Accepted Manuscript

Introduction to the Special issue on Methods, Tools and Languages for Model-driven Engineering and Low-code Development

Geylani Kardas, Federico Ciccozzi, Ludovico Iovino

DOI: 10.1016/j.cola.2022.101190

To appear in: *Computer Languages, Systems & Structures*

Published online: 25 January 2023

Please cite this article as: Geylani Kardas, Federico Ciccozzi, Ludovico Iovino, Introduction to the Special issue on Methods, Tools and Languages for Model-driven Engineering and Low-code Development, Journal of Computer Languages (2023), doi: 10.1016/j.cola.2022.101190

Introduction to the Special issue on Methods, Tools and Languages for Model-driven Engineering and Low-code Development

Geylani Kardas[1], Federico Ciccozzi[2], Ludovico Iovino[3]

[1] Ege University, International Computer Institute, 35100, Bornova, Izmir, Turkey
geylani.kardas@ege.edu.tr

[2] Mälardalen University, Box 883, Västerås, 72350, Sweden
federico.ciccozzi@mdh.se

[3] Gran Sasso Science Institute, Computer Science Department, L'Aquila, Italy
ludovico.iovino@gssi.it

Model-driven engineering (MDE) has been leveraged for many years to reduce the complexity of developing systems and software in various domains including automotive, cyber-physical systems, defense industry, embedded systems, and telecommunication. Models are being used throughout MDE processes for providing abstract representations of real problems as well as for supporting the communication and knowledge sharing between stakeholders and generating artifacts by automatically manipulating and transforming models. In this context, research has focused on multiple aspects of definition, maintenance and evolution of meta-models and modeling languages, definition and evaluation of model-driven methodologies, and provision of automated development support in terms of model-driven computer-aided software engineering (CASE) tools. However, there are still many challenges related to the adoption and application of MDE especially when coping with the heterogeneity and the scalability of modern systems as well as the need to manage variability and complexity of interrelated models. Quality of methodologies and techniques is important for MDE to continue being successful in industry. Frameworks and methods to systematically evaluate applied MDE approaches possibly with some predefined quantitative and qualitative metrics/criteria are very much needed.

Most MDE processes entail the use of domain-specific modeling languages (DSMLs) to some extent. The literature is rich with successful stories on the practical use of DSMLs. However, as their use becomes widespread, new challenges, especially related to co-evolution, usability, and collaboration across DSMLs arise and shall be investigated thoroughly.

Recently, industry has focused on low-code development (LCD) technologies for the creation of especially software-as-a-service (SaaS) and platform-as-a-service (PaaS) solutions. Companies such as Mendix, Salesforce, Outsystems provide LCD tools for their customers to develop and manage enterprise applications rapidly, reducing the traditional programming efforts. While using these tools, the developers benefit from modeling and MDE, e.g. with built-in graphical application development editors. Nevertheless, the cost of adopting LCD methodologies and evaluation of using them in industry need further investigation.

This special issue of the Journal of Computer Languages (COLA) aims at presenting the most recent theoretical and practical studies on the interplay of MDE and LCD along with the above mentioned research challenges and directions. This mainly includes metamodeling / modeling languages for MDE and LCD, new MDE and LCD techniques / methodologies, model transformation languages and model transformations processes, Artificial Intelligence for MDE, quality aspects of applying MDE and LCD and the industrial applications of MDE and LCD. After a rigorous peer-review process involving at least three reviewers per paper, four papers were finally accepted for inclusion in this special issue.

The first paper, entitled "*Generating customized low-code development platforms for digital twins*", by Manuela Dalibor, Malte Heithoff, Judith Michael, Lukas Netz, Jérôme Pfeiffer, Bernhard Rumpe, Simon Varga and Andreas Wortmann, introduces an integrated method for the MDE of LCD platforms for digital twins (DTs) in which the domain experts can create and operate DTs for cyber-physical systems. In the proposed method, models and languages for the DTs to be developed with the platform are collected first. That leads to obtaining the first code generator which produces a web-based LCD platform capable of modeling with these languages. Second, the DT to be developed is configured with various models, hence a second code generator which is an executable instance of these DTs is produced with self-adaptive behavior and visualization capabilities.

The second paper, entitled "*Automatic resolution of model merging conflicts using quality-based reinforcement learning*", by Mohammadreza Sharbaf, Bahman Zamani and Gerson Sunyé, discusses the application of Artificial Intelligence techniques to facilitate collaborative modeling and enabling the personalized and quality-based integration of model versions. Within this context, the authors propose the use of reinforcement learning algorithms to achieve merging conflict resolution encountered in collaborative modeling with a high degree of automation. The resolution of Unified Modeling Language (UML) class diagram conflicts using a learning process is shown. Moreover, the applicability of the proposed approach through a proof of concept implementation is demonstrated and its accuracy is evaluated by comparing with the greedy and search-based algorithms. Usability of the approach is also evaluated with the selection of resolution actions for different syntactic and semantic conflicts.

In the third paper of this SI, entitled "*Process-aware digital twin cockpit synthesis from event logs*", Dorina Bano, Judith Michael, Bernhard Rumpe, Simon Varga and Mathias Weske, present an LCD approach to reduce the amount of hand-written code needed to derive process-aware digital twin cockpits (PADTCs). Following a clear definition for digital twin cockpits and PADTCs, a method using process mining techniques to extract an event log from sensor data and then applying data-to-model transformations to infer the data model, and to discover the process model and roles is introduced. Model-to-model and model-to-code transformation techniques are proposed to automate the digital twin generation process where the discovered models are used as inputs for a code generator. A case study from an automated hospital transportation system is discussed to show how a high degree of automation in the PADTC engineering process can be achieved.

Finally, the fourth paper, entitled "*What about the usability in low-code platforms? A systematic literature review*", by Daniel Pinho, Ademar Aguiar and Vasco Amaral, examines the current literature about low-code and no-code to discover their relationship with usability, particularly its quality, aiming at determining which factors are the most relevant, and how users view these related tools. The systematic literature review (SLR) performed in this study covers the analysis, synthesis and reporting of 38 out of 207 relevant articles. The conducted SLR shows that a formal definition for low-code and/or no-code development does not currently exist although common characteristics have already been specified in the examined studies. Moreover, the SLR reveals that some of the authors performed the feasibility studies

on their implementations or listed factors that influence the user experience in a given tool. However, it also exposes that the researchers are still considering usability factors unconsciously, and a higher presence of intentional usability may benefit the low-code field.

We would like to thank all reviewers who participated in the reviewing process of the manuscripts submitted to this special issue. Finally, we would also like to thank COLA journal's Editor-in-Chief, Marjan Mernik, for his support and help he provided in all stages of preparing this special issue.