Code, and Other Laws of Wikipedia

September 18, 2006

Original link

Wikimedia 2006 Elections

Part 1: Wikimedia at the Crossroads
Part 2: Who Writes Wikipedia?
Part 3: Who Runs Wikipedia?
Part 4: Making More Wikipedians
Part 5: Making More Wikipedias
Part 6: **Code, and Other Laws**
If you translate this essay, please contact me.

Vote for me in the election for the Wikimedia Foundation's Board of Directors.

Code is law, Lawrence Lessig famously said years ago, and time has not robbed the idea of any of its force. The point, so eloquently defended in his book *Code, and Other Laws of Cyberspace*, is that in the worlds created by software, the design of the software regulates behavior just as strongly as any formal law does; more effectively, in fact.

The point is obvious in some contexts. In the online 3D universe of Second Life, if the software prevents you from typing a certain word, that's a far more effective restraint on speech in that world than any US law could ever be in ours. But the point is far more subtle than that; it applies with equal force to the world of Wikipedia, the thriving community and culture that our wiki software creates.

For one thing, the software decides who gets to be part of the community. If using it is clear and simple, then lots of people can use it. But, if it's complicated, then only those who take the time to learn it are able to take part. And, as we've seen, lots of intelligent people don't even understand how to edit Wikipedia, let alone do any of the other things on the site.

For another, the software decides how the community operates. Features like administrative controls privilege some users over others. Support for things like stable revisions decide what sorts of things get published. The structure of talk pages help decide what and how things get discussed.

The page design the site uses encourages specific actions by making some links clear and prominent. Software functions like categories make certain kinds of features possible. The formatting codes used for things like infoboxes and links determine how easy it is for newcomers to edit those pieces of the site.

All of these things are political choices, not technical ones. It's not like there's a right answer that's obvious to any intelligent programmer. And these choices can have huge effects on the community. That's why it's essential the community be involved in making these decisions.

The current team of Wikipedia programmers is a volunteer group (although a couple of them were recently hired by the Wikimedia Foundation so they could live a little more comfortably) working much like a standard free software community, discussing things on mailing lists and IRC channels. They got together in person in the days before Wikimania to discuss some of the current hot topics in the software.

One presentation was by a usability expert who told us about a study done on how hard people found it to add a photo to a Wikipedia page. The discussion after the presentation turned into a debate over *whether* Wikipedia should be easy to to use. Some suggested that confused users should just add their contributions in the wrong way and a more experienced users would come along to clean their contributions up. Others questioned whether confused users should be allowed to edit the site at all — were their contributions even valuable?

As a programmer, I have a great deal of respect for the members of my trade. But with all due respect, are these really decisions that the programmers should be making?

Meanwhile, Jimbo Wales also has a for-profit company, Wikia, which recently received $4 million in venture capital funding. Wales has said, including in his keynote speech at Wikimania, that one of the things he hopes to spend it on is hiring programmers to improve the Wikipedia software.

This is the kind of thing that seems like a thoughtful gesture if you think of the software as neutral — after all, improvements are improvements — but becomes rather more problematic if technical choices have political effects. Should executives and venture capitalists be calling the shots on some of these issues?

The Wikipedia community is enormously vibrant and I have no doubt that the site will manage to survive many software changes. But if we're concerned about more than mere survival, about how to make Wikipedia the best that it can be, we need to start thinking about software design as much as we think about the rest of our policy choices.