ASML Enhanved Code Generation

Ethical Design v0.0.1

Erik van de Laar Victor Doychev Kaloyan Preslavski
Aleksandra Krasteva Emilia Stoyanova

April 11, 2024

Contents

1 Introduction **3**

2 Questions **4**

2.1 Are you honest about how your product works? 4

2.2 If the environment was your client how would your product change? 4

2.3 What could a bad actor do with your technology? 5

2.4 What happens if we do not solve the problem? 5

2.5 What is the biggest negative effect your product can cause? 5

1 Introduction

During our "Advanced Software" semester, we were tasked with completing a TICT ethical game, in order to expand our knowledge and to guarantee proper ethical compliance. Our semester mentor brought a pair of "cards" with and distributed them to us. Each card contained a question on the front, together with a handful of subquestions on the back.

We were required each to pick a question we found suitable for our project. Then we would all investigate it, discuss and come up with appropriate comments about. This was as this semester we are tasked with optimising a code generation tool, developed by ASML. That means that not only is the product, we are working on, an internal one that does not face external clients, but also: our work is contained to pure performance beast-of-burden labour.

That constrained our possible choices a little, as a lot of prescient topics around ethics, could not be tackled during our exercise. We were however, able to pick a few appropriate questions by which we were able to investigate ethical dilemmas, around topics such as environmentalism and supply chains and stark security guidelines.

Furthermore, we were notified that other than the quick card game we played together with our mentor on the 8th of April 2024, we were also guided towards an online tool with which we could also assess our ethical footprint[1]. The report below catalogues our questions and the answers to them; however, it does not include the online questionnaire.

[1] https://www.tict.io

2 Questions

2.1 Are you honest about how your product works?

Taking into consideration that the project is sophisticated and the task for this semester, our group should make sure that we understand how the product works before we start optimizing.

After an extensive and detailed research and analysis of the source code, our group has managed to gain a lot of knowledge on how the generator works. However, since we have dove into the code just recently, there are still blind spots. Due to that we cannot guarantee that any change which we make and is not covered by unit tests won't affect the functionality of the generator. To prevent any unnoticed change in the functionality we will be honest with our work and our group is going to document each step that we took while working on the project. This approach will allow ASML's employees to track our work.

2.2 If the environment was your client how would your product change?

This question is quite topical, given that it seems to be in concord with the direction we are already headed towards. Unfortunately, CPUs are very wasteful in terms of energy. That results in them releasing most of that energy as heat into the atmosphere, rather than using it productively for their designated task. As such, any reduction in the complexity of a given software contributes to making the software more environmentally friendly.

The paragraph above tries to illustrate cleanly how this question has affected us practically and how it gets us to actively think about our impact. Furthermore however, to properly answer the question, we would need stray quite far away from reality and speculate a little.

Firstly, we may look deeply into changing the implementation detail of our software. The current application using a very outdated *JVM* and an obscure programming language by the name of *Xtend*. Although, these technology choices might have made sense for their given requirements in the past, in 2024, this seems like a recipe for disaster. Using something still safe, but much more optimised like Rust might be a possible avenue, one could explore.

Moreover, the famous computer scientist Joe Armgstrong has states in the past that if heat generation is of upmost priority, designing an *ASIC* and running the program on it is, quite frankly, the most reasonable choice. On the other hand, one obviously also has to consider the human capital involved in such a venture. A change of this magnitude would require a great many hours of human work; that itself presupposed all of them contributing to pollution due to daily commutes.

Overall this is one large decision which has striking consequences. An executive should take great care in following with such a decision if it comes to that. But companies that deal with software that is connected to industrial machines of this calibre should be well prepared to deal with such decision.

2.3 What could a bad actor do with your technology?

ASML has a pretty tight connection to the world's economy. Thus, if something were to happen to the codebase, it could have devastating effects on that. However, we do not forsee that the DIPgen codebase can be used to reverse-engineer the lithography machines. The DIPgen application is only used to create templates for the machine. On that note, it may be possible that the trade secrets revealed within DIPgen may affect ASML's operations if revealed.

A possible risk could be that a third party gains access to one of the project members' computer. The bad actor may slip in a commit to the Gitlab repository. It is therefore paramount that the code submitter and reviewer both look at the code in depth before it gets merged into the main branch. That said, ASML will most likely not carelessly adopt our changes into their codebase. They will most likely review our changes and documentation, and decide whether or not it should be adopted. They will most likely make the changes themselves as well, based on our input.

2.4 What happens if we do not solve the problem?

Given that the problem we have at hand is complicated and requires extensive research and time to fully understand the issue and how it can be solved, we expect to not be able to solve the problem and reach a 50% performance increase. In the cases where we are not able to have a positive result, but also in the cases in which we do succeed, everything will be documented properly, so that the next group working on this project can continue from where we left off. To ensure that our solution is working, we will be using the provided benchmarks and unit tests.

What is more, the problem itself, can not completely be solved by caching or loop optimization or the other improvements we plan on doing, but requires some changes in the architecture and the overall structure of the code. Because of that, it can be said that we are not truly solving the problem, but rather patching the symptoms.

2.5 What is the biggest negative effect your product can cause?

Mistakes in our project, which is crucial for chip manufacture and development, might have serious consequences. Given the impact of the chip shortage (2020-2023), which led in major price hikes in the automotive and consumer electronics sectors, we feel that if a malfunction or defect is overlooked during the development process, something similar may occur, albeit not to the same amount.

To prevent this, proper testing of the project needs to be put in place. Unfortunately, what we noted is that testing for the code generator is incomplete and this could lead to mistakes creeping in and causing problems in the future.

A precaution we can take is to document everything that we change and modify. This approach will help all team members, as well as ASML to be informed and detect possible problematic code. By doing this we improve our ability to detect and address issues promptly, this way preserving the integrity and reliability of the project.