Blended Modelling for Software Architectures

Malvina Latifaj, Federico Ciccozzi, Muhammad Waseem Anwar
School of Innovation, Design and Engineering
Mälardalen University
Västerås, Sweden
federico.ciccozzi@mdu.se, muhammad.waseem.anwar@mdu.se,
malvina.latifaj@mdu.se

Kousar Aslam, Ivano Malavolta
Computer Science Department
Vrjie Universiteit Amsterdam
Amsterdam, The Netherlands
k.aslam@vu.nl, i.malavolta@vu.nl

I. Summary

Topic. Blended modelling is an emerging trend in Model-Driven Engineering for complex software architectures. It enables the modelling of diverse architectural aspects through multiple editing notations seamlessly, interchangeably, and collaboratively. Blended modelling is expected to significantly improve productivity and user experience for multiple stakeholders. To manually architect and build a blended modelling environment is not trivial. To support architects in this task, in the scope of the ITEA3 BUMBLE project, we have designed and developed a blended modelling framework that aids architects in designing and semi-automatically generating blended modelling environments for architecting software.

In this tutorial we will demonstrate two major activities in modelling and designing software architectures:

1) Given a domain-specific language at hand, we will show how to architect a blended modelling environment for it. From that, we will show how to leverage our framework to semi-automatically generate the blended modelling environment in the tool ecosystem (Eclipse Modeling Framework).
2) Once the blended modelling environment is generated, we will show how to architect software systems by using the generated blended modelling environment, demonstrating its seamless synchronization capabilities across different modelling notations and collaborative editing features.

Take-away messages:

1) Blended modelling is apt to simplify architecting of software systems
2) Architecting blended modelling environments is not hard with the appropriate support

State-of-the-art. There exist solutions for blended modelling of specific subsets of certain languages, but there is currently no solution for the architecting and generation of the blended modelling environment itself starting from a language specification (see Section II).

Audience. Research and practitioners interested in efficient architecting of tooling infrastructures for architecting software systems and/or in architecting software systems in a more effective manner.

Relevance for ICSA. This tutorial is intended to attract both software tool architects and software architects by showing how to more efficiently architect tools for software architects and how to use those tools for architecting software systems.

II. Background and related work

The aim of this section is to contextualize this tutorial and describes its scope.

Ciccozzi et al. [1] formally introduced the concept of blended modeling as:

> Blended modeling is the activity of interacting seamlessly with a single model (i.e., abstract syntax) through multiple notations (i.e., concrete syntaxes, allowing a certain degree of temporary inconsistencies.)

The notion of blended modeling may seem similar or even overlapping with multi-view modeling [2] that is based on the paradigm of viewpoint/view/model as formalized in the ISO/IEC 42010 standard. For example, Wimmer and Kramler [3] proposed a conceptual mapping between the textual languages EBNF and corresponding meta-models to achieve multi-view modeling features. In another study [4], Scheidgen proposed a multi-view modeling approach using a parsing tree technique to provide textual editors for Graphical Modeling Framework (GMF) based editors. Although these studies provide supporting features for blended modeling, some essential characteristics of blending, such as seamless synchronization between graphical and textual editors, are missing.

While multi-view modeling aims at defining viewpoints/views, blended modeling aims at providing a powerful multi-notation characterization that may be used to define viewpoints/views. Maro et al. [5] proposed a blended modeling methodology for Ericsson Hive -– a proprietary UML-based DSML. Particularly, authors proposed an approach to automatically generate an Ecore DSML from the Hive profile, exploited the features of Xtext to provide textual modeling for Hive, and achieved synchronization between graphical and textual notations. However, the synchronization is based on two separate resources (i.e., UML and Xtext) and rely on fixed bi-directional transformations. Addazi et al. [6] proposed a blended modeling approach for UML profiles where synchronization between graphical and textual notations is achieved through one single underlying resource (i.e., UML), thus minimizing the need for complex synchronization transformations. Recently, Latifaj et al. [7] proposed blended modeling support

for UML-RT state machines through graphical and textual notations in the HCL RTist environment[1].

To summarize, existing approaches either focus on approaches orthogonal to blended modeling, e.g. multi-view modeling, or provide partial blended modeling solutions for specific languages. Furthermore, existing studies do not target certain core blended modeling aspects, such as mechanism to generate both graphical and textual notations automatically, nor mechanisms to record domain expert's feedback during the mapping process between notations to ensure a semantically accurate synchronization between them. To conclude, there is a dire need for a unified, automated and customizable blended modeling solution that is not dependent on a single language, but rather applicable to multiple languages and application domains [8].

III. Implementation

Duration of the tutorial. The tutorial is intended to run for half-day given its nature and its implementation. In addition, from experience, full-day tutorials tend to be less attended and tire participants.

Preliminary schedule and tutorial description.

- Introductory presentation and contextualization of the tutorial (presentation): 30 minutes
- Introduction to the technical solution (demo): 30 minutes
- Architecting a blended modelling environment in Eclipse starting from an Ecore-based domain-specific modelling language (demo): 75 minutes
  - Definition of mapping rules between graphical and textual notations
  - Generation of the graphical and textual notations
  - Generation of the synchronisation transformations via higher-order transformations
  - Exploration of the resulting blended modelling environment
- Architecting software systems using the generated modelling environment in Eclipse: 75 minutes
  - Modelling of software architectures via graphical and textual notations
  - Synchronisation of model changes across notations
  - Collaborative interactive model editing with multiple modellers
  - Model differencing and merging
- Facilitated discussion and retrospective session (discussion): 45 minutes

Tutorial composition, notes and execution. The tutorial will conceive an initial short introductory presentation, a set of live demonstrations, and a final facilitated discussion and retrospective session. All presentations and demonstrations will be made publicly available on the web and the demonstrated tools will be made available as open-source as Eclipse plugins for download and further experimentation.

The tutorial will be run on-site, with the possibility for virtual participation if the conference organization requires so.

[1]https://www.hcltech.com/brochures/software/hcl-rtist

IV. Presenters' background

Federico Ciccozzi is an Associate Professor (Docent in Computer Science) at Mälardalen University (Västerås, Sweden), leader of the Automated Software language and Software engineering (ASSO) research group and Head of Research Education in Computer Science and in Electronics. His research focuses on several aspects of modelling, architecting and engineering complex (often embedded) systems with domain-specific languages. Among them, he specializes in: definition of modelling languages, automatic model manipulations through transformations, system properties preservation, just to mention a few. Moreover he conducts research in the area of multi-paradigm modeling, model versioning, (co)evolution and synchronization, as well as the application of MDE and CBSE techniques to complex cyber-physical systems systems. Federico Ciccozzi has co-authored 100+ publications, served in 40+ program committees and 35+ organization committees (including ICSA, ICSE, MODELS). He has run a technical briefing on engineering of robotic software at ICSE 2018.

Muhammad Waseem Anwar (postdoctoral researcher) and **Malvina Latifaj** (Ph.D. student), part of the ASSO research group led by Federico Ciccozzi, have heavily contributed to the technical contributions demonstrated in this tutorial as well as the preparation of the tutorial material and notes.

Ivano Malavolta is an Associate Professor in the Department of Computer Science and Director of the Network Institute, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands. His research interests include data-driven software engineering, with a special emphasis on software architecture, mobile software development, robotics software, and Model-Driven Engineering. He authored several scientific articles in international journals and peer-reviewed international conferences proceedings. He is program committee member and reviewer of international conferences and journals in the Software Engineering field and Associate Editor of IEEE Software (responsible for the Software Design and Architecture area). He received a Ph.D. in computer science from the University of L'Aquila, Italy. He is a Member of IEEE, ACM, VERSEN, Amsterdam Young Academy, and Amsterdam Data Science. He has run a technical briefing on engineering of robotic software at ICSE 2018 and a tutorial on Web-based hybrid mobile apps at MOBILESoft 2016.

Kousar Aslam (postdoctoral researcher), part of the Software and Sustainability group at the Vrije Universiteit Amsterdam, has heavily contributed to the technical contributions demonstrated in this tutorial as well as the preparation of the tutorial material and notes.

Acknowledgment

References

[1] F. Ciccozzi, M. Tichy, H. Vangheluwe, and D. Weyns, "Blended Modelling - What, why and how," in *Procs of MODELS Companion*. IEEE, 2019, pp. 425–430.

[2] A. Cicchetti, F. Ciccozzi, and A. Pierantonio, "Multi-view approaches for software and system modelling: a systematic literature review," *Software and Systems Modeling*, vol. 18, no. 6, pp. 3207–3233, 2019.

[3] M. Wimmer and G. Kramler, "Bridging grammarware and modelware," in *Procs of MoDELS*. Springer, 2005, pp. 159–168.
[4] M. Scheidgen, "Textual modelling embedded into graphical modelling," in *Procs of ECMFA*. Springer, 2008, pp. 153–168.
[5] S. Maro, J.-P. Steghöfer, A. Anjorin, M. Tichy, and L. Gelin, "On integrating graphical and textual editors for a uml profile based domain specific language: an industrial experience," in *Procs of SLE*, 2015, pp. 1–12.
[6] L. Addazi and F. Ciccozzi, "Blended graphical and textual modelling for uml profiles: A proof-of-concept implementation and experiment," in *Journal of Systems and Software*, vol. 175. Elsevier, 2021.
[7] M. Latifaj, F. Ciccozzi, M. Mohlin, and E. Posse, "Towards automated support for blended modelling of uml-rt embedded software architectures," in *Proceedings of ECSA*, 2021.
[8] I. David, M. Latifaj, J. Pietron, W. Zhang, F. Ciccozzi, I. Malavolta, A. Raschke, J.-P. Steghöfer, and R. Hebig, "Blended modeling in commercial and open-source model-driven software engineering tools: A systematic study," *Software & Systems Modeling*, 2022.