Analytical Prompt Essay

Colbyn Wadman

December 24, 2020

1   For the Reader

I'm replying to the following:

> 3."The aim of argument, or of discussion, should not be victory, but progress." – Joseph Joubert. Sometimes, people talk a lot about popular subjects to assure 'victory' in conversation or understanding, and leave behind topics of less popularity, but great personal or intellectual importance. What do you think is important but under-discussed? Take care not to sermonize on the topic you choose.

2   What do you think is important but under-discussed?

Technical Computing & Simulations

There is this very interesting video essay by the Heme Review Podcast (i.e. Dr. Bernard) called "A Future Of Medicine". In this essay, Dr. Bernard explains how our current medial system works and some more ideal hypothetical system, respectively. What is important, and the thesis of his essay, are the flaws of our

current medical system (which he explains in exquisite detail). Overall, what is interesting -I think- is how our modern evidence based system isn't as ideal as I once thought, and this is due to it's dependence on population data, so such may not be directly applicable to a given person. For example, if a trial was done on a bunch of men, the speaker said clinical experience suggests that there will be differences for women taking the same medication. Whereas, his proposed hypothetical system disregards the need for such models VIA personalized simulations of a given individual, and I suppose all their biological processes. I.e. if this technology was available, there would be no need to study the efficacy VIA population models, because with this hypothetical technology, such simulations would be far more accurate, would be far more personalized to the biochemistry of the given individual, and not from generalizations of population data. Furthermore, if possible, he said, such could likewise -hypothetically speaking- be extended to drug development VIA optimization algorithms. [Bernard]

The above review is a fragment from an old english essay last summer. The video itself is fascinating! (I even showed it to my Mom and she liked it.) Overall, I wish more of such content would get produced and discussed.

Open Source Software

Meta

The following essay has some history to it. Originally, this was written for an english essay last summer.

Some bits have been improved and others removed. But you can find the original version over here https://github.com/colbyn/DC-Essay.

Open Source Software

I propose that there exists a community quite unlike anything you have probably seen from past students. Still in it's infancy, this community has nevertheless become a foundation of the IT industry. As it matures it will become a cornerstone of efficiency gains in all STEM fields. As our collective knowledge of all technical matters get reduced to mere functions -mapping problems to solutions- this community will become the librarian, troubleshooter, curator, documenter

and coordinator of such knowledge.

This community -the printing press of computable knowledge- is the rather humble open source community and industry.

For a brief overview for the layperson: the open source community -as the name implies- is a community built upon open source software, where open source software may be regarded as software built from publicly accessible source code. The open source community may be regarded as the 'human aspect' of open source software. As in communication facilities between the producers and consumers of such software. As well as mechanisms for updating the source code that implements such software.

My journey in the open source industry began when I worked at a former startup called upLynk (now absorbed into what's called Version Digital Media Services). At upLynk there was a rule of only using free and open source software, as is common in the IT industry. This wasn't necessarily because it's free (the company was immensely profitable as is the industry overall), but due to a multitude of factors. For instance, open source software can be a means of industry standardization. Furthermore, for software that upLynk replied upon, my boss even encouraged me to submit any bug fixes I found in such software back to the original author. This is because it's generally in the companies interest to do so, because otherwise their version of this software component will be 'out of sync' with the official version, so any updates from the 'official version' will therefore need to be merged into this nonstandard version, if they wish to benefit from such updates.

To justify my overall thesis, consider the open source ecosystem as a system where discrete units build upon other discrete units and therein produce more complex non-discrete units. This system permits for abstraction, so complex non-discrete units may be abstracted into simple discrete units. Thereafter this process of production and abstraction enables further production and abstraction and so forth. Each iteration or generation may be considered to be more sophisticated than prior generations, given that each generation is a product of prior generations... From this analogy, you can imagine these bottom-up and cumulative processes will eventually give rise to very sophisticated products, and perhaps one day, akin to how emergence gives rise to the complexity found in nature.

From personal experience, my https://imager.io project wouldn't be possible without the various open source components it's built upon. Simply because my time is finite, and especially because lower-level encoding details are just **too**

complicated for me to understand and implement on my own. I am nevertheless able to compose such components into a larger and more sophisticated end product, from the preexisting output of resources and information from the global open source community. Overall added value that may be considered to be greater than the sum of its components, and therefore emergent in a manner of speaking. In an old English paper I likened the open source community as "the printing press of computable knowledge", and perhaps even more significant than the advent of the printing press itself, because as the industrial revolution introduced a force multiplier of human muscle, so too does abstraction introduce a force multiplier of the human mind.

Because while a book may describe a life's work in mathematics and abstractions therein, the medium is itself rather passive. A book may describe a life's work in applied mathematics, yet a mind is required to manifest its application. Whereas, imagine a medium where the most knowledgeable of experts can record their understanding of a given domain as functions that map problems to solutions, in a manner that can be utilized by any layperson, and thereafter this record can be reapplied, reused, and so forth, forever thereafter. This is software, while the open source community is what facilitates the overall adoption, use, and further development of such products.

Certainly a topic seldom discussed, yet it's likely that for instance, the browser you are currently using is either 100% open source, or is based on lower-level open source components.

Given the topic, I suppose it goes without saying that this essay has been open sourced, and is available on GitHub:
https://github.com/colbyn/analytical-prompt-essay-2020.

Likewise, the original version can be found over here:
https://github.com/colbyn/DC-Essay.