

Integrating Model-Driven Engineering with Machine Learning for Intelligent Systems: Literature Review [†]

Kaouthar Elgueddari ^{1,*}, Zineb Aarab ^{1,2,3} , Achraf Lyazidi ²  and Adil Anwar ¹

¹ Information Systems and Web (SIWEB), Mohammadia School of Engineers, Mohammed V University, Rabat 10000, Morocco; aarab@emi.ac.ma (Z.A.); anwar@emi.ac.ma (A.A.)

² Research Laboratory in Computer Science and Telecommunications (LRIT), Faculty of Sciences, Mohammed V University, Rabat 10000, Morocco; lyazidi.achraf@gmail.com

³ Research Laboratory in Universal Systems Engineering (GENIUS), SUPMTI of Rabat, Rabat 10000, Morocco

* Correspondence: elguedarik@gmail.com

[†] Presented at the 7th edition of the International Conference on Advanced Technologies for Humanity (ICATH 2025), Kenitra, Morocco, 9–11 July 2025.

Abstract

The rapid development of intelligent systems has created a need for techniques capable of handling complexities and developing in an automatic way. The integration of machine learning in model-driven engineering (MDE) offers several advantages for the development and improvement of complex and intelligent systems. While machine learning (ML) also offers robust techniques and MDE has systematic approaches aimed at code generation and abstraction, in this review, while presenting the principles of MDE and ML, the article also critically explores the integration of ML in MDE. Starting with the fundamental concepts of MDE, then the principles and algorithms of ML, the focus of the discussion is on how machine learning techniques can improve model-driven engineering processes. By presenting the motivations for their combined use in the development of intelligent systems, based on the recent literature, the article describes the challenges and potential future directions, noting that the integration of machine learning into model-driven engineering not only accelerates development but also enhances the adaptability and performance of intelligent and complex systems, making it an increasingly relevant approach to addressing the complexities of modern intelligent systems.

Keywords: machine learning; model-driven engineering; literature review; intelligent systems



Academic Editors: Brahim El Bhiri
and Ayman S. Mosallam

Published: 5 November 2025

Citation: Elgueddari, K.; Aarab, Z.; Lyazidi, A.; Anwar, A. Integrating Model-Driven Engineering with Machine Learning for Intelligent Systems: Literature Review. *Eng. Proc.* **2025**, *112*, 67. <https://doi.org/10.3390/engproc2025112067>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The emergence of complex and intelligent systems requires the development of methodologies that can cope with both the complexity of system requirements and the speed of technological innovation. MDE and ML have emerged as two essential paradigms to address these challenges. MDE focuses on raising the level of abstraction of software development by using models as primary artifacts, thereby simplifying design, validation, and code generation. In contrast, ML focuses on extracting patterns and knowledge from large datasets, thereby enabling systems to learn and adapt over time. The beneficial possibilities are offered by the joint approach of these paradigms. By integrating machine learning techniques into the work of model-driven engineering, it can lead to improved automation, enhanced predictive capabilities, and the development of systems capable of handling random scenarios. The article provides an in-depth exploration of the theoretical foundations of MDE and ML. It further details how linking the approaches can contribute

to the evolution of intelligent systems. The review is structured into three main sections: the first section discusses the principles of MDE, the second section provides an overview of ML, and the third section presents a comprehensive analysis of the integration of MDE and ML. In doing so, this work aims to provide researchers and practitioners with a synthesized understanding of the current state of the art, as well as directions for future research.

2. Model-Driven Engineering

“MDE is a software development approach that emphasizes using models as a primary means for specifying, designing, and implementing software systems. A simple and popular formula used to describe MDE is the following: Models + Transformations = Software” [1,2]. To provide important context, we present the key concepts involved. “A model is an abstract representation of a system that eliminates unnecessary details to increase the users’ understanding of the system under consideration” [1]. A metamodel, in turn, is a specific type of model that defines the structure and rules of a modeling language; it serves as the abstract syntax of that language. Model transformations are specialized programs that convert one model into another or into textual representations (source code is an example), based on predefined transformation rules. Transformations are of two types: model-to-model, and model-to-text, where the input is a model and the output is textual code in languages such as Java.

MDE is a software development approach based on the construction and use of high-level abstract models; these models represent various aspects of a system. Models can be analyzed in an organized manner, validated, and transformed using dedicated tools to produce executable and efficient software. The primary goal of MDE is to improve development productivity and software quality by enabling a more abstract design process, thus reducing the gap between system specification and implementation. MDE is particularly beneficial for large and complex systems, where traditional development methods can become inefficient. It uses modeling languages such as UML or SysML and domain-specific languages (DSLs), as well as supporting tools such as model checkers and transformation engines for M2M and M2T processes [1–3].

3. Machine Learning

ML is a subfield of artificial intelligence (AI) that focuses on developing algorithms that can learn from data. ML allows systems to improve their performance over time without being explicitly programmed for each task. ML is categorized into different types of learning. There is unsupervised learning, which identifies patterns in unlabeled data through strategies such as clustering and dimensionality reduction, commonly applied to customer segmentation and document categorization; and there is supervised learning, which is used for classification and prediction tasks, based on labeled data to map inputs to desired outputs. Algorithms such as decision trees, support vector machines, and artificial neural networks fall under this category. Semi-supervised learning combines principles of both supervised and unsupervised approaches, making it useful when labeled data is sparse. Reinforcement learning works on a reward-based system, hence it is usable to improve and develop systems and applications, such as robotics, autonomous driving, and AI games [4].

4. Advantages of Integrating ML into MDE

Integrating machine learning into model-driven engineering (see Figure 1) offers several advantages. For example, enhanced automation, whereby developers automate code generation in an efficient and more convenient way, based on the use of machine learning algorithms to predict the greatest model transformation paths. Improved adaptability

is mandatory to adapt intelligent systems to changing and developing contexts as well. Incorporating machine learning into model-driven engineering workflows enables systems to learn from operational data and adjust models accordingly. Also, bridging the abstraction gap is an essential benefit for integrating ML into MDE. ML requires detailed data and algorithms, but MDE often operates at high levels of abstraction. Integration enables a bi-directional flow where high-level models guide the use and configuration of ML algorithms, and insight from ML inform model refinement and increase robustness.

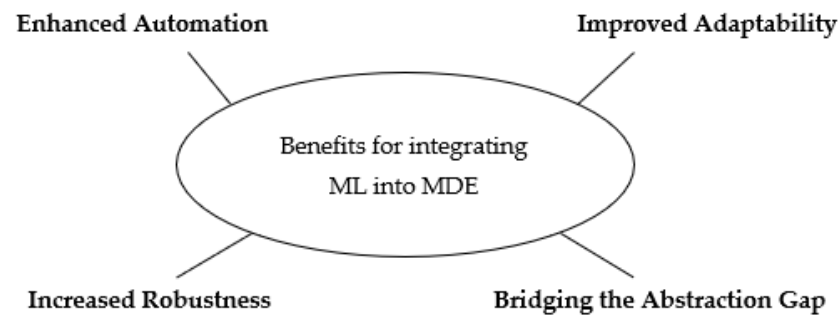


Figure 1. Benefits of integrating ML into MDE.

4.1. Approaches to Integration MDE and ML

Integration of ML into modeling, in particular modeling languages, should be extended to enable seamless incorporation of both model-learned attributes and “default” ones. Modeling language must be specific in a fine-grained manner about what must be learned, similar to how machine learning algorithms learn, and also specify why something must be learned. Learning units should be independently computable and updateable for it to be appropriate for learning. So, there is a meta-meta model to define and specify this weaving because a meta model defines what can be represented in meta models that conform to it, or what notions may be expressed in a real meta model. This makes it possible for domain modes to convey learning issues. This allows us to build a concrete modeling language that includes the constructs required to integrate machine learning into domain modeling [5]. In this approach, there are micro learning units that are usable for small fine-grained learning units and the micro learning units for decomposed and structured complex learning tasks with reusable and computable elements. Modeling language: The modeling language for domain data, and its structure and associated learning unit. To avoid ambiguities in usable language to specify this language, the following definitions for the latest developments in meta-modeling language served as inspiration for the language. For example: UML, SysML [5,6].

4.2. Addressing ML Development Challenges with Model-Driven Engineering (MDE)

It might be difficult to develop and manage systems that use ML models and components. Immature requirement definition, continuously changing data, a lack of domain knowledge in machine learning, integration with conventional software, responsible usage of machine learning, and the deployment and upkeep of ML models are some of the components of this complexity. These complexities present a number of challenges. Nils Baumann et al. [7] explain this well; for example, regarding the difficulty of managing evolving datasets, machine learning engineers must manually combine new and old datasets and retrain the entire machine learning model. According to Benjamin Jahi et al. [8], describing the dataset and neural network well to meet customer expectations is difficult; according to Benjamin et al., creating a highly efficient machine learning pipeline is a very difficult and complex job too; data scientists need to be skilled and experienced to sort through a variety of data preprocessing and machine learning models and choose the best one. Kaan Koseler et al. [9] discuss the challenges developers face when trying to apply machine learning

techniques with large amounts of data; developers must be familiar with the problem space, domain, and concepts of machine learning. To successfully and efficiently manage and control these problems, solutions are needed. Software models are used to guide the development and administration of ML components, demonstrating the synergy between MDE and ML development. This is not the same as AI or machine learning for MDE (AI4MDE), which helps users with modeling and related tasks using highly intelligent agents and recommendation systems. There are several very potential benefits for developers using MDE for AI-based systems (MDE4ML), including less complexity, time, and effort. Through the abstraction and automation of MDE, AI can also be used by software engineers, subject matter experts, and AI novices. In addition, MDE can improve the quality of the AI-based system by making it easier to maintain, scale, reuse, and interoperate.

4.3. Integrating Machine Learning into Model-Driven Engineering: Approaches, Strategies, and Benefits

A group of researchers addressed the work on the integration of ML in model-driven engineering through their studies and presented various examples in an effective and smooth manner. Naveed et al. [1] did a systematic literature review (SLR) on the practice and use of MDE, improving and developing machine learning components. The research of these researchers highlights different structures and strategies to adapt model transformations to meet the important and specific requirements of machine learning, specifically regarding data flow management, training parameters, and evolving nature of machine learning models over time, by applying the principles of model-driven engineering. So, the research explains how automated transformations can streamline the machine learning development process and ensure better alignment within high-level designs and their corresponding implementations. [1]. On the other hand, Rädler et al. [10] has also explored the use of SysML to improve and generate ML code, presenting high-level system models on how they can be directly translated into executable and practical ML code. The approach by Rädler et al. [10] significantly facilitates the transition from design to implementation, ensuring that the generated code remains compliant with the initial system specifications. By applying SysML, complex ML workflows are defined by developers in a structured manner, thereby enhancing both the organization and efficiency of ML software development. Thus, this approach optimizes manual coding efforts and minimizes inconsistencies between conceptual models and final implementations [10]. On the other hand, Hartmann et al. [5] present a particularly representative example of the seamless integration of machine learning into domain modeling. To create models that are inherently aware of ML components, domain-specific languages (DSLs) are used in this approach. These models are enhanced to directly integrate machine learning algorithms, creating a direct mapping between model elements and AI features. According to the presented approach, the integration offers several key benefits. First, it improves the traceability and ease of system architecture, thus facilitating the updating and adaptation of models as technology evolves. It also significantly minimizes development time by automating parts of the process of generating and using machine learning models. Furthermore, the use of DSLs allows every aspect of the model to align closely with the underlying machine learning mechanisms, thus helping to improve consistency and interpretability during the system lifecycle [5].

4.4. Analysis of Scientific Studies in MDE

In this section we will present our analysis of some scientific studies of MDE based on the following criteria: languages, quality assessment, application domains, and methodologies. The following table (Table 1) presents a classification of the articles that are treated in this study. Articles are classified among the classification of ML and MDE, i.e., papers. Papers are classified by model transformation language: Xtend, Epsilon Generation Language

(EGL), MontiAnna/MontiArc generators, Acceleo, Atlas Transformation language (ATL), TouchCore, Apache Velocity, Langium, OPC UA code generator P45. Then, classification of papers is performed by application domain (intelligent systems): Healthcare, Agricultural and City and Energy, Manufacturing, Building, Environment, Transport. Finally, classification of papers is performed by modeling language for developing IoT applications: DSLs, UML, BPMN. The papers discuss the main challenges and issues of MDE. Others present how to use this approach to model systems. An article discusses the application of model transformations in the process of intelligent systems engineering.

Table 1. Classification of papers.

Criteria	Classification of Papers by ML with MDE	Classification of Papers by Model Transformation Language	Classification of Papers by Application Domain (Intelligent Systems)	Classification of Papers by Modeling Language for Developing IoT Applications
Papers	Naveed et al. (2024) [1] Kelly et al. (2024) [3] Lee et al. (2024) [4] Brambilla et al. (2024) [11] Ciccozzi et al. (2024) [12] López et al. (2024) [13]	XTend Jahić et al. (2023) [8] Ries et al. (2021) [14] García-Díaz et al. (2015) [15]		
		Epsilon Generation Language (EGL) Koseler et al. (2019) [9] Yohannis et al. (2022) [16] Kourouklidis et al. (2021) [17]	Healthcare Brambilla et al. (2017) [5] Mehrabi et al. (2022) [29] Meliá et al. (2021) [30] Kotronis et al. (2018) [31] Morin et al. (2016) [32] Mezghani et al. (2017) [33] Veščauskas et al. (2016) [34]	DSLs Veščauskas et al. (2016) [34] Erazo-Garzón et al. (2022) [48]
		MontiAnna/MontiArc generators Baumann et al. (2022) [7] Gatto et al. (2019) [19] Kusmenko et al. (2019) [20]	Agricultural City Energy Manufacturing Barriga et al. (2022) [35–37] Nepomuceno et al. (2020) [38] Ziaei et al. (2020) [39]	Ihirwe et al. (2021) [49] Karaduman et al. (2021) [37,50]
		Acceleo Espinosa et al. (2029) [21] Tabbiche et al. (2023) [22] Safdar et al. (2022) [23]	Building Berrouyne et al. (2022) [40] Barriga et al. (2022) [36,37] Berrouyne et al. (2020) [41] Kirchhof et al. (2022) [42] Alulema et al. (2021) [43]	UML Parri et al. (2021) [51] Karaduman et al. (2021) [50,52] Plazas et al. (2020) [53] Jahed et al. (2019) [54] Moreira et al. (2019) [55]
		Atlas Transformation language (ATL) Mili et al. (2012) [24] Santos et al. (2018) [25] Krstić et al. (2022) [26]	Environment Karaduman et al. (2020) [44] Asici et al. (2019) [45] Durmaz et al. (2017) [46]	BPMN Moreira et al. (2019) [55] Sosa-Reyna et al. (2018) [56]
		TouchCore Shi et al. (2022) [27]	Transport Anwer et al. (2020) [47] Berrouyne et al. (2022) [40]	SysML Costa et al. (2016) [57]
		Apache Velocity Hartmann et al. (2017) [5]		
		OPC UA code generator Shin et al. (2020) [28]		

5. Conclusions and Future Work

In this paper, we have presented the main roles of model-driven engineering and machine learning in the development of intelligent systems. MDE is a systematic approach to system development with an emphasis on abstraction, automation, and model transformations, and ML provides powerful methods to learn from data and adapt to changing environments. Therefore, developers can achieve a higher degree of automation, better adaptability, and increased robustness of the system, based on the integration of these two techniques. In our future work, we plan to further explore the integration of MDE with ML techniques. In particular, we aim to develop a framework that facilitates the automated generation and refinement of models through learning from existing data, and

to develop an innovative intelligent system in a specific field using our MDE approach and ML techniques.

6. Discussion and Research Perspectives

This study offered a clear and structured overview of the different current engineering approaches led by models (MDE) for smart systems. If it highlights the important advances already made, it also reveals several tracks that are still little explored. For example, we especially note the lack of link between the MDE and the explanatory automatic learning techniques (explaining ML). In a world where user transparency and confidence have become essential, integrating this explainable dimension in MDE processes seems to be a promising path. Future research could thus be interested in how to combine performance and understanding of systems by their users.

Another point that deserves more attention is that of automatic refinement of models. Despite progress in synthesis and transformation of models, few studies have really explored how to adapt the models automatically and continuously in the face of changes in needs or returns in real time. This area represents a great opportunity to develop MDE frameworks capable of dynamically adapting to evolutionary contexts.

In summary, it is important that future research does not only solve current technical problems, but that they also incorporate interdisciplinary approaches. By bringing the MDE closer to domains such as artificial intelligence or software engineering, we can expand horizons and significantly advance the design of smart systems of tomorrow.

Author Contributions: Conceptualization, K.E., Z.A., A.L. and A.A.; methodology, K.E., Z.A. and A.L.; validation, K.E., Z.A. and A.L.; formal analysis, K.E., Z.A. and A.L.; writing—original draft preparation, K.E.; writing—review and editing, K.E., Z.A. and A.L.; supervision, K.E., Z.A., A.L. and A.A.; project administration, K.E., Z.A., A.L. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Naveed, H.; Arora, C.; Khalajzadeh, H.; Grundy, J.; Haggag, O. Model driven engineering for machine learning components: A systematic literature review. *IEEE Trans. Softw. Eng.* **2024**, *46*, 912–930. [\[CrossRef\]](#)
2. Stahl, T.; Völter, M.; Czarnecki, K. *Model-Driven Software Development: Technology, Engineering, Management*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2006.
3. Kelly, S.; Tolvanen, J.-P. *Domain-Specific Modeling: Enabling Full Code Generation*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2008.
4. Lee, I.; Shin, Y.J. Machine learning for enterprises: Applications, algorithm selection, and challenges. *Bus. Horiz.* **2020**, *63*, 157–170. [\[CrossRef\]](#)
5. Hartmann, T.; Moawad, A.; Fouquet, F.; Le Traon, Y. The next evolution of MDE: A seamless integration of machine learning into domain modeling. *Softw. Syst. Model.* **2019**, *18*, 1285–1304. [\[CrossRef\]](#)
6. Object Management Group: OMG Unified Modeling Language, Version 2.5. 2015. Available online: <http://www.omg.org/spec/UML/2.5/PDF> (accessed on 28 September 2025).
7. Baumann, N.; Kusmenko, E.; Ritz, J.; Rumpe, B.; Weber, M.B. Dynamic data management for continuous retraining. In Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, Montreal, QC, Canada, 23–28 October 2022; pp. 359–366. [\[CrossRef\]](#)

8. Jahić, B.; Guelfi, N.; Ries, B. SEMKISDSL: A Domain-Specific Language to Support Requirements Engineering of Datasets and Neural Network Recognition. *Information* **2023**, *14*, 213. [\[CrossRef\]](#)
9. Koseler, K.; McGraw, K.; Stephan, M. Realization of a Machine Learning Domain Specific Modeling Language: A Baseball Analytics Case Study. In Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development, Prague, Czech Republic, 20–22 February 2019; pp. 13–24. [\[CrossRef\]](#)
10. Rädler, S.; Rupp, M.; Rigger, E.; Rinderle-Ma, S. Model-Driven Engineering for Machine Learning Code Generation using SysML. In Proceedings of the Modellierung 2024, Potsdam, Germany, 12–15 March 2024; pp. 197–212.
11. Brambilla, M.; Cabot, J.; Wimmer, M. *Model-Driven Software Engineering in Practice*; Morgan & Claypool: San Rafael, CA, USA, 2017.
12. Ciccozzi, F.; Malavolta, I.; Selic, B. Execution of UML models: A systematic review of research and practice. *Softw. Syst. Model.* **2019**, *18*, 2313–2360. [\[CrossRef\]](#)
13. López, J.A.H.; Cánovas Izquierdo, J.L.; Cuadrado, J.S. ModelSet: A dataset for machine learning in model-driven engineering. *Softw. Syst. Model.* **2022**, *21*, 967–986. [\[CrossRef\]](#)
14. Ries, B.; Guelfi, N.; Jahic, B. An mde method for improving deep learning dataset requirements engineering using alloy and uml. In Proceedings of the 9th International Conference on Model-Driven Engineering and Software Development, Online, 8–10 February 2021; pp. 41–52. [\[CrossRef\]](#)
15. Cueva-Lovelle, J.M.; García-Díaz, V.; Pelayo García-Bustelo, B.C.; Pascual-Espada, J. Towards a standard-based domain-specific platform to solve machine learningbased problems. *Int. J. Interact. Multimed. Artif. Intell.* **2015**, *3*, 351. [\[CrossRef\]](#)
16. Yohannis, A.; Kolovos, D. Towards model-based bias mitigation in machine learning. In Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems, Montreal, QC, Canada, 23–28 October 2022; pp. 143–153. [\[CrossRef\]](#)
17. Kourouklidis, P.; Kolovos, D.; Noppen, J.; Matragkas, N. A model-driven engineering approach for monitoring machine learning models. In Proceedings of the 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Fukuoka, Japan, 10–15 October 2021; pp. 160–164.
18. Al-Azzoni, I. Model Driven Approach for Neural Networks. In Proceedings of the 2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA), Valencia, Spain, 19–22 October 2020; pp. 87–94. [\[CrossRef\]](#)
19. Gatto, N.; Kusmenko, E.; Rumpe, B. Modeling deep reinforcement learning based architectures for cyber-physical systems. In Proceedings of the 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Munich, Germany, 15–20 September 2019; pp. 196–202. [\[CrossRef\]](#)
20. Kusmenko, E.; Nickels, S.; Pavlitskaya, S.; Rumpe, B.; Timmermanns, T. Modeling and training of neural processing systems. In Proceedings of the 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS), Munich, Germany, 15–20 September 2019; pp. 283–293. [\[CrossRef\]](#)
21. Espinosa, R.; García-Saiz, D.; Zorrilla, M.; Zubcoff, J.J.; Mazón, J.N. S3Mining: A model-driven engineering approach for supporting novice data miners in selecting suitable classifiers. *Comput. Stand. Interfaces* **2019**, *65*, 143–158. [\[CrossRef\]](#)
22. Tabbiche, M.N.; Khalfi, M.F.; Adjoudj, R. Applying Machine Learning and Model-Driven Approach for the Identification and Diagnosis Of COVID-19. *Int. J. Distrib. Syst. Technol. (IJ DST)* **2023**, *14*, 1–27. [\[CrossRef\]](#)
23. Safdar, A.; Azam, F.; Anwar, M.W.; Akram, U.; Rasheed, Y. MoDLF: A model-driven deep learning framework for autonomous vehicle perception (AVP). In Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems, Montreal, QC, Canada, 23–28 October 2022; pp. 187–198. [\[CrossRef\]](#)
24. Mili, S.; Meslati, D. Modeling bio inspired systems: Towards a new view based on MDE. *J. Theor. Appl. Inf. Technol.* **2012**, *38*, 63–73.
25. Santos, F.; Nunes, I.; Bazzan, A.L. Modeldriven agent-based simulation development: A modeling language and empirical evaluation in the adaptive traffic signal control domain. *Simul. Model. Pract. Theory* **2018**, *83*, 162–187. [\[CrossRef\]](#)
26. Krstić, D.; Petrović, N.; Al-Azzoni, I. Model-driven approach to fading-aware wireless network planning leveraging multiobjective optimization and deep learning. *Math. Probl. Eng.* **2022**, *2022*, 4140522. [\[CrossRef\]](#)
27. Shi, Y.; Kienzle, J.; Guo, J.L. Feature-oriented modularization of deep learning APIs. In Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings, Montreal, QC, Canada, 23–28 October 2022; pp. 367–374.
28. Shin, S.-J. An opc ua-compliant interface of data analytics models for interoperable manufacturing intelligence. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3588–3598. [\[CrossRef\]](#)
29. Mehrabi, M.; Zamani, B.; Hamou-Lhadj, A. HealMA: A model-driven framework for automatic generation of IoT-based Android health monitoring applications. *Autom. Softw. Eng.* **2022**, *29*, 56. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Meliá, S.; Nasabeh, S.; Luján-Mora, S.; Cachero, C. MoSIoT: Modeling and Simulating IoT Healthcare-Monitoring Systems for People with Disabilities. *Int. J. Environ. Res. Public Health* **2021**, *18*, 6357. [\[CrossRef\]](#) [\[PubMed\]](#)

31. Kotronis, C.; Nikolaidou, M.; Dimitrakopoulos, G.; Anagnostopoulos, D.; Amira, A.; Bensaali, F. A model-based approach for managing criticality requirements in e-health iot systems. In Proceedings of the 2018 13th Annual Conference on System of Systems Engineering (SoSE), Paris, France, 19–22 June 2018; pp. 60–67.
32. Morin, B.; Fleurey, F.; Husa, K.E.; Barais, O. A Generative Middleware for Heterogeneous and Distributed Services. In Proceedings of the 2016 19th International ACM SIGSOFT Symposium on Component-Based Software Engineering (CBSE), Venice, Italy, 5–8 April 2016; pp. 107–116.
33. Mezghani, E.; Exposito, E.; Drira, K. A model-driven methodology for the design of autonomic and cognitive IoT-based systems: Application to healthcare. *IEEE Trans. Emerg. Top. Comput. Intell.* **2017**, *1*, 224–234. [\[CrossRef\]](#)
34. Venčkauskas, A.; Štuikys, V.; Toldinas, J.; Jusas, N. A model-driven framework to develop personalized health monitoring. *Symmetry* **2016**, *8*, 65. [\[CrossRef\]](#)
35. Barriga, J.A.; Clemente, P.J.; Sosa-Sánchez, E.; Prieto, Á.E. SimulateIoT: Domain Specific Language to design, code generation and execute IoT simulation environments. *IEEE Access* **2021**, *9*, 92531–92552. [\[CrossRef\]](#)
36. Barriga, J.A.; Clemente, P.J. Designing and simulating IoT environments by using a model-driven approach. In Proceedings of the 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), Madrid, Spain, 22–25 June 2022; pp. 1–6.
37. Barriga, J.A.; Clemente, P.J.; Hernández, J.; Pérez-Toledano, M.A. SimulateIoT-FIWARE: Domain Specific Language to Design, Code Generation and Execute IoT Simulation Environments on FIWARE. *IEEE Access* **2022**, *10*, 7800–7822. [\[CrossRef\]](#)
38. Nepomuceno, T.; Carneiro, T.; Maia, P.H.; Adnan, M.; Nepomuceno, T.; Martin, A. AutoIoT: A framework based on user-driven MDE for generating IoT applications. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Virtual, 30 March–3 April 2020; pp. 719–728.
39. Ziaei, M.; Zamani, B.; Bohlooli, A. A Model-Driven Approach for IoT-Based Monitoring Systems in Industry 4.0. In Proceedings of the 2020 4th International Conference on Smart City, Internet of Things and Applications (SCIOT), Mashhad, Iran, 16–17 September 2020; pp. 99–105.
40. Berrouyne, I.; Adda, M.; Mottu, J.M.; Tisi, M. A Model-Driven Methodology to Accelerate Software Engineering in the Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 19757–19772. [\[CrossRef\]](#)
41. Berrouyne, I.; Adda, M.; Mottu, J.M.; Royer, J.C.; Tisi, M. A Model-Driven Approach to Unravel the Interoperability Problem of the Internet of Things. In *International Conference on Advanced Information Networking and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 1162–1175.
42. Kirchhof, J.C.; Rumpe, B.; Schmalzing, D.; Wortmann, A. Montithings: Model-driven development and deployment of reliable iot applications. *J. Syst. Softw.* **2022**, *183*, 111087. [\[CrossRef\]](#)
43. Alulema, D.; Criado, J.; Iribarne, L. An Approach to Integrate IoT Systems with No-Web Interfaces. In *International Conference on Information Technology & Systems*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 417–427.
44. Karaduman, B.; Challenger, M.; Eslampanah, R.; Denil, J.; Vangheluwe, H. Analyzing WSN-based IoT Systems using MDE Techniques and Petri-net Models. In *STAF Workshops; CEUR Workshop Proceedings*; Bergen, Norway, 2020; pp. 35–46.
45. Asici, T.Z.; Karaduman, B.; Eslampanah, R.; Challenger, M.; Denil, J.; Vangheluwe, H. Applying model driven engineering techniques to the development of contiki-based IoT systems. In Proceedings of the 2019 IEEE/ACM 1st International Workshop on Software Engineering Research & Practices for the Internet of Things (SERP4IoT), Montreal, QC, Canada, 27 May 2019; pp. 25–32.
46. Durmaz, C.; Challenger, M.; Dagdeviren, O.; Kardas, G. Modelling contiki-based IoT systems. In Proceedings of the 6th Symposium on Languages, Applications and Technologies (SLATE 2017), Vila do Conde, Portugal, 26–27 June 2017.
47. Anwer, H.; Azam, F.; Anwar, M.W.; Rashid, M. A model-driven approach for load-balanced MQTT protocol in internet of things (IoT). In *Complex, Intelligent, and Software Intensive Systems: Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019)*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 368–378.
48. Erazo-Garzón, L.; Cedillo, P.; Rossi, G.H.; Moyano, J. A Domain-Specific Language for Modeling IoT System Architectures That Support Monitoring. *IEEE Access* **2022**, *10*, 61639–61665. [\[CrossRef\]](#)
49. Ihrwe, F.; Di Ruscio, D.; Mazzini, S.; Pierantonio, A. Towards a modeling and analysis environment for industrial IoT systems. *arXiv* **2021**, arXiv:2105.14136. [\[CrossRef\]](#)
50. Karaduman, B.; Mustafiz, S.; Challenger, M. FTG+ PM for the Model-Driven Development of Wireless Sensor Network based IoT Systems. In Proceedings of the 2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Fukuoka, Japan, 10–15 October 2021; pp. 306–316.
51. Parri, J.; Sampietro, S.; Vicario, E. FaultFlow: A tool supporting an MDE approach for Timed Failure Logic Analysis. In Proceedings of the 2021 17th European Dependable Computing Conference (EDCC), Munich, Germany, 13–16 September 2021; pp. 25–32.
52. Karaduman, B.; Challenger, M. Model-driven development for ESP-based IoT systems. In Proceedings of the 2021 IEEE/ACM 3rd International Workshop on Software Engineering Research and Practices for the IoT (SERP4IoT), Madrid, Spain, 3 June 2021; pp. 9–12.

53. Plazas, J.E.; Bimonte, S.; de Vaulx, C.; Schneider, M.; Nguyen, Q.D.; Chanet, J.P.; Shi, H.; Hou, K.M.; Corrales, J.C. A conceptual data model and its automatic implementation for IoT-based business intelligence applications. *IEEE Internet Things J.* **2020**, *7*, 10719–10732. [[CrossRef](#)]
54. Jahed, K.; Dingel, J. Enabling model-driven software development tools for the internet of things. In Proceedings of the 2019 IEEE/ACM 11th International Workshop on Modelling in Software Engineering (MiSE), Montreal, QC, Canada, 26–27 May 2019; pp. 93–99.
55. Moreira, J.L. SEMIoTICS: Semantic Model-Driven Development for IoT Interoperability of Emergency Services: Improving the Semantic Interoperability of IoT Early Warning Systems. Ph.D. Thesis, University of Twente, Enschede, The Netherlands, 2019.
56. Sosa-Reyna, C.M.; Tello-Leal, E.; Lara-Alabazares, D.; Mata-Torres, J.A.; Lopez-Garza, E. A Methodology Based on Model-Driven Engineering for IoT Application Development. In Proceedings of the 12th International Conference on Digital Society and eGovernments (ICDS 2018), Rome, Italy, 25–29 March 2018.
57. Costa, B.; Pires, P.F.; Delicato, F.C.; Li, W.; Zomaya, A.Y. Design and analysis of IoT applications: A model-driven approach. In Proceedings of the 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Auckland, New Zealand, 8–12 August 2016; pp. 392–399.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.