some of their basic assumptions and focal points.

The goal of the workshop on Interplay of Model-Driven and Component-Based Software Engineering 2017 (ModComp'17) was to gather researchers and practitioners to share opinions, propose solutions to open challenges, and generally explore the frontiers of collaboration between MDE and CBSE. ModComp'17 aimed at attracting contributions related to the subject at different levels, from modelling to analysis, from componentization to composition, from consistency to versioning; foundational contributions as well as concrete application experiments were sought.

## II. SUMMARY

The workshop was co-located with the ACM/IEEE 20th International Conference on Model Driven Engineering Languages & Systems (MODELS), and represented an active forum for practitioners and researchers. We received eight full submissions, out of which the following six were selected for inclusion in the workshop's program and proceedings:

- Model-based Design of Reusable Secure Connectors. Authors: Michael Shin, Hassan Gomaa and Don Pathirage
- A Classification of Dynamic Reconfiguration in Component and Connector Architecture Description Languages. Authors: Arvid Butting, Robert Heim, Oliver Kautz, Jan Oliver Ringert, Bernhard Rumpe and Andreas Wortmann
- Model-driven Development of Adaptive IoT Systems. Authors: Mahmoud Hussein, Shuai Li and Ansgar Radermacher
- OCL Framework to Verify Extra-Functional Properties in Component and Connector Models. Authors: Shahar Maoz, Ferdinand Mehlman, Jan Oliver Ringert, Bernhard Rumpe and Michael von Wenckstern
- Model-driven generation of a BPMS portal based on Interaction Flow Modeling Language models. Authors: Daniel Calegari and Andrea Delgado
- A Multipurpose Framework for Model-based Reuse-oriented Software Integration Synthesis. Authors: Alexander Perucci, Marco Autili and Massimo Tivoli

In addition to paper presentations, we hosted two invited talks to get insights from both academic and industrial perspectives:

- CBSD through MDE vs MDE through CBSD: two different stories, an academic perspective on the interplay of MDE and CBSE given by Antonio Cicchetti, Associate Professor at Mälardalen University (Sweden)
- Various avataar of component concept and MDE – a practitioner's perspective, given by Vinay Vasant Kulkarni, Chief Scientist and Head of Software Systems Research at Tata Consultancy Services (India)

The accepted papers and invited talks covered many different forms of intertwining of MDE and CBSE, including: design of secure reusable connectors, dynamic reconfiguration of components with MDE, MDE of adaptive component-based IoT systems, verification of extra-functional properties of components through MDE, different flavours of the concept of component in industrial MDE, MDE through CBSE and vice versa from a research perspective.

This was the fourth edition of the workshop and the very good received attention (between 20 and 30 participants) demonstrates that the topics discussed in the workshop are relevant both in practice and in theory.

## III. OUTLOOK

We are planning to organise another theme issue on ModComp topics at an international journal in the software engineering field, as we did for ModComp 2015 [4]. One of the clear messages that arose during paper presentations and

discussions is the need and importance of a more focused research effort in the interplay of MDE and CBSE. For this reason, we are already planning for the next edition of ModComp, which we hope to be able to propose once again at MODELS.

## ACKNOWLEDGMENT

The organization of the ModComp workshop is partially supported by the Knowledge Foundation through the MOMENTUM project (<http://www.es.mdh.se/projects/458-MOMENTUM>). We would like to thank the authors and the program committee for their hard and precious work before, during, and after the workshop.

## REFERENCES

- [1] D. C. Schmidt, "Model-driven engineering," *COMPUTER-IEEE COMPUTER SOCIETY*, vol. 39, no. 2, p. 25, 2006.
- [2] I. Crnkovic, "Component-based software engineering: new challenges in software development," *Software Focus*, vol. 2, no. 4, pp. 127–133, 2001.
- [3] F. Ciccozzi, I. Crnkovic, D. Di Ruscio, I. Malavolta, P. Pelliccione, and R. Spalazzese, "Model-driven engineering for mission-critical IoT systems," *IEEE Software*, vol. 34, no. 1, pp. 46–53, 2017.
- [4] F. Ciccozzi, J. Carlson, P. Pelliccione, and M. Tivoli, "Editorial to theme issue on model-driven engineering of component-based software systems," *Software & Systems Modeling*, pp. 1–4.