Towards Augmented Enterprise Models as Low-Code Interfaces to Digital Systems

Hans-Georg Fill[0000-0001-5076-5341], Felix Härer[0000-0002-2768-2342], Fabian Muff[0000-0002-7283-6603], and Simon Curty[0000-0002-2868-9001]

University of Fribourg, 1700 Fribourg, Switzerland,
[firstname].[lastname]@unifr.ch,
WWW home page: http://www.unifr.ch/inf/digits

Abstract. Traditionally, enterprise models have been used for representing knowledge on all aspects of an organization. This aided not only in composing a holistic picture of the different layers of an enterprise in terms of its business model, products and services, business processes and IT architecture, but also for describing the inter-dependencies between the layers. Depending on the degree of formalization, algorithms may be applied to the models, e.g. for simulations. With the upcoming of low-code approaches in software engineering, we regard in this position paper how similar concepts may be integrated in enterprise engineering. In particular we regard augmented enterprise models as interfaces to digital systems and illustrate this view with approaches for semantic technologies, data analytics and blockchain platforms. It is envisaged that such approaches will aid domain experts in integrating digital technologies in their daily work practices.

Keywords: enterprise modeling, low-code development, digital systems

1 Motivation

With the constant technological advancement and the permanent introduction of new technologies, enterprises need to react quickly to new developments and analyze potential effects on their business operations [7,28]. For that purpose, new technologies have to be rapidly understood and adopted through integrating them in existing environments if necessary [10]. In order to assess the potential of a new technology, a systematic approach is required that considers all necessary business and technological aspects [23]. This can be achieved through enterprise modeling which takes an engineering-oriented approach for representing all relevant aspects of an organization including its business model, products and services, business processes and organizational structures, IT services and workflows, as well as the IT and physical infrastructure [40,39,17,42]. Some approaches in enterprise modeling take a holistic perspective in the form of complete frameworks, e.g. MEMO [3], 4EM [41], ArchiMate [43], whereas others focus on partial aspects, e.g. just on business models [48] or process-related aspects in BPMN [34].

In addition, enterprise modeling has been regarded on different levels of formalization [4]. Whereas some approaches can be characterized as partially formal or semi-formal as they provide an unambiguous definition of the syntax of a modeling language and describe its semantics in natural language, e.g. as found in the iStar framework [29] or related approaches for goal modeling, others operate on a level where both structure and behavior are formally specified, e.g. the Semantic Object Model (SOM) [11] or Heraklit [13]. Furthermore, formal specifications may be added during run-time, e.g. via annotations [15,14], which serves the purpose of adding further processing capabilities such as semantic reasoning or for realizing simulations.

From the perspective of software systems development, the concept of so-called *low-code development platforms* recently gained attention [8]. Low-code platforms can be regarded as the next step in the evolution of techniques for creating software applications with less effort for writing programming code. This has been addressed previously in model-driven engineering and model-driven architectures (MDE/MDA) [12] and computer-aided software engineering (CASE) [45]. Today, these platforms are run in cloud environments and make use of dynamic graphical user interfaces, visual diagrams and declarative languages for realizing fully operational applications [8]. Whereas model-driven engineering and CASE went into similar directions, low-code adds a new perspective by not aiming for a complete elimination of the need to write code (no-code), but stresses the possibility of adding code segments where necessary. In addition, the topic of low-code approaches is characterized by recent increased interest from the side of industry [7].

As both enterprise modeling and low-code development often revert to graphical models, we will discuss in the following how the concepts of low-code development may be joined with enterprise modeling. The main idea thereby is to create enterprise modeling platforms that are capable of directly interacting with digital systems. Whereas some areas in enterprise modeling have followed such ideas early on, e.g. for executing business process models using workflow [22] or simulation engines [2], or the introduction of MDE approaches for dedicated business applications [27] and the derivation of requirements for software systems from enterprise models [49], most types of enterprise models today do not target the interaction with technical systems but stay on a conceptual level. The augmentation of enterprise models with interfaces to digital systems has the potential to enhance the productivity of domain experts and decision makers by combining the conceptual capabilities of enterprise models with machine-processing features.

The remainder of the paper is structured as follows. In Section 2 we will review the main characteristics of low-code software development approaches. This is followed by a discussion on how augmented enterprise models can act as low-code interfaces to digital systems and illustrating the idea with examples from the areas of semantic technologies, data analytics, and blockchains in Section 3. Subsequently, requirements for augmenting enterprise models are derived in Section 4 and a conclusion and outlook are given in Section 5.

2 Characteristics of Low-code Software Development

Whereas *no-code software development* aims at creating software applications entirely without writing any code in a programming language [8], *low-code development* takes a slightly relaxed perspective and is directed towards simplifying software development with *less* coding effort [44]. That means that low-code platforms may still require some coding knowledge or specialist skills, e.g. for entering mathematical formulas, but have the objective of shifting programming tasks from software engineers to domain experts [6]. For that purpose they revert to cloud-based architectures for the easy versioning, deployment and monitoring of applications via SaaS or PaaS, visual and domain-specific languages, and machine-learning techniques for assistance functionalities [44,6]. A particular advantage of both no-code and low-code approaches is seen in the possibility of enabling domain experts to create software applications and thus address the rising demand for software that so far required extensive coding skills [8,6].

Examples of no-code and low-code platforms can today be found across many domains. A recent survey showed a number of approaches for the visual encoding of blockchain applications and smart contracts [25], and from an industrial perspective, low-code approaches are investigated today especially for intelligent business process management, robotic process automation, or citizen automation and development [21].

3 Augmented Enterprise Models as Low-Code Digital Interfaces

When applying the idea of low-code software development to the realm of enterprise engineering [24,46], the following analogies can be found. Similar to the field of software engineering and the coding of software applications, also the interaction with state-of-the-art digital systems in enterprise engineering requires highly specialized knowledge. Consider for example the complexity of configuring a data analytics engine for applying machine-learning algorithms to sales data, the interaction with a blockchain platform for the realization of a decentralized autonomous organization, or the use of semantic technologies for classifying textual information.

In recent years, several efforts have thus been made for enabling domain experts in enterprise engineering with no or little coding knowledge to use such digital systems using no-code or low-code approaches. In contrast to software engineering it is thereby not aimed at creating arbitrary applications but rather at configuring and interacting with existing software systems. Examples include visual configuration platforms for machine learning and data analytics such as Knime, Azure Machine Learning Designer or RapidMiner Studio [1,31,36], the visual specification of semantic rules [35], or visual languages for production automation in Industry 4.0 scenarios [47,5].

However, these approaches have been mostly treated separately and were not integrated with enterprise models [46]. Such an integration would permit

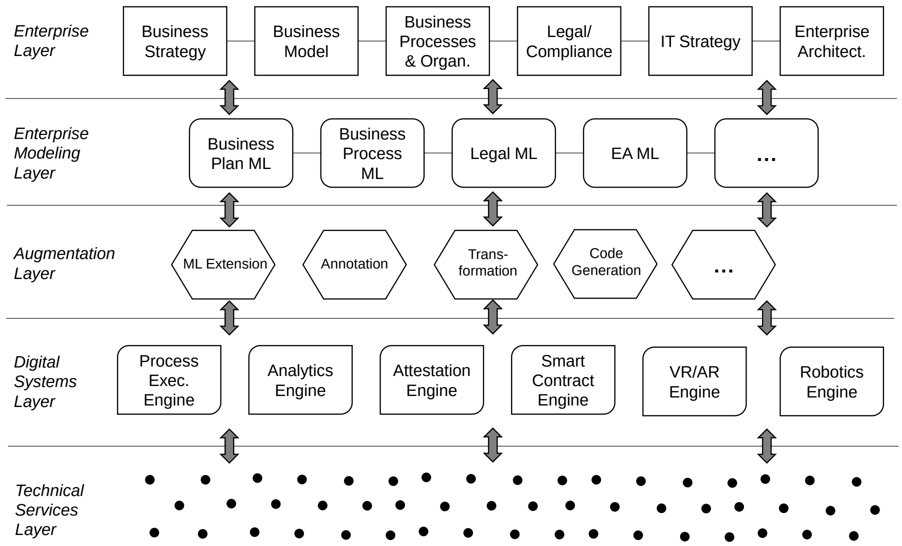

Fig. 1. Layers in Augmented Enterprise Modeling

the seamless derivation of requirements for example in data analytics from the business strategy and the direct provision of according data.

For achieving this integration, we propose the following layers as depicted in Figure 1. On the *Enterprise Layer*, the business aspects are shown that are relevant for enterprise engineering. Subsequently, the *Enterprise Modeling Layer* contains formal and semi-formal modeling languages (ML) for formalizing the business aspects. This permits for example to transform descriptions of business plans or business processes into machine-processable representations. Thereby, the degree of formalization determines the applicability and types of algorithms to be used. Whereas models whose behavioral properties are formally specified may be easily processed by algorithms, semi-formal models will permit only the application of selected algorithms [4]. For joining enterprise models with digital systems, we further propose an *Augmentation Layer*. This layer serves for the augmentation of enterprise models towards enabling a low-code interaction with digital systems. This can be achieved for example via the extension of modeling languages, the annotation of models, their transformation or the generation of code from models. The goal of this layer is to abstract from the technical details required for the interaction with the digital systems and embed their functionalities within the environment of the domain experts. The results of this augmentation then permit the interaction on the *Digital Systems Layer* with systems such as process execution or analytics engines, novel forms of attestation or smart contract engines or engines for the interaction with physical hardware such as VR/AR equipment or robotics to name some examples. Finally, the *Techni-*

cal Services Layer stands for the concrete implementations behind the digital systems, which typically follows today the paradigm of service orientation [9].

For illustrating how augmented enterprise models can be used in practical scenarios, we discuss in the following three examples where this approach has been already successfully applied. The first example presents the use of annotation models for enabling reasoning with semantic technologies, the second example the integration of functionalities from a data analytics platform in a business process improvement approach and the third example the integration of a blockchain-based attestation engine for enterprise models.

3.1 Example: Reasoning with Semantic Technologies

With semantic technologies, different kinds of classification and reasoning tasks can be accomplished. In particular, ontology and rule models permit to define formal conceptualizations of domains based on axioms. Subsequently, reasoners and rule engines can be applied for deriving new facts or classifying existing information. In the SeMFIS approach, enterprise models are augmented through annotation models that reference concepts in ontology and rule models [15,35] - see Figure 2. The thus enriched models are then transformed into formats that can be processed by reasoners and rule engines. The resulting classifications or new facts are subsequently fed back into the enterprise models or may be used for interactions with digital systems. Thereby, all information is expressed visually without the need for coding.

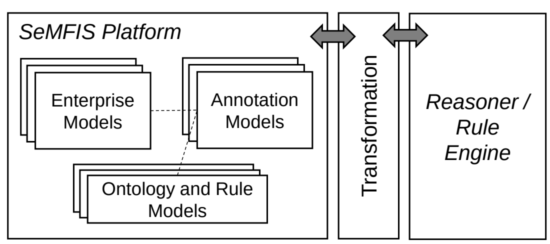

Fig. 2. Architecture of the SeMFIS Annotation Approach

3.2 Example: Integrating Data Analytics

For the integration of data analytics functionalities in enterprise modeling, the RUPERT approach for business process improvement reverted to the architecture shown in Figure 3 [30,20]. Through linking enterprise models to so-called statistical interface models and from these to according data sources, code for the R analytics platform is generated. In the statistical interface models it can be chosen from different kinds of pre-defined data analyses with the optional specification of parameters. The results generated by the R platform are then delivered back to the modeling platform and represented graphically.

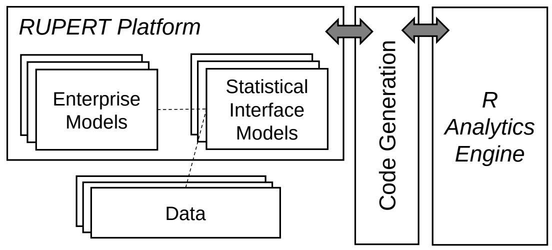

Fig. 3. Architecture of the RUPERT Approach for Integrating Data Analytics

3.3 Example: Attestation using Blockchains

In enterprise engineering, digital artifacts in the form of enterprise models are at the core of planning the behavior and structure of processes, decisions, and any other domain-specific reality captured by models. For allowing the integration of models with digital systems to a greater degree, attestation permits the creation of dependable and binding models possessing a contractual character [26]. Building on smart contract technologies and blockchains, the attestation of process steps, business services, or other artifacts can certify their exact state, timestamps, instance-level changes, and author information. In contrast to the traditional way of certifying documents with a trusted third party (TTP), blockchains here permit the immutable recording and decentralized validation of models and instances through an attestation engine acting as TTP - see Fig. 4.

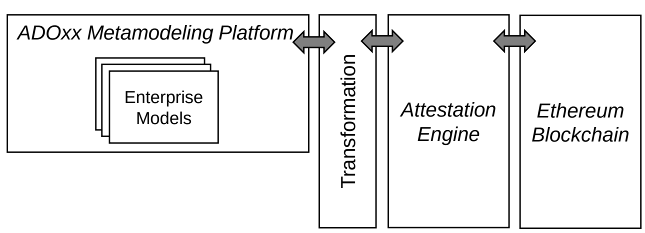

Fig. 4. Attestation Architecture for ADOxx

In this way, the certification of information such as the recording of processes and instances becomes possible in a distributed setting. When using technical systems such as for process execution, other engines can dependably rely on the state and progression of artifacts. Further applications include the verifiable issuance of certifications on the domain level - e.g. issuing degrees from universities to students [26] - and the attestation of technical artifacts such as machine learning algorithms, ontologies, or abitrary types of enterprise models [19,16,18].

4 Requirements for Augmenting Enterprise Modeling

When aiming for an augmentation of enterprise models towards their use as low-code interfaces to digital platforms, we can formulate the following three requirements based on the experiences gained with the implementations shown above.

First, the parts of enterprise models that shall act as interfaces to digital systems respectively the extensions of enterprise models need to be *sufficiently formally specified* [4]. This is necessary to have a clear view on the required behavior and ease the design of algorithms for processing the contained information [24]. The degree of formalization is thereby inter-dependent with the algorithms - i.e., for an algorithm that shall execute a process-oriented model, a more elaborate formal specification is necessary than for an algorithm that shall just derive the graphical representation of a model or that extracts some parameters for configuring a system.

Further, the digital systems that shall act as the target of the interaction need to have well-accessible and well-specified interfaces. Ideally, they provide an *openly-accessible API* or *well-specified exchange formats*. Today this is typically accomplished using REST interfaces and standardized exchange formats such as XML or JSON [37]. The behavior of the APIs needs to be well-documented and endpoints have to be accessible in a granularity that is suited for the interaction with the modeling tools. In case the granularity is not adequate for a direct coupling, interfaces or even separate engines for transforming the content of models have to be added separately.

Finally, a major challenge is to find a good trade-off between an adequate *user experience* for interacting with the technologies from the perspective of the domain experts, the scope of the provided *functionalities* and the *technical efficiency* of the interaction in the form of algorithms. Whereas users require a certain redundancy in the encoding of information for reducing interpretation errors, c.f. [32], algorithms require unambiguous, mathematical specifications in formats that are efficient for processing. Such design decisions therefore require engineering-oriented approaches and can hardly be achieved through purely scientific approaches [46].

5 Conclusion and Outlook

In this paper we discussed how enterprise models can be augmented in a way that is similar to low-code approaches in software engineering. The goal is to enable domain experts to directly interact with digital systems without requiring extensive coding knowledge.

One field that we will target in the future are enterprise modeling applications in the field of VR and AR where first approaches for displaying ER models and legal models using augmented reality are available [38,33]. With the help of the Augmentation Layer and the Digital System Layer, AR-based enterprise modeling tools can be designed, in which the underlying language from the Enterprise Modeling Layer is extended or annotated and automatically transformed and represented in VR or AR with the help of a VR/AR engine. This would permit to create direct interfaces between enterprise models and the physical or virtual world, e.g. for displaying the content of enterprise models in the physical space of the user or for immersing users in the knowledge space of enterprise models.

References

1. Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., Wiswedel, B.: KNIME - the konstanz information miner: version 2.0 and beyond. SIGKDD Explor. 11(1), 26–31 (2009)
2. Bocciarelli, P., D'Ambrogio, A., Giglio, A., Paglia, E.: Bpmn-based business process modeling and simulation. In: 2019 Winter Simulation Conference (WSC). pp. 1439–1453 (2019)
3. Bock, A., Frank, U.: Multi-perspective enterprise modeling—conceptual foundation and implementation with adoxx. In: Domain-specific conceptual modeling, pp. 241–267. Springer (2016)
4. Bork, D., Fill, H.: Formal aspects of enterprise modeling methods: A comparison framework. In: HICSS 2014. pp. 3400–3409. IEEE (2014)
5. Bork, D., Fill, H., Karagiannis, D., Utz, W.: Simulation of multi-stage industrial business processes using metamodelling building blocks with adoxx. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. 13, 333–344 (2018)
6. Bucchiarone, A., Ciccozzi, F., Lambers, L., Pierantonio, A., Tichy, M., Tisi, M., Wortmann, A., Zaytsev, V.: What is the future of modeling? IEEE Softw. 38(2), 119–127 (2021)
7. Burke, B.: Top Strategic Technology Trends for 2021. Gartner (2020)
8. Caballar, R.D.: Programming without code: The rise of no-code software development. IEEE Spectrum (March 11, 2020) (2020)
9. Demirkan, H., Kauffman, R.J., Vayghan, J.A., Fill, H., Karagiannis, D., Maglio, P.P.: Service-oriented technology and management: Perspectives on research and practice for the coming decade. Electron. Commer. Res. Appl. 7(4), 356–376 (2008)
10. Denning, P.J., Lewis, T.G.: Technology adoption. Commun. ACM 63(6), 27–29 (2020)
11. Ferstl, O.K., Sinz, E.J.: Modeling of business systems using som. In: Handbook on architectures of information systems, pp. 347–367. Springer (1998)
12. Fettke, P., Loos, P.: Model driven architecture (MDA). Wirtsch. 45(5), 555–559 (2003)
13. Fettke, P., Reisig, W.: Modelling service-oriented systems and cloud services with heraklit. In: Advances in Service-Oriented and Cloud Computing - International Workshops of ESOCC 2020. vol. 1360, pp. 77–89. Springer (2020)
14. Fill, H.G.: Using semantically annotated models for supporting business process benchmarking. In: International Conference on Business Informatics Research. pp. 29–43. Springer (2011)

15. Fill, H.: SeMFIS: A flexible engineering platform for semantic annotations of conceptual models. Semantic Web 8(5), 747–763 (2017)
16. Fill, H.: Applying the concept of knowledge blockchains to ontologies. In: Martin, A., Hinkelmann, K., Gerber, A., Lenat, D., van Harmelen, F., Clark, P. (eds.) Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2019) Stanford University. CEUR Workshop Proceedings, vol. 2350. CEUR-WS.org (2019)
17. Fill, H.: Enterprise modeling: From digital transformation to digital ubiquity. In: Federated Conference on Computer Science and Information Systems. Annals of Computer Science and Information Systems, vol. 21, pp. 1–4 (2020)
18. Fill, H.G., Härer, F.: Knowledge blockchains: Applying blockchain technologies to enterprise modeling. In: Proceedings of the 51st Hawaii International Conference on System Sciences (2018)
19. Fill, H.G., Härer, F.: Supporting Trust in Hybrid Intelligence Systems Using Blockchains. In: Proceedings of the AAAI 2020 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2020). vol. 2600. CEUR-WS, Stanford University, Palo Alto, California, USA (2020)
20. Fill, H., Johannsen, F.: A knowledge perspective on big data by joining enterprise modeling and data analyses. In: Bui, T.X., Jr., R.H.S. (eds.) Hawaii International Conference on System Sciences. pp. 4052–4061. IEEE (2016)
21. Gartner: Gartner forecasts worldwide low-code development technologies market to grow 23% in 2021 (2021), https://www.gartner.com/en/newsroom/press-releases/2021-02-15-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-23-percent-in-2021 last accessed 2021-03-30
22. Geiger, M., Harrer, S., Lenhard, J., Wirtz, G.: Bpmn 2.0: The state of support and implementation. Future Generation Computer Systems 80, 250–262 (2018)
23. Gong, Y., Janssen, M.: Roles and capabilities of enterprise architecture in big data analytics technology adoption and implementation. J. Theor. Appl. Electron. Commer. Res. 16(1), 37–51 (2021)
24. Grüninger, M., Fox, M.S.: The role of competency questions in enterprise engineering. In: Benchmarking—Theory and practice, pp. 22–31. Springer (1995)
25. Härer, F., Fill, H.G.: A Comparison of Approaches for Visualizing Blockchains and Smart Contracts. Jusletter IT Weblaw, ISSN 1664-848X February 2019 (2019)
26. Härer, F., Fill, H.: Decentralized attestation of conceptual models using the ethereum blockchain. In: Becker, J., Novikov, D.A. (eds.) 21st IEEE Conference on Business Informatics, CBI. pp. 104–113. IEEE (2019)
27. Henkel, M., Stirna, J.: Pondering on the key functionality of model driven development tools: The case of mendix. In: Forbrig, P., Günther, H. (eds.) Perspectives in Business Informatics Research. LNBIP, vol. 64, pp. 146–160. Springer (2010)
28. Hopkins, J.L.: An investigation into emerging industry 4.0 technologies as drivers of supply chain innovation in australia. Comput. Ind. 125, 103323 (2021)
29. Horkoff, J., Elahi, G., Abdulhadi, S., Yu, E.: Reflective analysis of the syntax and semantics of the i* framework. In: International Conference on Conceptual Modeling. pp. 249–260. Springer (2008)
30. Johannsen, F., Fill, H.G.: Meta modeling for business process improvement. Business & Information Systems Engineering 59(4), 251–275 (2017)
31. Microsoft: Azure machine learning designer (2020), https://azure.microsoft.com/de-de/services/machine-learning/designer/#product-overview last access 2021-03-31

32. Moody, D.: The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on Software Engineering 35(6), 756–779 (2009)
33. Muff, F., Fill, H.: Towards embedding legal visualizations in work practices by using augmented reality. Jusletter IT (forthcoming) (2021)
34. OMG: Business process model and notation 2.0.2 (2014), https://www.omg.org/spec/BPMN
35. Pittl, B., Fill, H.: A visual modeling approach for the semantic web rule language. Semantic Web 11(2), 361–389 (2020)
36. RapidMiner: Rapidminer studio (2020), https://docs.rapidminer.com/latest/studio/ last access 2021-03-31
37. Reitemeyer, B., Fill, H.: Future research directions for improved service modeling. In: Koschmider, A., Michael, J. (eds.) Enterprise Modeling and Information Systems Architectures 2021. forthcoming (2021)
38. Ruiz-Rube, I., Baena-Pérez, R., Mota, J.M., Sánchez, I.A.: Model-driven development of augmented reality-based editors for domain specific languages. IxD&A 45, 246–263 (2020)
39. Sandkuhl, K., Fill, H.G., Hoppenbrouwers, S., Krogstie, J., Leue, A., Matthes, F., Opdahl, A.L., Schwabe, G., Uludag, Ö., Winter, R.: Enterprise modelling for the masses – from elitist discipline to common practice. In: The Practice of Enterprise Modeling. pp. 225–240. Springer (2016)
40. Sandkuhl, K., Fill, H., Hoppenbrouwers, S., Krogstie, J., Matthes, F., Opdahl, A.L., Schwabe, G., Uludag, Ö., Winter, R.: From expert discipline to common practice: A vision and research agenda for extending the reach of enterprise modeling. Bus. Inf. Syst. Eng. 60(1), 69–80 (2018)
41. Sandkuhl, K., Stirna, J., Persson, A., Wißotzki, M.: Overview of the 4EM method. In: Enterprise Modeling, pp. 75–86. Springer (2014)
42. Shishkov, B.: Designing Enterprise Information Systems - Merging Enterprise Modeling and Software Specification. Springer (2020)
43. The Open Group: Archimate® 3.1 specification (2019), https://pubs.opengroup.org/architecture/archimate3-doc/
44. Tisi, M., Mottu, J., Kolovos, D.S., de Lara, J., Guerra, E., Ruscio, D.D., Pierantonio, A., Wimmer, M.: Lowcomote: Training the next generation of experts in scalable low-code engineering platforms. In: Bagnato, A., Brunelière, H., Burgueño, L., Eramo, R., Gómez, A. (eds.) STAF 2019 Co-Located Events Joint Proceedings. CEUR Workshop Proceedings, vol. 2405, pp. 73–78. CEUR-WS.org (2019)
45. Tolvanen, J.P., Rossi, M.: Metaedit+ defining and using domain-specific modeling languages and code generators. In: ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications. pp. 92–93. ACM (2003)
46. Vernadat, F.: Enterprise modelling: Research review and outlook. Computers in Industry 122, 103265 (2020)
47. Vještica, M., Dimitrieski, V., Pisarić, M., Kordić, S., Ristić, S., Luković, I.: An application of a dsml in industry 4.0 production processes. In: Lalic, B., Majstorovic, V., Marjanovic, U., von Cieminski, G., Romero, D. (eds.) Advances in Production Management Systems. The Path to Digital Transformation and Innovation of Production Management Systems. pp. 441–448. Springer (2020)
48. Wieland, M., Fill, H.G.: A domain-specific modeling method for supporting the generation of business plans. In: Modellierung 2020. GI LNI (2020)
49. Zikra, I., Stirna, J., Zdravkovic, J.: Bringing enterprise modeling closer to model-driven development. In: The Practice of Enterprise Modeling - 4th IFIP WG 8.1 Working Conference. LNBIP, vol. 92, pp. 268–282. Springer (2011)