Using i* to Analyze Collaboration Challenges in MLOps Project Teams

Rohith Sothilingam[1,2], Vik Pant[1] and Eric Yu[1,2]

[1]Faculty of Information, University of Toronto, Toronto, Canada
[2]Department of Computer Science, University of Toronto, Canada

Abstract
The rapidly growing interest of applying continuous software engineering practices from DevOps in Machine Learning (ML) software projects has led to the relatively new area of Machine Learning Operations (MLOps). The need for software engineers to collaborate with data scientists, ML Engineers, DevOps Engineers, and other specialists has contributed to the emergence of MLOps. MLOps introduces unique challenges involving the intersection of infrastructure engineering to the exploratory model development process of ML. The collaboration of team members of diverse skills and knowledge is required due to the need for continuous evolution and monitoring of both ML systems and the underlying infrastructure. The capabilities of i* modeling are potentially a good fit for the the unique characteristics of MLOps with respect to the analysis of nuances in strategic interests between actors of diverse disciplinary backgrounds that commonly face challenges during collaboration. In this work, we use i* Strategic Rationale modeling of actor relationships to analyze and resolve common conflicts and challenges faced in MLOps project teams during collaboration in the development of production ML systems. Examples from common key MLOps challenges are used to illustrate.

Keywords
Conceptual Modeling, Requirements Engineering, MLOps

1. Introduction

Continuous software engineering practices such as DevOps has contributed greatly to business value in software development teams. The recent growth in interest toward applying the practices of DevOps towards Machine Learning (ML) has led to the emergence of MLOps. MLOps is an intersection between ML and DevOps practices. Tamburri [1] defines MLOps as the distribution of a set of software components and middleware encompassing five ML pipeline functions: data ingestion, data transformation, continuous ML model (re-)training, (re-)deployment, and output presentation.

MLOps projects and team organization are often at the early stages of the maturity curve, compared to other types of software projects [2]. The software lifecycle for MLOps is similar to traditional practice of DevOps within other software engineering areas, which involve collaboration among people from diverse backgrounds and skills [3]. While DevOps aims

iStar'22: The 15th International i Workshop, October 17th, 2022, Hyderabad, India*

rohith.sothilingam@mail.utoronto.ca (R. Sothilingam); vik.pant@utoronto.ca (V. Pant); eric.yu@utoronto.ca (E. Yu)

to shorten the development lifecycle of software systems, MLOps aims to borrow DevOps principles to automate and operationalize ML applications and workflows.

Unlike more mature areas of software development, MLOps practices are still evolving and ill-defined, leading to challenges upon collaboration between roles of diverse backgrounds and technical skills. Conflicts are particularly apparent during collaboration within ML teams due to the interdisciplinary nature of MLOps and lack of MLOps practices in place, which can often pose challenges. Other studies have observed that differing expectations, perspectives, and interests in a software system can lead to conflicts during collaboration due to methodological differences [4]. To achieve successful interdisciplinary collaboration in MLOps, it is important to account for and analyze professional roles, strategic interests, skills, and attributes of team members involved.

In this paper we use i* modeling to examine how MLops practices are able to overcome key challenges of ML before the adoption of MLOps practices. We use i* Strategic Rationale modeling to demonstrate why and how MLOps practices are able to overcome those challenges. This paper contributes to i* with respect to identifying nuanced challenges of MLOps that are rooted in collaboration. The main contribution of this paper is its demonstration of the ability of i* to convey the effect of collaboration patterns because of enabling MLOps practices. The analysis demonstrated in this paper contributes to our ongoing work of examining whether i* Strategic Rationale models can sufficiently address intricate issues of MLOps collaboration.

2. Using i* Strategic Actor Modeling to Identify Challenges in MLOps Project Team Collaboration

Recent empirical studies have found that while some organizations have adopted MLOps practices more successfully than others [3], many struggle setting up structures, processes, and tooling for effective collaboration among team members with different backgrounds when developing ML-enabled systems.

Much of software code in ML systems is dedicated to pipelines which contribute to continuous learning, such as the training pipeline and data pipeline [5]. The processes used to develop such pipelines are often subject to technical debt. In this section, we use i* Strategic Rationale modeling to demonstrate an example of a design pattern as identified in Sculley et al. [5] which can lead to technical debt [5]. Specifically, we demonstrate how i* can be used to identify the underlying goals and soft-goals which lead to the issue of *Process Management Debt*.

In more mature ML project teams, numerous ML models may be running simultaneously [6] which raises the important challenge of managing changes and versioning of models safely and automatically. It is challenging to respect different, often conflicting business priorities among models and to subsequently detect any blockages or issues in the related pipelines.

To understand such points of conflict, we pose the following questions. What tasks and goals must be achieved for MLOps project teams to maintain continuous model performance? What tasks and goals must be achieved to continually monitor and train models simultaneously across different releases? In the following i* modeling, we will identify what modeling elements are not satisfied and the significance of those elements not being satisfied. Our intention is to demonstrate collaboration challenges known before the construction of the i* models. The target

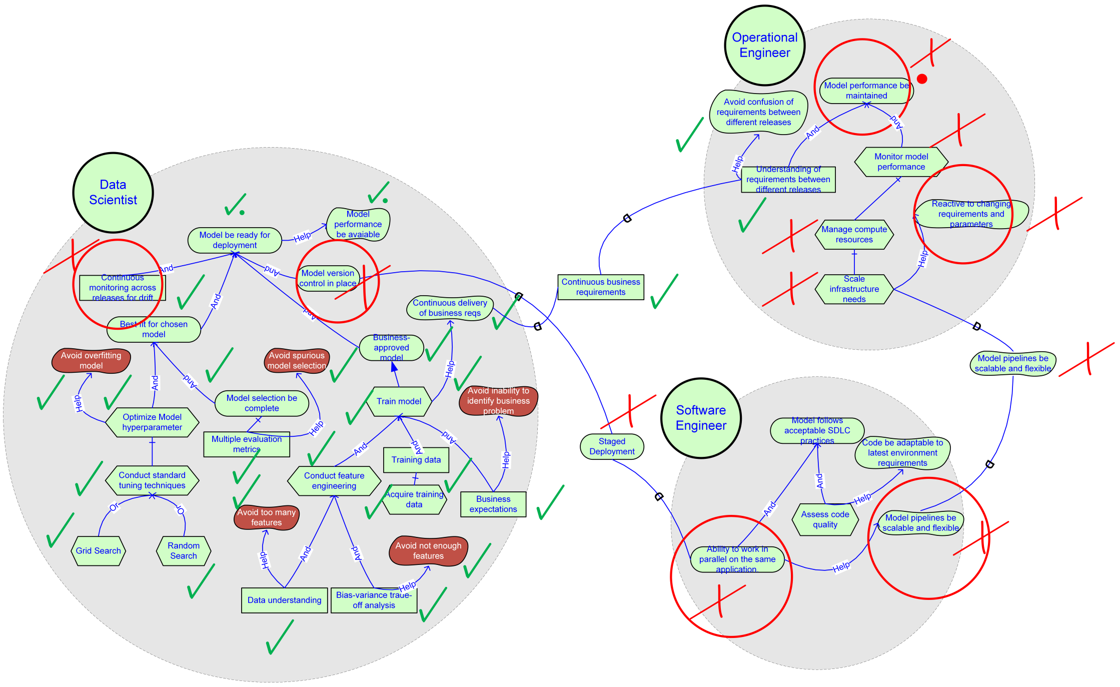
Data Scientist
Model be ready for deployment
Help
Model performance be avaiable
And
Continuous monitoring across releases for drift
And
And
Model version control in place
Best fit for chosen model
And
Business-approved model
Continuous delivery of business reqs
Avoid overfitting model
Avoid spurious model selection
Help
And
And
Model selection be complete
Train model
Help
Avoid inability to identify business problem
Optimize Model hyperparameter
Multiple evaluation metrics
Help
And
And
Training data
And
Conduct standard tuning techniques
Conduct feature engineering
Acquire training data
Help
Business expectations
Or
Or
Avoid too many features
Grid Search
Random Search
Help
And
And
Avoid not enough features
Help
Data understanding
Bias-variance trade-off analysis
Continuous business requirements
Operational Engineer
Avoid confusion of requirements between different releases
Model performance be maintained
And
And
Help
Monitor model performance
Understanding of requirements between different releases
Reactive to changing requirements and parameters
Manage compute resources
Help
Scale infrastructure needs
Model pipelines be scalable and flexible
Staged Deployment
Software Engineer
Model follows acceptable SDLC practices
Code be adaptable to latest environment requirements
And
And
Help
Assess code quality
Model pipelines be scalable and flexible
Ability to work in parallel on the same application
Help

Figure 1: i* Strategic Rationale Model showing collaboration challenges as a result of Process Management Debt (as mentioned in [5])

audience of the i* models are ML model designers in practical settings who would consume and analyze such i* models. The i* model in Fig. 1 is an example of one of many common recurrent challenges faced with MLOps regarding team collaboration. Though the model deals with a specific problem domain, the model can be customized to specific MLOps project challenges.

This model in Fig. 1 conveys the attempted collaboration between three important actors of any MLOps project team [7]: the *Data Scientist*, *Operational Engineer*, and the *Software Engineer*. The *Data Scientist* is typically responsible for the design of the model, including feature engineering, model training, and model fit with accordance to the business requirements. The *Operational Engineer* is responsible for constantly monitoring the performance of the model, with dips in performance possibly indicating that this entire process may need to be repeated to update the model to understand new trends. The *Software Engineer* is responsible for deploying and integrating the models into the application through model pipelines. In the following paragraphs, we will show the i* modeling elements in Fig. 1 are not satisfied as a result of a lack of established collaborative practices between each actor. The i* modeling elements which are not satisfied are highlighted using red circles in Fig. 1.

The *Operational Engineer* Actor cannot achieve the Goal of *Model performance be maintained* as the task dependum *Scale infrastructure needs* between the *Operational Engineer* and *Software Engineer* Actors is not satisfied. Scalability is a crucial contribution of the DevOps aspect of MLOps. MLOps adopts the DevOps principles of Continuous Integration (CI) and Continuous Delivery (CD) [8]. From the DevOps perspective of CD, processed datasets and trained models

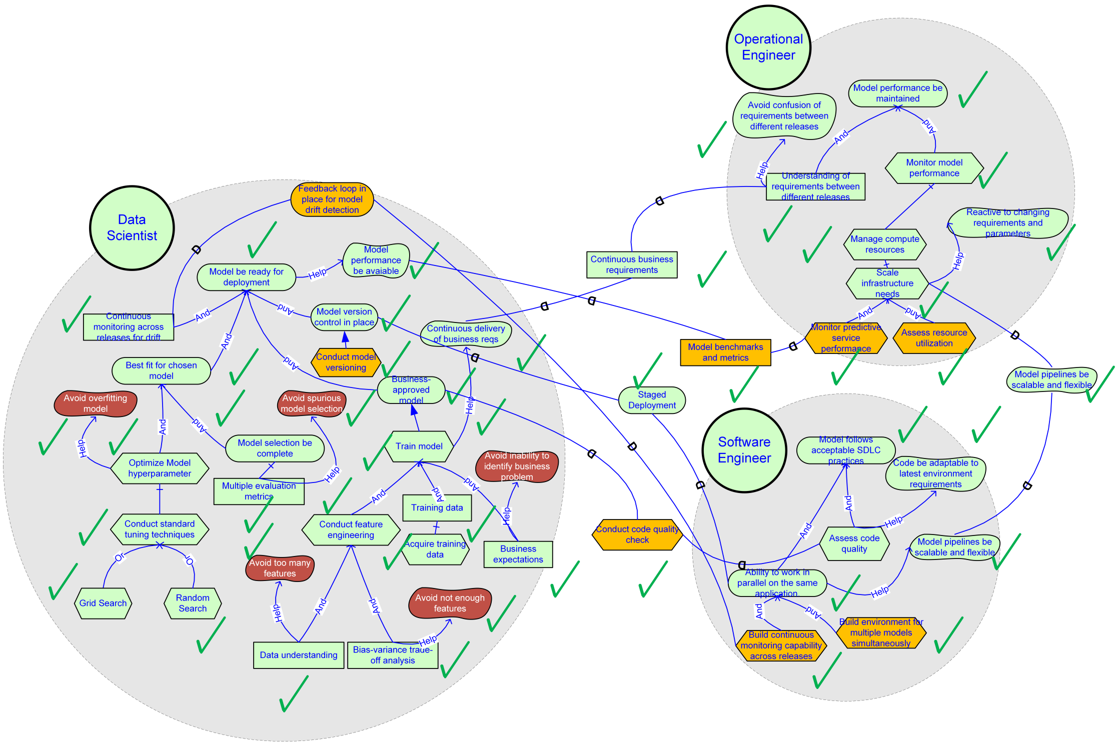
Data Scientist
Feedback loop in place for model drift detection
Model be ready for deployment
Help
Model performance be avaiable
Continuous monitoring across releases for drift
And
And
And
Model version control in place
Conduct model versioning
Continuous delivery of business reqs
Best fit for chosen model
And
Avoid overfitting model
Help
Optimize Model hyperparameter
And
Avoid spurious model selection
Business-approved model
Help
Model selection be complete
Multiple evaluation metrics
Help
Train model
Avoid inability to identify business problem
And
And
Training data
Acquire training data
Help
Business expectations
Conduct standard tuning techniques
Or
Or
Grid Search
Random Search
Conduct feature engineering
Avoid too many features
Help
And
And
Avoid not enough features
Help
Data understanding
Bias-variance traue-off analysis
Operational Engineer
Avoid confusion of requirements between different releases
Model performance be maintained
And
And
Help
Understanding of requirements between different releases
Monitor model performance
Continuous business requirements
Reactive to changing requirements and parameters
Manage compute resources
Scale infrastructure needs
Help
And
And
Model benchmarks and metrics
Monitor predictive service performance
Assess resource utilization
Model pipelines be scalable and flexible
Staged Deployment
Software Engineer
Model follows acceptable SDLC practices
Code be adaptable to latest environment requirements
And
And
Help
Assess code quality
Conduct code quality check
Model pipelines be scalable and flexible
Ability to work in parallel on the same application
Help
And
And
Build continuous monitoring capability across releases
Build environment for multiple models simultaneously
D

Figure 2: i* Strategic Rationale Model showing the effects of MLOps practices to overcome collaboration challenges faced by MLOps project teams

are automatically and continuously delivered by data scientists to Operational Engineers. From the perspective of Continuous Training, introduction of new data and avoidance of model performance degradation require a trigger to retrain the model or improve model performance through online methods. Without scalable infrastructure, the Operational Engineer cannot maintain continuous model performance.

As a result of the lack of continuous processes in place, the *Data Scientist* Actor cannot achieve the Goal of *Model be ready for deployment* because the underlying Goal *Model version control in place* and Resource *Continuous monitoring across releases for drift* are not satisfied. The *Data Scientist* Actor cannot achieve the Goal *Model version control in place*, leading to the inability of the *Software Engineer* Actor to satisfy the Goal *Ability to work in parallel on the same application*, to train and deploy multiple models continuously.

3. Overcoming Challenges in ML Project Team Collaboration and Relationships with MLOps Practices

The i* Strategic Rationale model in Fig. 2 conveys how the adoption of appropriate MLOps practices can help overcome such challenges. In this section, we demonstrate an i* Strategic Rationale model with MLOps practices added (highlighted as yellow elements) to Fig. 1. Adding the i* intentional elements conveying MLOps practices and its effects on goals, soft-goals, and dependum relationships fosters the ability to understand why and how MLOps practices can improve collaboration challenges among relationships in MLOps project teams. Modeling elements conveying MLOps practices are drawn from recent research studies on MLOps practices [7] [9][10].

To address Process Management Debt, it is crucial to develop tooling to aid recovery from production incidents and manage conflicting priorities [6]. In Fig. 2, we added dependency relationships with MLOps practices to address the collaboration relationships which were unsatisfied in Fig. 1.

The Task dependum *Conduct code quality check* allows the *Software Engineer* Actor to now satisfy the task *Assess code quality* while collaborating with the *Data Scientist* Actor to ensure the *Business-approved model* meets the expectations of the code quality check.

The Goal dependum *Feedback loop in place for model drift detection* [10] allows the *Data Scientist* Actor to satisfy the Resource element *Continuous monitoring across releases for drift*. This is done with improved collaboration between the *Data Scientist* Actor and *Software Engineer* Actor with the newly added task *Build continuous monitoring capability across releases* [9] which ensures CD principles are applied to continuous model training and monitoring.

The Resource dependum *Model benchmarks and metrics* allows the *Operational Engineer* to satisfy the task *Scale infrastructure needs* as the MLOps practices of *Assess resource utilization* [7] and *Monitor predictive service performance* are added as task elements to provide the *Operational Engineer* with the resource and practices required to ensure the compute resources of the ML model can be optimized and ultimately satisfy the goal *Model performance be maintained*.

The problems identified in Fig.1 are resolved using i* with the addition of these i* elements. The collaborative relationships between the *Data Scientist*, *Software Engineer*, and *Operational Engineer* Actors are improved as a result. Though i* can address some aspects of MLOps challenges, there are important limitations. Firstly, though i* is able to implicitly express the existence of a feedback loop, it cannot explicitly express the feedback loop and its continuity temporally. This is an important limitation of i* as each stage of the MLOps lifecycle is continuous and cyclical. Key examples include continuous training, deployment, monitoring, and simultaneous training of multiple models. Secondly, it is not clear the relationship between the skills, knowledge, and training of the Data Scientist and their ability to achieve a satisfactory level of code quality. Further work is required to better analyze how such knowledge can evolve and be incorporated at different points in collaborative relationships regarding MLOps projects.

4. Ongoing Work

In this work, we used i* Strategic Rationale Modeling to demonstrate the effects of MLOps practices on overcoming collaboration challenges in MLOps project teams. Modeling the relationships at an intentional level, compared to conventional process models, offers a higher level of abstraction for analysis. In ongoing work, we are applying the i* concepts of Agents, Roles, and Positions for modeling complex organizational relationships with respect to MLOps project teams [11] [12]. As mentioned in Section 3, one important example of a challenge that requires deeper analysis is the relationship between skills of a person and the expectations or goals of the corresponding job role they are expected to occupy. Challenges can occur due to an Agent not having the skill required to a Role, or having Roles with conflicting goals under one Position. By identifying where such issues occur, we will use i* Agents, Roles, and Positions to explore how organizations can diagnose challenges in team design in greater detail, through the early detection of the problem. This ongoing work is a part of larger PhD thesis objectives, which will include the following: (1) a requirements-driven framework which deals with conflicting goals at design decision points throughout MLOps; (2) compilation and codification of design knowledge from pertinent literature on MLOps and responsible AI to be available during design decision in the form of knowledge catalogs; (3) tool support for the proposed framework.

References

[1] D. A. Tamburri, Sustainable mlops: Trends and challenges, in: 2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), IEEE, 2020, pp. 17–23.

[2] M. M. John, H. H. Olsson, J. Bosch, Towards mlops: A framework and maturity model, in: 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2021, pp. 1–8.

[3] N. Nahar, S. Zhou, G. Lewis, C. Kästner, Collaboration challenges in building ml-enabled systems: Communication, documentation, engineering, and process, Organization 1 (2022) 3.

[4] S. Passi, P. Sengers, Making data science systems work, Big Data & Society 7 (2020) 2053951720939605.

[5] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, D. Dennison, Hidden technical debt in machine learning systems, Advances in neural information processing systems 28 (2015).

[6] D. Sculley, M. E. Otey, M. Pohl, B. Spitznagel, J. Hainsworth, Y. Zhou, Detecting adversarial advertisements in the wild, in: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011, pp. 274–282.

[7] P. Ruf, M. Madan, C. Reich, D. Ould-Abdeslam, Demystifying mlops and presenting a recipe for the selection of open-source tools, Applied Sciences 11 (2021) 8861.

[8] S. Mäkinen, H. Skogström, E. Laaksonen, T. Mikkonen, Who needs mlops: What data scientists seek to accomplish and how can mlops help?, in: 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN), IEEE, 2021, pp. 109–112.

[9] H. Baniecki, W. Kretowicz, P. Piatyszek, J. Wisniewski, P. Biecek, dalex: Responsible machine learning with interactive explainability and fairness in python, arXiv preprint arXiv:2012.14406 (2020).

[10] M. Treveil, N. Omont, C. Stenac, K. Lefevre, D. Phan, J. Zentici, A. Lavoillotte, M. Miyazaki, L. Heidmann, Introducing MLOps, O'Reilly Media, 2020.

[11] R. Sothilingam, S. Eric, Modeling agents, roles, and positions in machine learning project organizations., in: iStar, 2020, pp. 61–66.

[12] R. Sothilingam, Analyzing Organizational Processes in Machine Learning Projects: Exploring Modeling Approaches, Ph.D. thesis, University of Toronto (Canada), 2020.