Component-based Approach to Software Engineering of Machine Learning-enabled Systems

Vladislav Indykov
indykov@chalmers.se
University of Gothenburg | Chalmers
Gothenburg, Sweden

ABSTRACT

Machine Learning (ML) - enabled systems capture new frontiers of industrial use. The development of such systems is becoming a priority course for many vendors due to the unique capabilities of Artificial Intelligence (AI) techniques. The current trend today is to integrate ML functionality into complex systems as architectural components. There are a lot of relevant challenges associated with this strategy in terms of the overall system architecture and in the context of development workflow (MLOps). The probabilistic nature, crucial dependency on data, and work in an environment of high uncertainty do not allow software engineers to apply traditional software development methodologies. As a result, there is a community request to systematize the most relevant experience in building software architectures with ML components, to create new approaches to organizing the process of developing ML-enabled systems, and to build new models for assessing the system quality. Our research contributes to all mentioned directions and aims to create a methodology for the efficient implementation of ML-enabled software and AI components. The results of the research can be used in the design and development in industrial settings, as well as a basis for further studies in the research field, which is of both practical and scientific value.

CCS CONCEPTS

• **Software and its engineering** → *Requirements analysis*; *Software design engineering*; *Design patterns*; **Software design tradeoffs**.

KEYWORDS

machine learning, software architecture, software quality

ACM Reference Format:
Vladislav Indykov. 2024. Component-based Approach to Software Engineering of Machine Learning-enabled Systems. In *Conference on AI Engineering Software Engineering for AI (CAIN 2024), April 14–15, 2024, Lisbon, Portugal.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3644815.3644976

1 INTRODUCTION

Machine learning (ML) engineering is an emerging field of research that lies at the intersection of *software engineering* and *artificial intelligence (AI) development*. Great interest in this area from the scientific community, practitioners, and industries has been observed in recent years due to the dynamic spread of AI techniques in various fields. This PhD project strives to obtain the most relevant experience in the field of ML engineering, which includes an analysis of grey and white literature.

The current trend today is to integrate ML functionality into complex systems as architectural components [9]. Following this trend, many practitioners face a number of challenges due to AI specifics. There are several still unresolved issues associated with both the unstable quality of architectural solutions and the relatively low efficiency of the development process [4].

To address architectural challenges, the main emphasis of the research is put on *Component-based Software Engineering (CBSE)* of *ML-enabled systems*. The *CBSE* promotes software development through construction from existing software components, the development of components as reusable entities, and system evolution realization by the customization and replacement of components [18]. The operational side of software engineering of ML-enabled systems, known as *Machine Learning Operations (MLOPs)*, is also of particular interest for this research. *MLOps* is a set of principles and practices adopted from Development Operations (DevOps) and applied to the development of machine learning systems [1].

Previous studies were conducted to identify current issues in CBSE of AI-enabled software. There were a systematic literature review [12], a multi-vocal literature review from leading AI companies [11] and case studies in Swedish companies [10]. These works identified a wide range of challenges that can be classified into several categories from technical to organizational: problems in mixing data and code in version management and dependency management, prediction non-linearity, problems in the inability to ensure quality assurance, unreliable project planning, etc. Based on the results, it was concluded that several challenges have been experienced, but the overall solution for seamless continuous development, integration, deployment, operations, and evolution is still missing. The goal of this PhD project is to address this gap.

The main research hypothesis: "The approaches from component-based software engineering (both technical and operational) can be tailored for solving the architectural challenges created by AI specifics".

1.1 Related work

Nowadays, we are witnessing the dynamic growth of scientific activity in the field of ML-enabled software engineering. Some works describe how the engineering of AI-based systems is implemented in separate companies. Amershi et. al [2] shared the experience of Microsoft and highlighted the need for detailed studies of CBSE for ML-enabled systems, Google software engineers [15] presented their observations and emphasized the crucial role of hidden technical debt consideration in MLOps, SAP and Polytechnique Montreal [14] reported some recommendations for building ML applications and highlighted "a growing need for a consolidated set of guidelines regarding software engineering for machine learning".

Some authors conducted systematic literature reviews and concluded that "mature tools and techniques are missing to engineer ML systems" [5], the application of software design patterns to AI-based systems needs deeper investigation [22] and the existing fundamental differences between the development of traditional and ML-enabled software requires systematic study of corresponding software architectures [20]. An analysis of related work in the research area has shown the emergence of an increasing number of high quality papers, that identify practical and theoretical challenges in constructing AI-based systems and describe special cases of their solution [6, 7, 13, 16, 21]. However, most studies continue to highlight the need for in-depth research continuation and systematic exploration of the problems and possible decisions associated with the design of ML-enabled system architectures. There is an in-demand request for system-level common recommendations that would be relevant for the majority of ML-enabled solutions or divided by certain types of ML solutions (e.g. Deep Learning (DL)-based, Reinforcement Learning (RL) - based etc.).

Therefore, **the research problem** is formulated as "the lack of systematic work that brings together relevant scientific and practical experience in building architectures for ML-enabled systems and organizing the process of ML-enabled software development".

2 CONTRIBUTIONS

The main result of the research will be a new multifaceted approach to the design and development of ML-enabled systems. The project examines ML-enabled software engineering from different angles, however, all research contributions are united by one goal - increasing the quality of the development. Moreover, each subsequent contribution uses the results of the previous one as an input to obtain a new result. The approach will include the identification of common indicators for quality assurance (qualitative side); the design of a quality-driven reference architecture to achieve quality attributes identified above built on the most relevant practical and scientific experience and tested in industrial settings (architectural side); and the development of guidance for ML-enabled software engineers built on the best practices from experts for effective implementation of reference architectures (operational side). The description of contributions can be found below:

Common Quality Model of ML-enabled systems. A systematic white literature review in the field of ML-enabled software architectures identified a critical need to create a model that describes the quality of such systems and considers their specificity. More than 60 scientific papers were analyzed to exclusively identify quality attributes of ML-enabled systems. It is planned to test the relevance and adequacy of this model in industrial conditions through an expert interview and assessment. An initial evaluation of this model was conducted during its demonstration at the Swedish Requirements Engineering Meeting (SiREN 2023) and got positive appraisal from practitioners. The resulting model was descriptive according to DAP classification [19]. It was included in a scientific paper entitled "Architectural Tactics to Achieve Common Quality Attributes of Machine-Learning-Enabled Systems" that was submitted to an International Conference in December 2023 [8].

Quality-driven Software Reference Architecture. To achieve the identified quality attributes, various architectural tactics (ATs) were studied. In the absolute majority of cases, quality trade-offs arose (for example, architectural tactics can improve the model accuracy, but worsen resource efficiency). The study of ATs and trade-offs was also included in the systematic literature review mentioned earlier [8]. In the course of further research, it is necessary to explore design patterns and component models of existing AI-based solutions, as well as their relationship with quality attributes. The results of the study will be combined into reference architecture of ML-enabled systems, which can serve as the primary template for software engineers to architecturally achieve certain qualities. In the context of our project, the reference architecture corresponds to "Type 1" (Low-level) according to classification by Angelov et. al [3]. It remains to be explored whether it can be domain- and model-type-independent. If it is not, the development of several reference architectures is possible. The resulting reference architecture(s) will be evaluated by interviews with experts and relevant case studies.

Modelling Guidance for Software Architects. A significant part of the studied industrial experience can describe not only the architectures but also the process of their creation. It includes several organizational decisions, guidelines, and tooling. The collected experience can be compiled into guidance for software architects in the field of ML-enabled systems. This guidance is planned to be presented as a collection of best practices, which will give the architects flexibility in organizing the process. They will be able to use separate best practices as components of their own project without forcing themselves into a strict process model. In addition, various possible implementations of automated tools for supporting software architects based on the guidance are considered: modeling assistants, automated techniques for enforcing the correct implementation of reference architecture, and validation of conformance of the implemented system's behavior to architectural decisions based on run-time data. The results will be evaluated by a focus group of practitioners.

3 CONCLUSION

The results of this research can be considered as an extension to the Software Engineering Body of Knowledge (SWEBoK) [17] that explores the specifics of machine learning development in the context of overall system design. This PhD project contributes to the following SWEBoK's Knowledge Areas: Software Design, Software Engineering Management and Software Engineering Process, Software Quality, and Software Engineering Professional Practice. Other knowledge areas are left out of scope due to the time frames, however, they have great potential for further studies.

Acknowledgement. The author wishes to thank the reviewers for their constructive and high-quality feedback and his scientific supervisors Daniel Strüber and Rebekka Wohlrab for their constant support and guidance in running this PhD project. The author also expresses special thanks to the entire team of the CSE Department.

REFERENCES

[1] Sridhar Alla, Suman Kalyan Adari, Sridhar Alla, and Suman Kalyan Adari. 2021. What is MLOps? *Beginning MLOps with MLFlow: Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure* (2021), 79–124.

[2] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. IEEE, 291–300.

[3] Samuil Angelov, Paul Grefen, and Danny Greefhorst. 2009. A classification of software reference architectures: Analyzing their success and effectiveness. In *2009 Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture*. IEEE, 141–150.

[4] Josu Diaz-de Arcaya, Ana I Torre-Bastida, Gorka Zárate, Raúl Miñón, and Aitor Almeida. 2023. A joint study of the challenges, opportunities, and roadmap of mlops and aiops: A systematic survey. *Comput. Surveys* 56, 4 (2023), 1–30.

[5] Görkem Giray. 2021. A software engineering perspective on engineering machine learning systems: State of the art and challenges. *Journal of Systems and Software* 180 (2021), 111031.

[6] Lukas Heiland, Marius Hauser, and Justus Bogner. 2023. Design Patterns for AI-based Systems: A Multivocal Literature Review and Pattern Repository. *arXiv preprint arXiv:2303.13173* (2023).

[7] Hans-Martin Heyn, Eric Knauss, and Patrizio Pelliccione. 2023. A compositional approach to creating architecture frameworks with an application to distributed AI systems. *Journal of Systems and Software* 198 (2023), 111604.

[8] Vladislav Indykov, Daniel Strüber, and Rebekka Wohlrab. 2023. Architectural Tactics to Achieve Common Quality Attributes of Machine-Learning-Enabled Systems. In *in submission*. 1–12.

[9] C Kästner. 2022. Machine learning in production: From models to systems.

[10] Lucy Ellen Lwakatare, Ivica Crnkovic, Ellinor Rånge, and Jan Bosch. 2020. From a data science-driven process to a continuous delivery process for machine learning systems. In *Product-Focused Software Process Improvement: 21st International Conference, PROFES 2020, Turin, Italy, November 25–27, 2020, Proceedings 21*. Springer, 185–201.

[11] Lucy Ellen Lwakatare, Aiswarya Raj, Jan Bosch, Helena Holmström Olsson, and Ivica Crnkovic. 2019. A taxonomy of software engineering challenges for machine learning systems: An empirical investigation. In *Agile Processes in Software Engineering and Extreme Programming: 20th International Conference, XP 2019, Montréal, QC, Canada, May 21–25, 2019, Proceedings 20*. Springer International Publishing, 227–243.

[12] Lucy Ellen Lwakatare, Aiswarya Raj, Ivica Crnkovic, Jan Bosch, and Helena Holmström Olsson. 2020. Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. *Information and software technology* 127 (2020), 106368.

[13] Roger Nazir, Alessio Bucaioni, and Patrizio Pelliccione. 2024. Architecting ML-enabled systems: Challenges, best practices, and design decisions. *Journal of Systems and Software* 207 (2024), 111860.

[14] Md Saidur Rahman, Emilio Rivera, Foutse Khomh, Yann-Gaël Guéhéneuc, and Bernd Lehnert. 2019. Machine learning software engineering in practice: An industrial case study. *arXiv preprint arXiv:1906.07154* (2019).

[15] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. 2015. Hidden technical debt in machine learning systems. *Advances in neural information processing systems* 28 (2015).

[16] Alex Serban, Koen van der Blom, Holger Hoos, and Joost Visser. 2024. Software Engineering Practices for Machine Learning—Adoption, effects, and team assessment. *Journal of Systems and Software* 209 (2024), 111907.

[17] Computer Society. 2023. *SWEBOK Guide Version 4.0 beta*. IEEE.

[18] Clemens Szyperski, Dominik Gruntz, and Stephan Murer. 2002. *Component software: beyond object-oriented programming*. Pearson Education.

[19] Stefan Wagner. 2013. Software product quality control. (2013).

[20] Zhiyuan Wan, Xin Xia, David Lo, and Gail C Murphy. 2019. How does machine learning change software development practices? *IEEE Transactions on Software Engineering* 47, 9 (2019), 1857–1871.

[21] Stephen John Warnett and Uwe Zdun. 2022. Architectural design decisions for machine learning deployment. In *2022 IEEE 19th International Conference on Software Architecture (ICSA)*. IEEE, 90–100.

[22] Hironori Washizaki, Hiromu Uchida, Foutse Khomh, and Yann-Gaël Guéhéneuc. 2019. Studying Software Engineering Patterns for Designing Machine Learning Systems. In *2019 10th International Workshop on Empirical Software Engineering in Practice (IWESEP)*. 49–495. https://doi.org/10.1109/IWESEP49350.2019.00017