AI-based and Model-driven Methods to Guide Citizens through Processes in the Public Sector

Lukas Netz[1,*], Judith Michael[2] and Bernhard Rumpe[1]

[1]RWTH Aachen University, Software Engineering, Ahornstraße 55, 52074 Aachen, Germany
[2]University of Regensburg, Programming and Software Engineering, Bajuwarenstraße 4, 93053 Regensburg, Germany

Abstract

Digitalization of processes in the public sector is a challenging endeavor for citizens: too long or ambiguous forms and unfamiliar terms make it challenging to participate in these processes. What is needed are methods to guide citizens with IT systems through processes in the public sector in a user-centric way. Within this paper, we investigate how to develop web systems that provide citizen-centered guidance in several dimensions: explanatory content, guided navigation, contextual process support, and form completion. Our solution integrates generative AI and model-driven engineering methods to develop web applications for the public sector. This has not only advantages for citizens, but we can also enhance the system maintenance of such systems with these methods. We present the main processes of how to realize such integrated approaches, provide some examples from practice, and discuss open challenges for the approaches.

Keywords

Public Processes, Model-Driven Engineering, Artificial Intelligence, Large Language Model, AI4SE, Human-centric systems

1. Introduction and Motivational Statement

Processes in the public sector are are often complex: Citizens frequently encounter unfamiliar terms and have to complete complex forms, making it hard to participate in these processes without guidance [1]. In addition to the complexity of the forms, citizens' limited abilities or their general reluctance to complete bureaucratic forms are present further barriers to participation [2]. Specific challenges include too long forms, they ask for redundant information, or they appear overly invasive and ambiguous. While the digitalization of governmental processes is intended to enhance the quality of service for citizens [3], this goal is not always achieved in practice. In many cases the processes are optimized for one particular institution or organizational unit [4] without adequately considering the citizen's perspective. As a result, the digital transformation fails fo significantly improve user experience or reduce the interaction complexity from the citizen's point of view.

Within this work, we aim to *develop web systems for the public sector that provide citizen-centered process support, explanation, and navigation.* We can state the following research questions:

RQ1: How can generative AI techniques, particularly Large Language Models, be integrated with model-driven engineering methods to provide user-centric guidance for citizens navigating complex public sector processes?

We propose an engineering approach for the creation of web-based systems that go beyond merely presenting forms to be completed by citizens. These systems are designed to offer integrated support through explanatory content, guided navigation, and contextual process support. This way, we can improve the accessibility and usability of administrative processes for citizens, particularly for those who face barriers due to complexity, unfamiliarity, or cognitive load. Our approach applies Generative Artificial Intelligence (genAI) techniques and Model-driven Engineering (MDE) methods in an integrated

AI4DPS: Artificial Intelligence for Digital Public Services, June 11, 2025, Koblenz, Germany
*Corresponding author.
netz@se-rwth.de (L. Netz); judith.michael@ur.de (J. Michael); rumpe@se-rwth.de (B. Rumpe)
0000-0003-2013-2919 (L. Netz); 0000-0002-4999-2544 (J. Michael); 0000-0002-2147-1966 (B. Rumpe)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-4122/invited1.pdf
CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2025-12-10

way. This allows to use the deterministic generative power of MDE approaches and its fast development cycles to react to changes in combination with dynamically generating natural language explanations, reformulation of complex terms, and adaptation of content presentation to user needs.

A key design principle of our approach is that the AI components act from the perspective of the citizen rather than reflecting only the structure or logic of administrative procedures. This means that the behavior of the system should not be solely determined or constrained by the administration; instead, it should reflect the informational and procedural needs of diverse user groups. In line with this, our approach allows for the tailoring of content and interactions to accommodate various user characteristics, i.e., different levels of knowledge, simplified language, technology-aversity, and low levels of digital iteracy. The second research question is defined as:

RQ2: What are the technical and organizational challenges and limitations of developing and deploying AI- and MDE-based systems for public administration, and how can these be mitigated?

Our paper is structured the following: The next section provides relevant background information, while section 3 presents some important requirements important for developing web-based services supporting public processes. section 4 presents the toolchains for creating AI assistants, providing intelligent explanations and form completion as well as system maintenance support for developers. section 5 discusses some challenges when realizing these approaches and the last section concludes.

2. Background and Related work

To put our approach into context, we explain relevant foundations of genAI, MDE, and their combination as well as give some examples of the application of AI in the public sector.

2.1. Generative Artificial Intelligence

Generative Artificial Intelligence refers to AI technologies capable of producing content such as text, images, or code based on patterns learned from large datasets [5]. Modern genAI systems, often based on transformer architectures like Large Language Models (LLMs) [6], have demonstrated remarkable proficiency in understanding and generating human-like language. These models can perform a variety of tasks, including summarization, translation, question answering, and even software-related tasks such as code synthesis [7, 8]. GenAI systems are typically pre-trained on massive corpora and can be adapted to specific tasks or domains via fine-tuning or prompt engineering. Their interaction style—natural language prompts and responses—lowers the barrier for non-experts to engage with complex systems. As such, genAI enables new forms of human-computer interaction, particularly in areas where traditional rule-based systems fall short. However, due to their probabilistic nature and dependence on training data, they also introduce challenges regarding reliability, transparency, and domain adaptation. Understanding both their potential and limitations is essential for safely integrating them into structured engineering environments.

2.2. Model-Driven Engineering

MDE is a software engineering approach that emphasizes the use of high-level models as central artifacts throughout the development process [9, 10]. Instead of writing low-level code directly, developers create abstract models that describe the structure, behavior, and rules of a system. These models are then transformed into executable software using code generators.

A concrete example of MDE is MontiGem [11, 12], a framework developed at RWTH Aachen University. MontiGem allows developers to model domain-specific information systems by combining domain models specified as UML class diagrams [4], user interface models [13], and access control rules. From these models, MontiGem automatically generates large parts of a web-based information

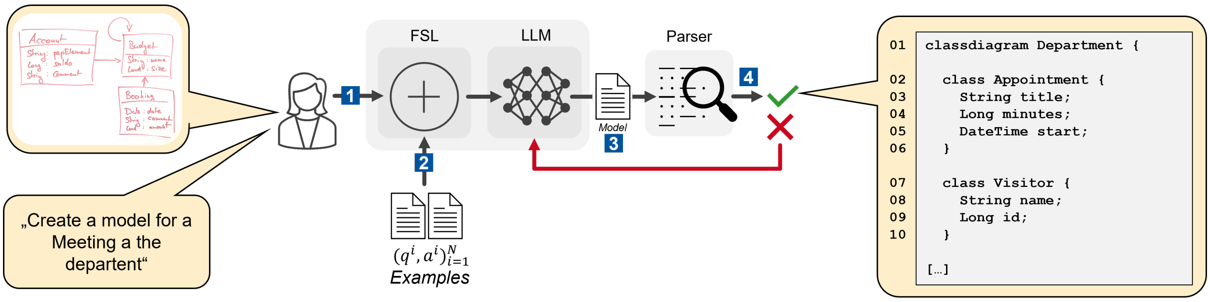


Figure 1: AI-Based Model-Driven Engineering: The conversational capabilities of an LLM can be used to transform Natural Language Input from a user (1) in combination with examples of the targeted modeling language (2) into a a running Model (3) that corresponds to the users description, and can be checked and validated (4).

system—including the backend, frontend, and even role-based access mechanisms—enabling rapid development and consistency across the application [14]. MontiGem is used to generate web applications, e.g., in research for representing processes extracted from event logs [15], or in practice to create the MaCoCo application [4], a software for handling public and third-party funding, their controlling, and staff management at university chairs that is in use since 2018.

Another example of MDE applied in public administration is the low-code development platform A12 [16, 17]: It realizes web applications using a client-server architecture. Besides several other application areas, it is used to create the tax forms in Germany together with a domain-specific language to define tax law [18, 19].

2.3. AI-based Model-driven Software Engineering

The integration of Artificial Intelligence, particularly generative AI, with Model-Driven Engineering [20] offers new capabilities for automating software development [21]. GenAI can be used to create models conforming to specific modeling languages - such as domain-specific languages for data structures or business processes—based [22, 23] on textual requirements or legacy code. These AI-generated models act as high-level abstractions, which can then be processed by model-driven toolchains to generate executable code, documentation, or test cases. This approach reduces manual modeling effort [24] and bridges the gap between informal requirements and formal model specifications. In addition, AI techniques can support model refinement, validation, and consistency checking across model layers.

2.4. AI in the Public Sector

Inspired by successful applications of Artificial Intelligence in the private sector, government agencies have begun exploring AI technologies in fields such as health, taxation, and education. However, further research is needed to fully realize the potential of AI in the public sector and to address pressing societal challenges. Androutsopoulou et al. [25] present an innovative approach in an ICT platform architecture that utilizes chatbots to improve government-to-citizen communications. There is a high potential for AI chatbots to enhance efficency of procedures in the public sector [26]. Especially in the context of digitalization [27], AI is seen as an accelerator and enabler [28].

3. Requirements for Application in the Public Sector

We highlight some of the requirements we were mainly considering when creating our solutions for developing web-based applications for the public sector. Please note that this list is only an excerpt and that it does not include the main function requirements coming from individual application cases.

Domain Knowledge Integration

Public sector processes are highly heterogeneous and can differ significantly between departments. Consequently, it is not feasible to rely on a generically trained large language model to support department-specific administrative workflows. Therefore, the proposed approach must incorporate a mechanism for integrating process-specific knowledge. Additionally, it must ensure that this tailored knowledge is not overridden or diluted by the generic, pretrained knowledge of the LLM, which may deviate from the requirements of the specific department.

Natural Language Support

In many cases, public sector processes are already documented in some form. An assistive approach should be capable of presenting this domain knowledge in a personalized and easily understandable manner. Furthermore, an interactive service can support follow-up questions and tailor responses to the specific needs of individual citizens. This is particularly beneficial for overcoming language barriers or for providing explanations at varying levels of detail, depending on the user's background and preferences.

Validation an Error Prevention

It is well documented that LLMs are prone to producing incorrect or fabricated information, commonly referred to as hallucinations [5, 29]. Therefore, the approach must include a mechanism to validate AI-generated content and detect potential errors. Since hallucinations cannot yet be reliably prevented, the system should be designed with the possibility of erroneous output in mind. Users or administrative staff should be informed about this limitation and involved in a verification step to ensure the accuracy of the provided information.

Modularity and Maintainability

Assisting citizens in navigating bureaucratic processes presents several challenges. Effective guidance is likely to require adaptation to the specifics of each individual process. To support this flexibility, a modular and agent-based approach is preferable over a monolithic architecture. Such a design enables the specialization of LLM interactions for distinct tasks, promoting more targeted and context-aware assistance.

Legal and Ethical Compliance

To be applicable in the public sector, any AI-based approach must adhere to relevant legal frameworks, including the EU AI Act [30] and data protection regulations. In addition to legal compliance, ethical considerations must be addressed. An AI system may interact with a diverse range of citizens, yet it is typically trained on largely opaque datasets that may contain biases and stereotypes. As a result, there is a risk that the system could generate guidance that disadvantages minority groups or suggests inappropriate actions. Therefore, the proposed approach should include an evaluation component specifically designed to identify and mitigate such biases in the AI's behavior.

4. Realizing web applications for public processes using genAI and MDE

Our approach is integrating LLM-based agents and MDE methodologies to develop intelligent web-based systems specifically tailored for public administration (see Figure 2). The proposed system functions as an advanced navigator and form assistant, streamlining interactions between citizens and administrative bodies.

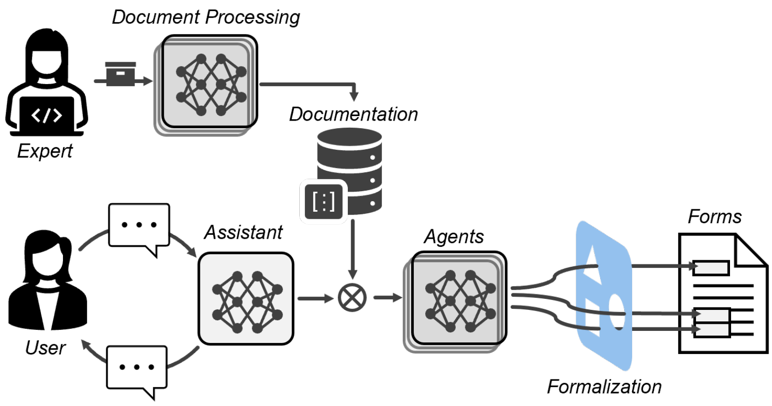


Figure 2: Expert feeds system with documentation, user can access documentation and formalization through AI-assistant.

An expert provides and maintains documentation about the processes within the specific administrative body. A collection of AI agents is used to process the different formats of the provided data. Once set up, a user can interact with an assistant who has access to that documentation. The assistant helps to guide the user through the correct processes and to the needed forms. In addition, the assistant can help users to fill out the given form or, under a user's supervision, fill out the forms.

4.1. Intelligent Explanation and Process Guidance

Complex administrative procedures frequently utilize domain-specific language, complicating interactions for users unfamiliar with such terminology. Our system addresses these challenges through intelligent explanation mechanisms:

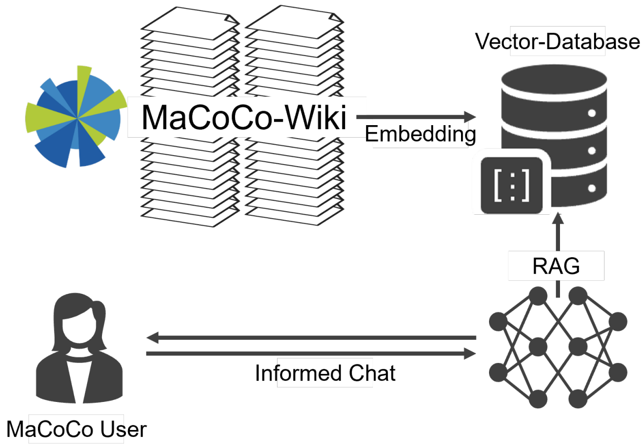
- **Domain-Specific Explanations:** The system leverages domain-specific knowledge, such as data from a Wiki, to automatically generate clear contextual explanations of procedural steps. We have developed this approach for a financial management system in the public domain [4].
- **Simplified Language Translation:** AI-driven Natural Language Processing can translate complex administrative jargon into simplified language, enhancing accessibility for different user groups [31].
- **Personalized Guidance:** LLM-based agents dynamically analyze user interactions to identify required information and offer targeted assistance for completing administrative processes.

Figure 3b presents an LLM-based AI assistant for users of an Information System [4]. A RAG based approach was used, to enrich an generic AI chat interface with platform specific knowledge. Additional guard railing keeps the conversational topic of the chatbot within help subjects and documentation of the targeted platform. Although online documentation offers a more reliable alternative than the chatbot, users seem to prefer the semantic search capabilities and natural language features that enable personalized explanations. Users preferred direct user support (e.g. via mail or telephone) over the chat bot in complex questions. This makes the assistant a good option to save resources of the support team by using it for common well documented questions and issues.

4.2. AI-Supported Form Completion Assistance

Administrative forms can be tedious and error-prone when filling them in. AI-enhanced mechanisms aim to simplify this by:

- **Automatic Form Filling:** Utilizing informally provided user information or integrated databases, the system autonomously pre-populates form fields.



(a) The user handbook (MaCoCo-Wiki) is initially embedded into an vector database and is from then on available for a RAG approach.

(b) Screenshot of the assistant feature in MaCoCo (v.2.16.0): Assistant aids in troubleshooting time sheets in the application.

Figure 3: Inclusion of an AI-assistant in order to aid users with complex processes within an application.

- **Field-specific Assistance and Clarifications:** Integrated AI agents can aid with context-sensitive explanations and audio-based assistance for individual form fields.
- **Verification and Validation:** User-submitted data undergoes preliminary verification, with final validation carried out by administrative authorities to ensure accuracy.

4.3. AI and MDE-Based Framework for System Maintenance

Public administrative processes evolve regularly due to regulatory or procedural changes. To speed up the development process and enable the authorities to react more quickly and independently to new requirements and legal changes, e.g., using the A12 platform [32, 33] has proven to be a successful MDE approach to generate public software - using it together with a domain-specific language to define tax law made it possible to put the form for tax relief due to COVID-19 (tax relief due to the effects of the coronavirus) online for certain federal states in Germany within a few days [18]. To manage the dynamism even better, our framework suggests the integration of AI-driven approaches and MDE principles.

- **Extensible Data Management:** The framework features a modular knowledge base that supports efficient integration, updating, and removal of administrative knowledge from diverse data formats (e.g., Wiki, Markdown, PDF, Video).
- **Multi-Agent Architecture:** Specialized agents within the framework handle tasks such as semantic processing, knowledge extraction, and data integration, ensuring robust management across multiple data formats and communication protocols.
- **Model Driven Engineering:** Leveraging MDE approaches, the system facilitates continuous deployment and rapid updates, maintaining alignment with evolving administrative requirements and regulatory standards.

Realizing these approaches requires a strong collaboration between domain experts from the public sector, software engineers, and involved user groups to realize well-accepted systems for the public sector. Figure 4 presents an approach on how to combine AI and MDE methods for this purpose. A developer or a domain-expert can informally define a set of requirements (1) in-context learning (IcL) [34] such as RAG or Few-Shot leaning can be used to enable the LLM to produce valid models [35] based on the user input. After a verification step, the model can be used as input for a generator (e.g., MontiGem [11]) that generates complete applications or parts of them. The generated application can

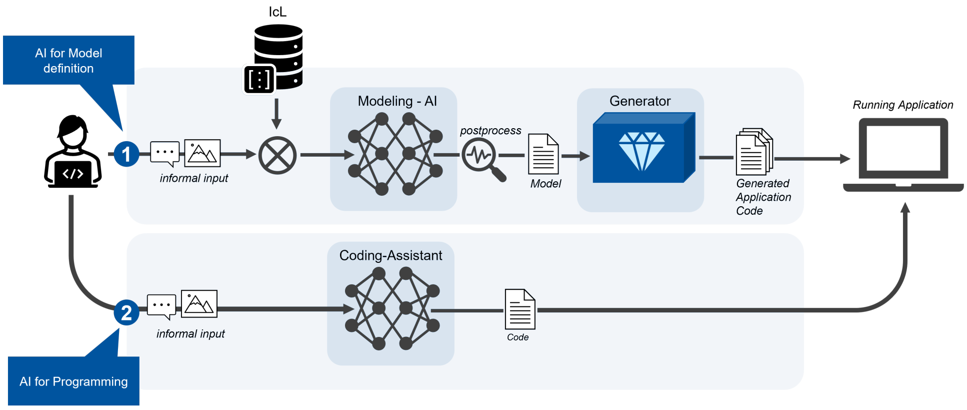


Figure 4: AI-based code generation: The developer can use AI to define models (1), that themselves are used to generate an application. In addition, genAI can be used to extend or adapt generated code (2).

either be modified directly of be modified with an coding-assistant (2). The capabilities of generative AI can be used to adapt the code that was produced by the generator further for customizing the software to the requirements of the domain expert. This method can be used to transform process documentation from the public sector into a running application [36].

5. Challenges of these approaches

The following challenges require an in-depth discussion and consideration when realizing web system generation and evolution with genAI and MDE methods.

5.1. Bias

The deployment of AI assistants, particularly those based on pre-trained Large Language Models, within public sector environments presents significant opportunities alongside notable challenges and risks. One primary concern pertains to inherent biases embedded within pre-trained models. Such biases, often reflective of the data used during the training phase, can result in problematic or skewed recommendations when interacting with diverse user groups. Specifically, user-bias amplification might lead to discriminatory practices or exacerbate existing inequalities by reinforcing biased decision-making.

5.2. Available Training Data

Another critical challenge involves the necessity for knowledge to be available predominantly in textual form. AI assistants rely heavily on structured and accessible textual resources, such as comprehensive process wikis, form-filling guidelines, or other explicit documentation. The absence or inadequacy of such resources limits the assistant's effectiveness, highlighting the need for robust and well-maintained textual repositories. Our approaches for exploring how to use LLMs for creating abstractions in models for uncommon modeling languages, so areas where not enough training data existed initially, have shown promising results [37]. These can be transferred to the public domain, where domain-specific modeling languages incorporating the relevant concepts of governance languages can be used as the basis for MDE toolchains, i.e., for describing tax law [18]. Moreover, AI-driven recommendations inherently carry the risk of inaccuracies or inappropriate suggestions, especially when encountering edge cases or ambiguous requests. Such errors can undermine trust, affect efficiency, or, in severe cases, lead to incorrect administrative decisions.

5.3. Alternative Non-AI Solutions

Determining appropriate scenarios for deploying AI-supported solutions as opposed to purely algorithmic or Model-Driven Engineering approaches is critical. AI assistance is advantageous in situations demanding flexibility, natural language processing, and adaptive decision-making capabilities. However, not every scenario requires AI; structured, predictable, and clearly rule-based administrative tasks often achieve better results through explicit, algorithm-based, or model-driven methodologies. Clearly distinguishing when each method is most suitable ensures transparency, consistency and reduces unnecessary complexity, risks, and resource waste when using computationally expensive AI methods.

5.4. Regulatory Constraints

Ensuring the inclusivity of AI-based support systems necessitates proactive measures. Inclusive design practices, comprehensive user testing across diverse demographics, and continuous feedback mechanisms are crucial strategies. Additionally, adhering to regulatory frameworks like the EU AI Act requires clarity about data management practices, especially beyond research contexts. Data utilized for public administration purposes must comply strictly with transparency, accountability, and fairness standards established by relevant regulations, ensuring ethical use and mitigating potential legal and social implications. To sum up, the integration of MDE and AI techniques could improve the engineering process of systems in the public sector, which would ultimately lead to citizens being guided through public processes in a more user-oriented way.

6. Conclusion

Public sector processes often pose significant barriers for citizens due to complex forms, specialized terminology, and a lack of user-friendly digital solutions. Motivated by this challenge, our research aimed to investigate how generative AI and model-driven engineering can be combined to create web-based systems that guide citizens through these processes in a more accessible and user-centric way.

To achieve this, we presented an integrated approach that leverages the natural language capabilities of Large Language Models for explanations, guidance, and form assistance, while utilizing MDE to ensure consistency, maintainability, and rapid adaptability of systems to evolving administrative requirements. Our methodology included designing modular architectures, developing AI-driven assistants, and illustrating these concepts through real-world examples such as the MaCoCo system and the A12 platform. However, the principles and techniques discussed are not limited to these specific frameworks; they are applicable to other model-driven development environments and modeling approaches as well, offering broad potential for adoption in diverse contexts.

Nonetheless, several limitations remain. Challenges such as mitigating biases in AI models, ensuring the availability and quality of domain-specific training data, identifying cases where AI solutions add genuine value compared to purely rule-based systems, and complying with regulatory frameworks like the EU AI Act are critical to address before widespread deployment. Moreover, our approach relies on high-quality domain knowledge repositories, which may not exist for all public-sector processes.

Despite these challenges, our results indicate significant potential for AI- and MDE-based methods to transform citizen interactions with public services. Future research should further explore robust validation mechanisms for AI-generated content, user studies to assess the practical usability of such systems across diverse populations, and methods for efficiently integrating evolving regulations into AI-assisted solutions. With growing momentum in digitalization efforts, we believe the integration of AI and MDE will play a key role in shaping the future of citizen-centered e-government services.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Writeflull in order to spell check and paraphrase. After using this tools, the authors reviewed and edited the content as needed.

References

[1] P. Dunleavy, H. Margetts, S. Bastow, F. Bouçek, R. Campbell, Difficult forms : how government agencies interact with citizens, 2003. Accessed: 20.6.2025.

[2] P. Esvary, Top 3 Reasons Why People Hate Filling Up Forms!, 2019. Accessed: 20.6.2025.

[3] K. Gasova, K. Stofkova, E-government as a quality improvement tool for citizens' services, Procedia Engineering 192 (2017) 225–230. 12th international scientific conference of young scientists on sustainable, modern and safe transport.

[4] A. Gerasimov, P. Letmathe, J. Michael, L. Netz, B. Rumpe, Modeling Financial, Project and Staff Management: A Case Report from the MaCoCo Project, Enterprise Modelling and Information Systems Architectures - International Journal of Conceptual Modeling 19 (2024). doi:10.18417/emisa.19.3.

[5] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large language models: A survey, 2025. URL: https://arxiv.org/abs/2402.06196. arXiv:2402.06196.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[7] S. Barke, M. B. James, N. Polikarpova, Grounded copilot: How programmers interact with code-generating models, Proceedings of the ACM on Programming Languages 7 (2023) 85–111.

[8] X. Hou, Y. Zhao, Y. Liu, Z. Yang, K. Wang, L. Li, X. Luo, D. Lo, J. Grundy, H. Wang, Large language models for software engineering: A systematic literature review, ACM Transactions on Software Engineering and Methodology 33 (2024) 1–79.

[9] T. Stahl, M. Völter, Model-driven software development: Technology, engineering, management, Wiley, Chichester, 2006.

[10] M. Brambilla, J. Cabot, M. Wimmer, Model-Driven Software Engineering in Practice, Springer International Publishing, Cham, 2017.

[11] C. Buschhaus, A. Gerasimov, J. C. Kirchhof, J. Michael, L. Netz, B. Rumpe, S. Stüber, Lessons Learned from Applying Model-Driven Engineering in 5 Domains: The Success Story of the MontiGem Generator Framework, Science of Computer Programming 232 (2024) 103033. doi:10.1016/j.scico.2023.103033.

[12] I. Drave, A. Gerasimov, J. Michael, L. Netz, B. Rumpe, S. Varga, A Methodology for Retrofitting Generative Aspects in Existing Applications, Journal of Object Technology (JOT) 20 (2021) 1–24. doi:10.5381/jot.2021.20.2.a7.

[13] A. Gerasimov, J. Michael, L. Netz, B. Rumpe, Agile Generator-Based GUI Modeling for Information Systems, in: A. Dahanayake, O. Pastor, B. Thalheim (Eds.), Modelling to Program (M2P), Springer, 2021, pp. 113–126.

[14] A. Gerasimov, J. Michael, L. Netz, B. Rumpe, S. Varga, Continuous Transition from Model-Driven Prototype to Full-Size Real-World Enterprise Information Systems, in: B. Anderson, J. Thatcher, R. Meservy (Eds.), 25th Americas Conference on Information Systems (AMCIS 2020), AIS, 2020.

[15] D. Bano, J. Michael, B. Rumpe, S. Varga, M. Weske, Process-Aware Digital Twin Cockpit Synthesis from Event Logs, Journal of Computer Languages (COLA) 70 (2022). doi:10.1016/j.cola.2022.101121.

[16] mgm technology partners GmbH, Widget showcase, 2021.

[17] J. Charles, J. Michael, L. Netz, B. Rumpe, Teaching Model-Driven Low-Code Development Platforms, in: Educators Symposium, MODELS Companion '24: International Conference on Model Driven Engineering Languages and Systems, Association for Computing Machinery (ACM), 2024, pp. 115–122. doi:10.1145/3652620.3687778.

[18] B. Rumpe, J. Michael, O. Kautz, R. Krebs, S. Gandenberger, J. Standt, U. Weber, Digitalisierung der Gesetzgebung zur Steigerung der digitalen Souveränität des Staates, volume 19 of *Berichte des NEGZ*, Nationales E-Government Kompetenzzentrum e. V., 2021.

[19] J. Michael, B. Rumpe, J. Standt, U. Weber, Domänenspezifische Modellierung für die Definition von Gesetzen und Ordnungen, 2022. URL: http://www.se-rwth.de/publications/Domaenenspezifische-Modellierung-fuer-die-Definition-von-Gesetzen-und-Ordnungen.pdf, gI

Radar, Gesellschaft für Informatik.
[20] V. Kulkarni, S. Reddy, S. Barat, J. Dutta, Toward a symbiotic approach leveraging generative ai for model driven engineering, in: 2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS), IEEE, 2023, pp. 184–193.
[21] S. Rädler, L. Berardinelli, K. Winter, A. Rahimi, S. Rinderle-Ma, Bridging mde and ai: a systematic review of domain-specific languages and model-driven practices in ai software systems engineering, Software and Systems Modeling (2024) 1–25.
[22] F. Benaben, M. Lauras, A. Fertier, N. Salatgé, Integrating model-driven engineering as the next challenge for artificial intelligence–application to risk and crisis management, in: 2019 Winter Simulation Conference (WSC), IEEE, 2019, pp. 1549–1563.
[23] B. S. Puranik, A. Sonawane, J. Jose, S. Chavan, Y. Patil, Enhancement of model driven software development using ai, in: 2024 4th Asian Conference on Innovation in Technology (ASIANCON), IEEE, 2024, pp. 1–5.
[24] N. Sherje, Enhancing software development efficiency through ai-powered code generation, Research Journal of Computer Systems and Engineering 5 (2024) 01–12.
[25] A. Androutsopoulou, N. Karacapilidis, E. Loukis, Y. Charalabidis, Transforming the communication between citizens and government through ai-guided chatbots, Government information quarterly 36 (2019) 358–367.
[26] M. Damar, A. Özen, E. Çakmak, E. Özoğuz, F. S. Erenay, Super ai, generative ai, narrow ai and chatbots: An assessment of artificial intelligence technologies for the public sector and public administration, Journal of AI 8 (2024) 83–106. doi:10.61969/jai.1512906.
[27] T. Avaner, M. Çelik, Türkiye'de dijital dönüşüm ofisi ve yapay zeka yönetimi: Büyük veri ve yapay zeka daire başkanlığı'nın geleceği üzerine, Medeniyet Araştırmaları Dergisi 6 (2021) 1–18.
[28] M. Arslan, S. Munawar, C. Cruz, Sustainable digitalization of business with multi-agent rag and llm, Procedia Computer Science 246 (2024) 4722–4731.
[29] F. Liu, Y. Liu, L. Shi, H. Huang, R. Wang, Z. Yang, L. Zhang, Z. Li, Y. Ma, Exploring and evaluating hallucinations in llm-powered code generation, arXiv preprint arXiv:2404.00971 (2024).
[30] Regulation (eu) 2024/1689 of the european parliament and of the council of 13 june 2024 laying down harmonised rules on artificial intelligence and amending regulations (ec) no 300/2008, (eu) no 167/2013, (eu) no 168/2013, (eu) 2018/858 and (eu) 2019/2144 (artificial intelligence act), https://eur-lex.europa.eu/eli/reg/2024/1689, 2024. Accessed: 16.06.2025.
[31] N. Freyer, H. Kempt, L. Klöser, Easy-read and large language models: on the ethical dimensions of LLM-based text simplification, Ethics and Information Technology 26 (2024).
[32] mgm technology partners, The A12 Enterprise Low Code Platform: Enterprise Low Code Digital Sovereignty for companies and public authorities, 2025.
[33] R. Krebs, S. Robles, J. Standt, Low code before it was cool, inf. Das Informatik Magazin (2024).
[34] N. Wies, Y. Levine, A. Shashua, The learnability of in-context learning, Advances in Neural Information Processing Systems 36 (2023) 36637–36651.
[35] L. Netz, J. Reimer, B. Rumpe, Using Grammar Masking to Ensure Syntactic Validity in LLM-based Modeling Tasks, in: Workshop on Artificial Intelligence and Model-driven Engineering, MODELS Companion '24: International Conference on Model Driven Engineering Languages and Systems (MDE Intelligence), Association for Computing Machinery (ACM), 2024, p. 570–577. doi:10.1145/3652620.3687805.
[36] L. Netz, J. Michael, B. Rumpe, From Natural Language to Web Applications: Using Large Language Models for Model-Driven Software Engineering, in: M. Weske, J. Michael (Eds.), Modellierung 2024, LNI, GI, 2024, pp. 179–195. doi:10.18420/modellierung2024_018.
[37] N. Baumann, J. S. Diaz, J. Michael, L. Netz, H. Nqiri, J. Reimer, B. Rumpe, Combining Retrieval-Augmented Generation and Few-Shot Learning for Model Synthesis of Uncommon DSLs, in: H. Giese, K. Rosenthal (Eds.), Modellierung 2024 - Workshopband, GI, 2024, pp. 1–15. doi:10.18420/modellierung2024-ws-007.