Marx, Anarchism, and Web Standards

Steve Klabnik

2011-12-15

Contents

An aside for software devs 3
Domain Specific Languages and information density 3
Marx and the labor theory of value 4
Anarchism and ‘anarcho’-capitalists 5
Web standards 8
It’s good for you! 9

An aside for software devs

You might not care about anarchism, and I can almost guarantee that you don't care about Marx, but please bear with me. I think my point is best made by putting the web stuff at the end, so please, just read it. ;) It'll be good for you, I swear. I'll explain better at the end.

Domain Specific Languages and information density

I've been noticing an effect lately that's certainly not new; it's just come up frequently. When working within a domain that's new to them, most people tend to not respect the density of the information being given to them, and it causes them to draw incorrect inferences.

For example, this tweet earlier:

> Does anybody else's head explode when they read an "unless" statement? What good is readability if comprehension goes out the window?
>
> — Christopher Deutsch (@cdeutsch) December 15, 2011

And this response:

> @cdeutsch There are weirder idioms. @users.collect{|u| u.email} vs @users.collect(&:email) :) That should weird you out more.
>
> — Brian P. Hogan (@bphogan) December 15, 2011

Now, I'm not saying that Christopher is in the wrong here, in fact, he agrees with what I'm about to say:

> @philcrissman @bphogan I'm sure I'll get used to the "unless" pattern someday. But from a Ruby outsiders perspective it's hard to comprehend
>
> — Christopher Deutsch (@cdeutsch) December 15, 2011

From someone new to Ruby, `Symbol#to_proc` or `unless` are hard to understand initially, and that's because they've increased the density of information being conveyed. `unless` is the same as `if not` and `&:foo` is the same as `{|a| a.foo }`. Both of these constructs condense something more complicated into something that is simpler, but denser.

You'll note that I said 'simpler,' but by a certain measure, `&:foo` is actually more complex. When I say `&:foo` is simpler, I'm meaning for someone who's well versed in Ruby, functional programming, or first-class functions. I have this exact background, and so for me, `collection.map &:foo` is simpler and more readable than `collection.map {|a| a.foo }`. When I read the first example, I say in my head "Map foo over the collection." You have to grok what map really is to get that sentence or the corresponding code. Whereas what (I imagine) someone who does not have this kind of background thinks when they see the second example is "Okay, so map is like an each, and for each thing in the collection, we're going to call foo on it." This is a totally valid interpretation, but notice how much longer it is, and how much more involved in the details it is. That's cool, but to someone who groks map, it has a much larger amount of mental

overhead, in the same way that my concise explanation causes much more thinking for someone who doesn't grok it.

This happens often in education. DHH made a comment about this recently that illustrates this principle, and he couches it in terms of "learnability" vs. "readability":

> If you optimize a framework for beginners, you're optimizing for learnability. That's great for the first few days or even weeks of learning. But once you're past that, it's not so great. What you care more about is the usability of the thing once you know it. There are plenty of cases where learnability and usability are in conflict. Letting learnability win is a short-term relief.
>
> If you on the other hand optimize for usability, for making things simple and easy to use for someone who understands the basics, you'll often end up with something that has great learnability as well. Maybe not as much as could be achieved as if that was your only goal, but plenty still.

I think that 'learnability' is a pretty good shortening for 'light density' and 'usability' is decent for 'heavy density.' It's the same effect, though. For the rest of this essay, I'll be using 'learnable' and 'usable' to mean this particular usage.

Any time you encounter experts in a particular domain, they'll often have fairly specific language that corresponds to that domain. This language is usually designed to be usable, not learnable. This is because they've already done the learning; learnability holds no utility for them. However, usability is incredibly… useful. To them. They're working at a higher level of abstraction, and don't want to get bogged down in details they already know well. Using learnable language would cause them to take twice as long to say things; to return to the density analogy, dense information is transferred from one person to another more quickly. If you can read a sentence a second, but that sentence is dense, you acquire much more information per second than if it was lighter.

This tendency means that most language that's specific to a domain will generally trend towards the usable at the expense of the learnable. The impact this has on individuals new to the domain, however, is that of a wall. An impediment. Overcoming this obstacle requires a bit of good faith on the part of the beginner; to cross quickly over the chasm between beginner and expert, they must recognize and respect this aspect of the conversations they will invariably become a part of. When faced with a term that is used in a strange way, beginners should ask for clarification, and not start arguments over semantics they don't yet even understand. Experts will recognize these arguments as coming from a place where concepts are not yet fully understood, and while they may recognize the need to help educate, if the newbie is being belligerent, they may just ignore them instead. Nobody wins; the signal/noise ratio has been decreased, the beginner doesn't learn, and everyone's time is wasted.

Here's three other situations where I've seen this happen lately:

Marx and the labor theory of value

Philosophy writing is generally a great example of text that is very much usable, and not at all learnable. Some writers can still be learnable, but most are not, in my experience. One of the

reasons this happens is that they introduce concepts early in a text and then utilize them later without referring back to the earlier definition. This isn't a problem for anyone who's thinking about taking off the training wheels or anyone who reads the entire text. The danger comes in when someone *not* versed in the entire text attempts to take portions of it out of its context.

Consider Marx, and *Capital*. The meaning of 'value' is a central concern of his writing, and indeed, entire critique of the political economy. It's so important that the first few chapters are devoted to an (excruciatingly, frankly) detailed explanation of his thoughts on the true meaning of value. The rest of *Capital* is built on top of this: at least in my understanding of Marx, it all boils back down to that one question. And when having discussions between people who are devotees of Marx and those who come from other schools of economics, this kind of language gets in the way.

It also causes ideas to be misrepresented: the first time I was ever introduced to the labor theory of value, it was by a close friend who's very libertarian. This was a few years ago, so it's an obvious paraphrase, but he summarized it thusly: "Yeah, I mean, the labor theory of value basically says that people should be paid for however much work they put into things, so if I take six hours to make a widget and you take four, the price of a widget from you should be six yet mine should only be four, and ideas like 'market value' should be disregarded. It's totally irrelevant if I suck at making widgets, I should get paid more than you anyway." Which, to put it lightly, is a misrepresentation. While explaining the labor theory of value is outside of the scope of this post, what I will say is that to Marx, 'value' is something intrinsic to an object; it's the 'socially necessary abstract labor' inherent to it. Talk about dense language! What a capitalist would call 'value,' a Marxist would call 'price.'

As you can see, even just one little word, 'value,' can be quite dense! Can you imagine a discussion intended to be 'learnable' to outsiders about what's meant? Imagine the expansion: 'value' -> 'socially necessary abstract labor' -> … Marx is already long enough; Capital would be thousands of pages! Yet to a beginner who flips to Chapter 12, they'll read a sentence that contains 'value' and draw poor conclusions! They wouldn't even realize they're making a mistake, I mean, how could five letters be misinterpreted?

Furthermore, people who haven't read Marx don't generally draw distinctions between his critique of capitalism and his solution: communism. This is annoying when trying to explain to people that I love his critique, but am critical of his answers to its problems; they perceive this weakness in his answer as a weakness in his description of the problem. Furthermore, they then say "but I thought you call yourself a communist?" and I respond with "sure; the issues I have are with the dictatorship of the proletariat, not with the general idea of communism" and then their eyes glaze over and they change the subject. Information density claims another victim…

Oh, and a great example of Marxist economics in a usable form is here.

Anarchism and 'anarcho'-capitalists

Arguments often boil down to these kinds of questions of definition, but one place where I see it happen almost *constantly* is amongst anarchists. I mean, from the outset, anarchists have to battle against the general definition of 'chaos and disorder' versus the domain-specific 'without rulers.' Within that, 'rulers' in anarchism is fraught with the same sort of questions that 'value'

has for Marx. The prime example of this are the terribly misguided 'anarcho'-capitalists, better described as 'voluntaryists.'

Here's the deal: ancaps lack an understanding of the vast majority of historical anarchist thought, and so try to appropriate the term 'anarchism' to describe their philosophy which is decidedly not anarchist. The ones who do have started using 'voluntaryist' to describe themselves, which is a great example of using information density to mislead, but that's a whole separate rant. Here's the 411, from the Anarchist FAQ, which has its own things to say about density when it comes to the language specific to political theory discussions:

> "Anarcho"-capitalists claim to be anarchists because they say that they oppose government. As noted in the last section, they use a dictionary definition of anarchism. However, this fails to appreciate that anarchism is a political theory. As dictionaries are rarely politically sophisticated things, this means that they fail to recognise that anarchism is more than just opposition to government, it is also marked a opposition to capitalism (i.e. exploitation and private property). Thus, opposition to government is a necessary but not sufficient condition for being an anarchist — you also need to be opposed to exploitation and capitalist private property. As "anarcho"-capitalists do not consider interest, rent and profits (i.e. capitalism) to be exploitative nor oppose capitalist property rights, they are not anarchists.
>
> Part of the problem is that Marxists, like many academics, also tend to assert that anarchists are simply against the state. It is significant that both Marxists and "anarcho"-capitalists tend to define anarchism as purely opposition to government. This is no co-incidence, as both seek to exclude anarchism from its place in the wider socialist movement. This makes perfect sense from the Marxist perspective as it allows them to present their ideology as the only serious anti-capitalist one around (not to mention associating anarchism with "anarcho"-capitalism is an excellent way of discrediting our ideas in the wider radical movement). It should go without saying that this is an obvious and serious misrepresentation of the anarchist position as even a superficial glance at anarchist theory and history shows that no anarchist limited their critique of society simply at the state. So while academics and Marxists seem aware of the anarchist opposition to the state, they usually fail to grasp the anarchist critique applies to all other authoritarian social institutions and how it fits into the overall anarchist analysis and struggle. They seem to think the anarchist condemnation of capitalist private property, patriarchy and so forth are somehow superfluous additions rather than a logical position which reflects the core of anarchism.

Part of the problem with the second half of this quote is that I'm such an 'expert' on this kind of language that I don't even know if it'll make sense to you; without the kind of background reading in socialist political philosophies, it might just be gibberish. At least, the first paragraph should be pretty straightforward, and you can take the second as an example of this kind of language.

Giving you a short explanation of why anarchists are against Capitalism is a great example in and of itself of domain specific language and density. Here:

> Property is theft.

This is a quote by Proudhon, the first person to call himself an anarchist. Let's unpack the first few layers of this statement:

Property. Here's Wikipedia's explanation:

> Private property is the right of persons and firms to obtain, own, control, employ, dispose of, and bequeath land, capital, and other forms of property. Private property is distinguishable from public property, which refers to assets owned by a state, community or government rather than by individuals or a business entity. Private property emerged as the dominant form of property in the means of production and land during the Industrial Revolution in the early 18th century, displacing feudal property, guilds, cottage industry and craft production, which were based on ownership of the tools for production by individual laborers or guilds of craftspeople.
>
> Marxists and socialists distinguish between "private property" and "personal property", defining the former as the means of production in reference to private enterprise based on socialized production and wage labor; and the latter as consumer goods or goods produced by an individual.

Whew! There's a few things to note here: Captialists like to pretend that capitalism is synonymous with 'trade,' and not something that started in the 1800s. Likewise, that private property rights are something that has always existed. However, as this alludes to, there are many different kinds of property rights that have existed at different places and times.

So in 'property is theft,' Proudhon is referring to private ownership of the 'means of production.' Let's expand that. Again, Wikipedia:

> Means of production refers to physical, non-human inputs used in production—the factories, machines, and tools used to produce wealth — along with both infrastructural capital and natural capital. This includes the classical factors of production minus financial capital and minus human capital. They include two broad categories of objects: instruments of labour (tools, factories, infrastructure, etc.) and subjects of labour (natural resources and raw materials). People operate on the subjects of labour, using the instruments of labour, to create a product; or, stated another way, labour acting on the means of production creates a product. When used in the broad sense, the "means of production" includes the "means of distribution" which includes stores, banks, and railroads. The term can be simply and picturesquely described in an agrarian society as the soil and the shovel; in an industrial society, the mines and the factories.

We could continue to discuss how to distinguish between this 'private property' and 'possessions,' which anarchists are *not* against, but I'm just trying to demonstrate that this word 'property' is incredibly complex.

Okay, so Proudhon claims that 'owning the physical inputs used in factories used to produce wealth is theft.' I could expand on 'theft,' but really, I think my point about density is made. For more on this, see Why are anarchists against private property?.

Web standards

Web standards are another great example of a domain that has a lot of very specific language. And one that people often think they can grab random chunks out of and quote without fully understanding the rest of the context.

This is going on right now with Rails. There's a discussion about if the `PATCH` HTTP verb should get support, and if it should be the verb that matches to the `update` action or not. It's ended up in a discussion about the semantics of `PUT`, which has resulted in a lot of random quoting of standards documents by myself and others. Here's some running commentary on some of the comments. It's impossible to do this without it becoming semi-personal, so let me just say upfront that I think everyone is participating honestly in this discussion, but I think it's a great example of people who aren't familiar with a domain jumping in and drawing incorrect conclusions.

First up, benatkin comments:

> I googled for HTTP verbs and clicked the first result and PATCH isn't listed.
>
> http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html
>
> Where is it?

Anyone participating in a discussion about `PATCH` should be reasonably familiar with `PATCH`. Learning where `PATCH` is defined is as simple as Googling HTTP PATCH, which shows it being defined in RFC 5879. With that said, this is a good example of asking for clarification, and not immediately progressing into an argumentatitive "It's not in HTTP 1.1, so it's bullshit!" style of learning where `PATCH` is being defined.

Of course, the thread starts to dissolve later, when @stevegraham mentions:

> i must be the only person in the world that disagrees with "PUT requires a complete replacement", as per RFC2616 "HTTP/1.1 does not define how a PUT method affects the state of an origin server"

Now, Steve is an awesome guy, but he's a bit misguided in this case. This is a great example of drawing an incorrect conclusion based on one sentence out of context. He's not *wrong* in a strict sense, RFC2616 does contain that line. However, this is because HTTP defines the semantics of communication, and the semantics are 'idempotent creation or replacement.' The fact that HTTP does not define how PUT affects state is irrelevant, an entire representation is needed for idempotency reasons. PUT's definition also says pretty clearly:

> The PUT method requests that the enclosed entity be stored under the supplied Request-URI. If the Request-URI refers to an already existing resource, the enclosed entity SHOULD be considered as a modified version of the one residing on the origin server. If the Request-URI does not point to an existing resource, and that URI is capable of being defined as a new resource by the requesting user agent, the origin server can create the resource with that URI.

"The enclosed entity be stored" is pretty straightforward: it needs an entity, not a portion of an entity. Furthermore, how is a server going to create a resource without a complete representation if the resource didn't already exist?

In this case, the standard's organization also doesn't help: if you just read the section titled `PUT`, you wouldn't get the full understanding, since the fact that it's safe and idempotent is mentioned above in the section regarding those two things. I'm not sure why those aspects aren't in each definition, and are in a different section above, but the point is that you need to consider the full document in its entirety to understand the semantics of `PUT`. Steve is only reading one section and then extrapolating from there.

There's a lot more in that pull request, but one more example: Konstantin points out that Rails supports a lot of things that aren't standards:

> You mean Rails should not support proposed standards like, say, Cookies?

Yep. Did you know that? Cookies aren't actually a standard, just a proposed one.

Anyway, I also want to say this: I am not a perfect interpreter of these things either. I often change my opinions after learning more things, and I think this is a good thing. There's nothing the matter with being wrong; it's how you handle it that matters. The discussion in this thread continues. RFCs are also not perfect, and do have wiggle-room; but it's important that agreements are followed. Amongst people who discuss REST and HTTP, the fact that PUT requires a full representation is not controversial; it's simply understood as true.

It's good for you!

I don't want to turn anyone off from learning new things; exploring new domains is a great path towards personal growth. I've said a few times that I think more programmers should read Marx, and it's because I think this experience of jumping into the deep end of a new set of language that you don't fully understand is a tremendous educational experience. But to truly learn, an open mind and open ears are crucial. Making arguments on semantics doesn't work if you don't understand the context and semantics of words, as words change significantly when placed in different surroundings.

Steve Klabnik
Marx, Anarchism, and Web Standards
2011-12-15