

Assignment 10

Alex Clemmer

Student number: u0458675

Olga Filippova, Andrey Filippov [sic?]*—Elphel*

The name of the talk is “High-End Open Hardware for Scientific Research”, but it’s too bad that the speakers didn’t talk much about *why* Elphel believes that Open Hardware is interesting or useful compared to purely proprietary hardware, or what possibilities Open Hardware make possible for the consumer. It’s too bad because they have really, really good, really interesting reasons for doing only Open Hardware, and their clientele has found some notable and interesting extended uses for this Open arrangement.

For those of us who don’t know, their 353 is the magic behind Google books, and their Eysis is the magic behind Google Maps. One good question is *why* this is the case. Especially for the former, which really just requires a regular camera to take pictures of book pages, Google probably could have picked any company they wanted to make a camera for them. Arguably bigger companies could have provided a better, more reliable, more feature-rich software interface, with better-tested, and possibly better-produced, cameras. Why Elphel at all?

In the first place, software from bigger firms — this is true, at least, for Nikon, and usually even more true for more scientifically-oriented camera companies — is almost uniformly terrible. They have the resources to build better software, but they don’t seem to know how to do it. The fact that this software is completely closed makes this picture much worse—if the camera software doesn’t do what you want it to, well, too bad. OSS is not always great, but the places where it is best tend to be where there is a communal need for something, but for whatever reason the proprietary software either is not, or does not need to be, competitive.

The 353 is also a very simple camera, conceptually. It’s high-quality, and takes high-quality images, but it definitely does not suffer from feature bloat. This in turn makes the software for it also very simple—a Nikon D300, for example, will have half a dozen different embedded devices that most cooperate with each other to provide a seamless user experience. Designing software for this sort of system is difficult, and by its nature more complicated. The 353, in contrast, is really just a couple of devices that perform most of these functions natively. Its goal is not to provide a user experience, but to provide a simple interface for getting images out of the world.

The advantage for any company that needs a camera conceptually as a part, rather than an end-of-the-line product, is unequivocal. Google used this camera to process millions of pages of books before Google Books was eventually ordered by the courts to stop. They snapped pictures of pages quickly and efficiently, and piped them directly from the camera to the larger Google Books infrastructure, and they were able to do it largely because they had control not just over the precise output of the camera, but because the camera and its software were simple enough that managing it was a tractable problem. This would not have been possible if they had to struggle with the (usually) awful proprietary software provided by the camera company. If they had been required to rely on the camera company to provide the software, it would have taken more time, more money, and almost certainly would have been less effective.

Craig Silverstein, the first real employee that Larry Page and Sergey Brin hired, has a good account of this problem in talks I’ve seen on YouTube. I am very willing to bet that the Google Maps situation is very similar to this, even though no one has explicitly told me this is the case.

I’m not prepared to say that OSS and OH are the solution to every problem. There are still some things that proprietary options are just better at, and the burden of the Richard Stallmans of this world is to show that OSS and OH, in the right system, necessarily are as good or better than their proprietary counterparts, or that there is some harm done by proprietary software that eclipses the fact that it is better. I will say that they tend to respond to different incentives—generally really good proprietary software has very good reasons for being really good, where in my experience OSS tends to be good precisely because proprietary software has failed for some reason or another.

As a company, Elphel has really been placed in a reasonably unique position to talk about these advantages and disadvantages, and while I am greatly interested in what they had to say, I do notice that their opinions about how being Open has changed the game (of their internal development, of the possibilities offered to companies like Google, and so on) are noticeably absent. The truly unfortunate thing is not that this is missing from their talk, but that there aren’t a lot of other people who could have said it.