

Chapter 19

Requirements Modeling: A Use Case Approach to Machine Learning

Murat Pasa Uysal

 <https://orcid.org/0000-0002-8349-9403>

Baskent University, Turkey

ABSTRACT

Industrial applications and research studies report that organizations using machine learning (ML) solutions may be at risk of failure or they could fall short of business objectives. One serious issue that is often neglected is meeting the specific requirements of machine learning-driven software systems (MLD-SS). The use of a variety of information technologies, integration methods and tools, and domain-specific processes can add to this complexity. The data-driven and black-box nature of ML may be another great challenge. Therefore, there is a clear need for adopting a stakeholder-centered approach to the requirements engineering (RE) of MLD-SS. Use case modeling (UCM) can make requirements simpler and understandable by all stakeholders, allow better communication of ideas and provide support to testing, validation, and verification processes. In this chapter, a ten-step RE method and a four-step UCM method are proposed for MLD-SS, and then these methods are applied in a real case study of a university hospital.

INTRODUCTION

Machine learning-driven software systems (MLD-SS) may be considered more complex than software systems, such as web applications, enterprise systems, etc. This situation can get worse when considering the vital importance of mission-critical systems in various domains, such as healthcare, finance, and smart production. The use of a variety of information technologies, integration methods, and tools, domain-specific processes can add to the complexity of MLD-SS. On the other hand, it is not unusual to see a considerable number of failure stories about Machine Learning (ML) projects. Industrial applications and research studies report that organizations using ML solutions may be at risk of failure or they can fall short of business and organizational objectives (Uysal, 2022a).

DOI: 10.4018/978-1-6684-9809-5.ch019

One serious issue that is often neglected is to meet the specific requirements of MLD-SS, such as software engineering (SE) requirements, ML-specific requirements, and domain-specific requirements. SE extends its methods and activities beyond data processing and model development, and it focuses on the software development processes. ML, on the other hand, emphasizes the data processing, model development and deployment. Therefore, aligning, and synchronizing data processing and model development with the SE processes is a serious issue (Amershi et al. 2019). Requirements processes, team management, quality management, risk management, and maintenance are also the core aspects both for SE and ML. However, requirements specification for ML is much more challenging (Rahman et al., 2019). Some studies indicate the dynamic and complex context of the design, development, and maintenance of large-scale MLD-SS. ML contexts differ from the plan-driven or agile software development contexts. Therefore, ML and SE workflow management needs to be integrated with the SE best practices (Lwakatare et al., 2020).

A Microsoft team conducted a research study on the SE challenges for ML and AI projects (Amershi et al. 2019). It is found that data discovery and management is complicated and more difficult than the other SE activities. Model customization and reuse require different skills than that are found in SE teams. Since SE and ML modules can be entangled in various ways, ML processes may also be more challenging than the software processes. Therefore, they suggest that the integration of software development infrastructure with ML development would speed up the ML experimentations. Moreover, automating the ML processes, such as data aggregation, synthesizing, feature extraction, and pipeline are the other major concerns for the teams.

Some studies present the SE challenges as they are related with the issues of requirements engineering (RE) of ML (Uysal, 2022b). The systematic mapping study of Villamizar et al. (2021) reveals that the handling customer expectations and lack of validated RE techniques as the main challenges. Giray's (2021) extensive systematic literature review (SLR) gives the state of art and SE challenges for ML. He adopts a SE perspective on engineering ML systems. Accordingly, forming coherent teams, requirements process, tailoring, and assessing the SE process, managing, and harmonizing the ML system development with software development are the main issues. Fernández et al (2021) conducts a SLR on SE for AI-based systems and finds safety and dependability as the most studied research topics. Software quality and software testing are dominant whereas the areas such as, software maintenance are neglected. They also identify various SE approaches to AI-based systems that are classified according to the Software Engineering Body of Knowledge (SWEBOK) (2014) knowledge areas (KAs). The data-driven and black box nature of ML may be another great challenge.

Kumeno (2019) conducted the first review to map the SE challenges for ML to the SE KAs defined by the SWEBOK standard. The topics in KAs, such as safety, security, validation, and verification are identified as the major challenge topics over the ones in other KAs. Lwakatare et al (2020) focuses on large-scale ML systems in real-world industrial settings. Adaptability and scalability are found as the most often reported SE challenges while safety and privacy are the least reported quality attributes. Nascimento et al (2020) highlights these five of thirteen categories: test, software quality, data, management, model development, and project management (PM). Design, construction, testing, and configuration management KAs have the highest number of solution proposals.

Behaviors of software systems are initially specified and defined by using design patterns and programming models. However, this is reverse in ML systems where system behaviors are continuously learned after processing the training data sets. Even minor changes in the input can drastically change the ML application behaviors, which means that ML projects need specific techniques for testing, validation,

13 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/requirements-modeling/330501

Related Content

Design of Semi-Structured Database System: Conceptual Model to Logical Representation

Anirban Sarkar (2013). *Designing, Engineering, and Analyzing Reliable and Efficient Software* (pp. 74-95).

www.irma-international.org/chapter/design-semi-structured-database-system/74875

PRISM: Visualizing Personalized Real-Time Incident on Security Map

Takuhiro Kagawa, Sachio Saiki and Masahide Nakamura (2018). *International Journal of Software Innovation* (pp. 46-58).

www.irma-international.org/article/prism/210454

Service-Oriented Computing and the Model-Driven Architecture

Giacomo Piccinelli and James Skene (2005). *Service-Oriented Software System Engineering: Challenges and Practices* (pp. 109-131).

www.irma-international.org/chapter/service-oriented-computing-model-driven/28952

Analysis of Psychological Distress During COVID-19 Among Professionals

Supriya Raheja (2022). *International Journal of Software Innovation* (pp. 1-17).

www.irma-international.org/article/analysis-of-psychological-distress-during-covid-19-among-professionals/309109

Using Attack Graphs to Analyze Social Engineering Threats

Kristian Beckers, Leanid Krautsevich and Artsiom Yautsiukhin (2015). *International Journal of Secure Software Engineering* (pp. 47-69).

www.irma-international.org/article/using-attack-graphs-to-analyze-social-engineering-threats/136466