WASP Software Engineering and Cloud Computing
Assignment 2

Konstantin Malysh
konstantin.malysh@cs.lth.se

September 30, 2022

1 Introduction

My PhD project is called "B2B Data Sharing for Industry 4.0 Machine Learning" and is mainly directed at designing a tool that would help industrial specialists, especially the Machine Learning engineers, to operate the internal and external data sharing operations. For software engineering collaboration tools like Git, Jenkins and Gerrit, provide a low-threshold entry to Open-Source Software. For data ecosystems, there is, for example, CKAN, an open source data portal, which is used for open government data and for research data. However, for B2B, where commercial trade-offs are involved, we have not found any general purpose tools, and its data need a different focus on security and integrity. My goal is to identify four key dimensions which tools for data ecosystems should support: data protection, data access, change support and data enhancement possibilities. The outcomes are meant to be the systematic guidelines for data ecosystems tools to support B2B sharing and collaboration, aligned with business models, to improve machine learning for Industry 4.0.

2 Discussions

Looking at the outlines of my research, Requirements Engineering area immediately looks like a very adjacent topic that I will have to face throughout the whole duration of my work. The IT companies that will be interviewed or researched in one way or another, whether it would be a case study or just a set of interviews or surveys (these come together and usually in a specific order to draw better conclusions) will be mentioning the different tasks and guidelines that the desired framework would follow, their concerns and outer-world limitations. However, it has to be important to not drown in the chaos of the requirements suggested by the multiple companies; also the requirements will be evolving through the development process, due to the newly discovered issues,

technical limitations or sudden realizations of the people of industry, so the goal of having all the requirements before the stage of the development might be tricky.

Security and privacy in the scope of the topic of my research are between the key dimensions for the developed tool - in the current era sharing an unencrypted piece of data could be dangerous and could cause damage to many parties involved. The potential research question for this area would be to define the common encryption solution that would satisfy all the users or providing a sufficient set of security tools to let the companies choose from in order to satisfy their needs and goals.
As for the privacy, it is important to provide the users the functionality to cover all the potential privacy types of the data spectrum: whether it is open, shared or closed (as the tool is also meant to be used internally-only), with access to anyone, public access, group-based access, named access or internal access. [1]

It will also be important for me as a tool designer to include the space for the potential changes in the future, so the others could continue the project maintenance and redevelopment. As the tool relies on the external sharing which, one of the core issues would be to tune the regulations from the all the parties, which could have different practices or be under different regulations (perfectly, the goal of the tool is to be used internationally). As the laws and regulations constantly change and evolve (for example, GDPR is fairly recent, however it does not apply to the USA which is a huge source of data companies).

3 Software Engineering future trends

I have worked as a Machine Learning engineer for the past couple if years before switching to the research in Software Engineering. Despite the Machine Learning, Data Science and Software Engineering (especially the first two in the industrial sense) are very adjacent, I cannot see any of the individual areas absorbing others. For example, for my research the potential ML-based issues are related to versioning, testing and reusability.
One of the core issues of versioning of the Machine Learning models is how large they are and how costly it is to rerun all the code to perform tiny adjustments on the model itself or on the data. Although there are some Machine Learning specific versioning tools, the most widely-used ones (e.g. GitHub) are not really suitable for the ML approach, and the ones that do exist are usually serving different purposes, making the future Software Engineering research in the area quite important.
Nearly the same approach can be applied to testing; however, another issue that arises is that the testing of Machine Learning products (perfectly) should be different from the classical Software testing, due to the probabilistic nature of the mathematics inside the model, there are many very niche factors to be accounted for when testing the model itself, which will not be suitable for the common Software Engineering testing routines.

[1] https://theodi.org/about-the-odi/the-data-spectrum/

This being said, in my opinion neither the Machine Learning will absorb Software Engineering, nor vice versa, due to the distinct (although overlapping) natures and sizes of both areas.