Open Peer Review on Qeios

[Commentary] Programming and Kant

So Katagiri

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.

Abstract

This exposition delves into an imaginative comparison between Immanuel Kant's iconic philosophical treatises and modern-day programming paradigms. Through an episodic narrative, we commence by allegorizing the data-model layer dichotomy in programming to Kant's discourse on intuition and enlightenment in his Critique of Pure Reason. Here, the act of data extraction and processing is juxtaposed against Kantian notions of intuition and formalism, rendering a profound analysis of data's abstract nature. The narrative advances to the Critique of Practical Reason, drawing parallels between Kant's moral law and programming conventions. The emphasis is laid on the potential transcendence of individualistic coding conventions to a universal moral code within the programming community, hinting at the obligation towards a collective ethical coding practice. The exploration culminates in aligning the Critique of Judgment with the aesthetic and organic attributes of code, positing a philosophical quest for the 'ultimate good code.' This venture not only sheds light on programming's practical and ethical dimensions through a Kantian lens but also whimsically ponders upon the sublime beauty residing in the art of coding, encouraging a profound reflection within the computational realm. Through this narrative, we invite programmers and philosophers alike to engage in a discourse where code meets philosophy, and perhaps, find an ethical and aesthetic enlightenment in the heart of the digital cosmos

So Katagiri

Nature and Environment, Faculty of Liberal Arts, The Open University of Japan, Chiba 261-8586, Japan

Introduction

The air had been buzzing all morning, but I decided it wasn't going to rain, so I took my laptop out. After paying for the coffee, I remembered that I had bought a coffee ticket. I should have taken out the ticket. But maybe I shouldn't have. I thought about it and realized that it was from the next store.

Critique

When I read Yasuhiro Sato's commentary on Kant's Critique of Pure Reason[1], the story of intuition and enlightenment seems to me to be a story about using object orientation to extract the appropriate data formalism from the data layer and process the data in the model layer. The data layer is an abstract concept from the computer's point of view, and if you think that it is different from the actual physical reality, the other side of such interface looks like an unreachable object itself. The entrance to the data is intuition, the form of the data is the formalism that intuition receives, enlightenment is the model layer, the class of objects processed in enlightenment is the category (or diagram), and compositional power is like a business module, right?

Then, what we are criticizing in the Critique of Pure Reason, i.e., the limits of reason, is like this. Even if we consider only classes as an ideal without actual data and think about the relation of objects, if we ultimately pursue this and create a model, we will always end up with a design that is mutually contradictory (antinomy). This is where transcendentalist apologetics comes in. In this antinomy, rational theology, rational cosmology, rational psychology, etc. are questioned. In particular, in this rational psychology, the belief in I or Descartes' I am is one of the concepts of enlightenment, which is always involved in Kant's intuition. Namely, if we assume that one of the functions of the model layer is to censor data, and that I am is the function of the model layer, then this function is a check on the data.

So, although we fall into antinomy with reason alone, we still have to practice programming, and Kant's Critique of Practical Reason is an approach to this practice (really?). Programming is practical, and it is not enough to talk about strict formalism. However, Kant's argument is that what is important is not the purpose of programming but rather the conventions (morality) (please don't believe me).

These conventions are like coding rules, design patterns, or DRY rules, but merely having each person follow his or her own conventions does not constitute morality. This state of affairs is called case rate. So what must be done for the maxim to be elevated to the status of morality? This is Kant's famous moral law.

Act so that the maxim of your will may always at the same time be the principle of universal legislation.

In other words, Kant said, Make sure that the way you are programming makes sense no matter which programmer is doing it. Kant said. Don't ask for what, because this is a categorical imperative! (Like, Because some big guy told me to do it.)

(Like someone told you to do it.) Yes, writing code like that does not promise to make programmers happy, but it does make programmers who follow the rules happy. But programmers who follow the rules should be happy. Good code is not good code if it does not make the programmer happy!

In this way, the highest good is demanded as the ultimate coding style, one that both adheres to universal conventions and brings happiness to the programmers who program it. In other words, in the world of programming, there may not be a God (logical God) in the display, but there may be a God (ethical God) who embodies the ultimate good code. In our minds, that is.

Final

You may have guessed by now that the last of the three critical books, The Critique of Judgment, is a treatise on beauty and the organism, that is, the beauty that resides in the code.

References

1. ^*Yasuhiro Sato, "The Concept of Life in the History of Philosophy", Open University of Japan Graduate School Teaching Materials(2010); https://ci.nii.ac.jp/ncid/BB01564953*