

Copilot, Copying, Commons, Community, Culture

by Robert F.J. Seddon, honorary fellow of University of Durham

Beyond enthusiasm for the recognition of communal rights in art [...] lie unexplored legal paths, some of which hint at troubling possibilities. Can the emergent notion of communal intellectual property be limited to Aboriginal communities, or will it inevitably spill over [...] to other self-defined “tribal” or ethnic groups?ⁱ

Programmers are not united by ethnicity, but they do share bonds of culture; and the gift economies of Free Software and Open Source Software, in particular, evoke other and older communities within which artistry and craftsmanship are practised and their knowledge shared and cultivated. For programmers are not the first to see their knowledge and techniques bundled up by a commercial enterprise, transformed, and sold back to them as a ‘new’ commodity. Indigenous peoples have voiced grievances when their botany or their art is commercialised for others’ benefit, and a substantial and growing body of legal and philosophical thought has emerged in response. Though the disanalogies are not small, in the context of Copilot the resemblances are telling—and serve as a warning to copyright minimalists.

Culture

While Copilot’s suggestions can occasionally be traced back to specific human authors, this is the exception to the general rule: that Copilot hoovers up the *collective* efforts of GitHub contributors as its learning corpus. It could not be otherwise: to foster sharing and collaboration is one of the benefits of FOSS. If anyone is wronged by Copilot or suffers infringement of rights, therefore, it is through participation in collective activity.

On the one hand, it does not seem that the many and various people and organisations independently putting their code on GitHub form some kind of corporate victim. Yet on the other it misses something to consider them merely as a class of similarly affected discrete entities. It might be useful to consider them as such for the purposes of, say, class action law (I am a moral philosopher, not a lawyer, and therefore have no comment on such matters), but we should respond to a dragnet by considering the welfare of the software ecosystem as well as identifiable harms done to particular individuals.

Individuals who are acting independently of one another and pursuing their own interests may nonetheless share mutual affinities. The act of publishing and licensing code so that others may benefit creates such an affinity: it is a public-spirited act, and a form of participation in a *culture* which fosters and values such acts.

There have been various, perhaps numerous, characterisations of programmer culture at its most general. Clive Thompson has described coders as the ‘new tribe’ who have remade the world.ⁱⁱ Erik Brunvand uses ‘the culture of the computer programmer’ semi-interchangeably with hacker culture.ⁱⁱⁱ A recent draft paper by Tomas Petricek considers how ‘cultures of programming’ take the forms of hacker culture, engineering culture, managerial culture and mathematical culture.^{iv}

Over and above that, programmers may explicitly identify themselves and their reasons for sharing code with an *ethos*. Richard Stallman defines Free Software (contrasted with Open Source) as ‘a movement for freedom and justice’.^v The hacker ethic named by Steven Levy has been philosophically explicated by Pekka Himanen as a ‘new work ethic’.^{vi} E. Gabriella Coleman places hacker culture within traditions of liberalism:

Free software hackers culturally concretize a number of liberal themes and sensibilities — for example, through their competitive mutual aid, avid free speech principles, and implementation of meritocracy along with their frequent challenge to intellectual property provisions. Indeed, the ethical philosophy of F/OSS focuses on the importance of knowledge, self-cultivation, and self-expression as the vital locus of freedom. Hackers bring these values into being through an astounding range of social and technical practices...^{vii}

GitHub in turn positions and promotes itself in communitarian style, as seen for example in the description it presents to search engines, which opens with an appeal not merely to *a* community on its website but to *the* global community it purports to serve: ‘GitHub is where over 65 million developers shape the future of software, together. Contribute to the open source community...’^{viii}

Collectivity

Institutions and principles of intellectual property are not designed to support the claims of collectivities which have blended their contributions into an intellectual commons. This observation is not a new one.

James Boyle, subsequently notable both as a theorist of intellectual property and the commons of the mind and as a founding board member of Creative Commons, has written that public goods problems arise because the knowledge of indigenous shamans ‘can find no place in a legal regime constructed around a vision of individual, transformative, original genius’.^{ix} Consequently ‘tribal lore and biological largesse find no place in the language of intellectual property’.^x

That coders are a ‘new tribe’ as per Thompson is an idea which can only be pushed so far: programmers collectively share neither lineage nor lands, and their ‘tribal lore’ involves no single language or religion to unite them. (The Church of Emacs is not, in practice, catholic.) Yet it helps, I suggest, to explain the misgivings which have arisen from Copilot’s announcement: the uncertainty whether Copilot is altogether above board, when, as the FSF’s call for papers puts it, ‘even if everything might be legally copacetic, activists wonder if there isn’t something fundamentally unfair about a proprietary software company building a service off their work.’^{xi}

The WIPO has been trying for years to reach agreement about whether Indigenous peoples should be able to ‘own’, or otherwise claim compensation for the commercial exploitation of, their traditional knowledge (TK) and traditional cultural expressions (TCEs).^{xii} (TK covers known facts about how the world works and can be navigated, such as how certain plants may be made into medicine. TCEs are forms of art, craft, etc.) This comes about precisely because proprietary interests are able to create commercial products using what Indigenous people have already laboured to discover or create. What is *traditional*, after all, is likely to have been around for long enough to fall out of patent or copyright protection and into the public domain.

Since this last point is not true of code placed on GitHub, these specifics of the legal and philosophical debate over TK and TCEs need not detain us. Neither need postcolonial or similarly political commentary, some of which places Indigenous knowledge not only outside Western culture but outside post-Enlightenment conceptions of knowledge and rationality altogether. Conrad Brunk’s writings on ethnobotany and pharmaceutical patents, for example, contend that ‘the scientific explanation of the therapeutic effect’ of certain medicinally useful plants, in converting Indigenous knowledge into scientific knowledge, is what ‘then allows its incorporation into a technology—a “product”’.^{xiii} No such distinction between technological and folkloric knowledge can plausibly be applied to Copilot: nobody could object that *source code* was unfit for legitimate ‘incorporation into a technology’.

Nevertheless, we can discern parallels. A community's collective know-how, owned and controlled by no-one in particular but shared and circulating in a gift economy, is siphoned up and incorporated into a product—or in this case a commercial service. Whereupon some members of the community wonder whether the enterprise behind the service is taking dubious advantage of their largesse, without regard for any consideration that in sharing their code with each other they enjoy shared interests too.

Commons

This raises two questions which ought to be distinguished. (1) Is GitHub's conduct in creating Copilot altogether above board? (2) If not, should there be efforts to seek some form of remedy, whether by any available legal means or through other forms of pressure?

In the rest of this paper I am concerned more directly with the second question; indeed, I think that reflection on it may inform responses to the first. For it is unlikely to have escaped readers' notice that to extend our formal ideas of proprietorship to what is shared amongst a loosely defined collectivity could amount to a considerable expansion of the private sphere. Movements towards any kind of 'cultural intellectual property' for TK and TCEs have historically met just such an objection: that such measures would erode the public domain and set a precedent for further expansion of intellectual property in turn.

In his writing on the intellectual commons, Peter Drahos distinguishes between four conceptions of community.^{xiv} One of his distinctions is between inclusive and exclusive community. The inclusive community means all humanity; an exclusive one has membership criteria. Given that somebody holds proprietary rights to practically any piece of code on GitHub, exclusive community—a pooling of such rights by and largely between their possessors—may initially seem the more natural fit.

Admittedly it is difficult to see how any clear criteria could be established and policed for membership of the community whose wealth is pooled on GitHub. (People who have published code under their GitHub accounts? People who have written code which exists on GitHub? People who have reported bugs to projects hosted on GitHub? People who have copied or downloaded material from GitHub? People who have programmed professionally or taken programming courses? People who understand FizzBuzz? People who compiled and ran a 'Hello world' program once?) But here too the situation is nothing new, since experience with TK and TCEs reminds us that cultures of all kinds have unclear and even contested borders.

Yet before we advance a concept of software developers as some kind of modern guild, upon whose collective interests GitHub's own programmers have infringed, we have cause to wonder how comfortably such an idea might sit alongside that of software freedom. If anything it might gold-plate proprietorial stances towards software, construing it as property both through ordinary copyright and through some additional notion of community interests.

The second distinction is between positive and negative community. Inclusive positive community implies a commons in which everyone in the world has a personal stake, i.e. what is part of the commons is owned in common. Inclusive negative community implies a commons which is unowned, and from which anyone may appropriate items, subject to the condition that the commons as a whole may not itself be appropriated. Exclusive positive community and exclusive negative community form a similar pair but with rights of access to the commons restricted to fewer people in the first place.

Drahos suggests, though rather tentatively, that positive community may create more robust interests in maintaining the commons. And indeed his remarks on the potential risks of negative community may sound very familiar in the context of Copilot:

The dangers of negative community for the intellectual commons come when technology makes new kinds of appropriation possible or when the regulatory conventions protecting it for one reason or another cease to work. The intellectual commons then becomes a hunting ground for the economically strong and the technologically capable.^{xv}

Of course GitHub might well respond that Copilot does not appropriate from anyone else what already exists in the commons: everything that was publicly available remains so, under the same licences. They might—to employ an image with which students of the debates around copyright should be familiar—suggest that they have employed one candle to light another without darkening anyone in the process.

I suspect that if there is a rejoinder to this then it lies in Drahos’s comment on ‘the economically strong and the technologically capable’. Knowledge, including technical knowledge, can be shared like candle flames, but economic strength, where it means strength compared to others’ relative weakness, is more of a zero-sum affair. People who shared their work with the goal of aiding and increasing the numbers of the technologically capable may unsurprisingly prove disenchanted if they find themselves enriching one of the already economically strong as a middleman. This would reflect what Boyle’s mention of ‘largesse’ hints at in the context of TK/TCEs: not ‘this is exclusively ours’ this time, but ‘we assert that this is a gift to mankind, not to a fortunate few’.

Concluding Comments

Software is part of the culture that creates it, but Copilot does not fit neatly into Lawrence Lessig’s distinction between Read/Write and Read/Only culture.^{xvi} It *encourages* adaptation and reuse of code, yet does so in a way that evokes top-down, one-to-many corporate influence as much as it does the gift economies and many-to-many networks of FOSS.

As such it falls within a wider and older debate about communal or collective interests in intellectual creations, and how to respond to the intervention of proprietary interests. TK and TCEs have blazed something of a trail here, and that alone gives reason to note the parallels, but the reverse could now also occur: for if any shift towards asserting collective interests of GitHub’s userbase were to emerge, the case for legal remedies grounded in collective interests of actual ethnicities would receive a boost. The implications for wider developments concerning the scope of intellectual property should therefore be of concern to copyright minimalists. Copyright as we know it is at least time-limited, even if those limits have proved remarkably elastic and in only one direction; but anything resembling a collective right might prove to be another means of enclosing the public domain. We should at the very least be trying not to drift into such an outcome by accident.

- i Brown, Michael F. *Who Owns Native Culture?* 2003, Harvard, pp. 65–6.
- ii Thompson, Clive. *Coders: The Making of a New Tribe and the Remaking of the World*. 2019, Penguin Books.
- iii Brunvand, Erik. ‘The Heroic Hacker: Legends of the Computer Age’. 1996, <https://www.cs.utah.edu/~elb/folklore/afs-paper/afs-paper.html> (retrieved 16th August 2021).
- iv Petricek, Tomas. ‘Cultures of Programming: Understanding the History of Programming Through Controversies and Technical Artifacts’. 2019, <http://tomasp.net/academic/drafts/cultures/> (retrieved 16th August 2021).
- v Stallman, Richard. ‘Why Open Source Misses the Point of Free Software’. 2007–2021, <https://www.gnu.org/philosophy/open-source-misses-the-point.html> (retrieved 16th August 2021).
- vi Himanen, Pekka. *The Hacker Ethic and the Spirit of the Information Age*. 2001, Random House.
- vii Coleman, E. Gabriella. *Coding Freedom: The Ethics and Aesthetics of Hacking*. 2013, Princeton, p.3.
- viii <https://github.com/> <meta> tag description (retrieved 17th August 2021).
- ix Boyle, James. *Shamans, Software, & Spleens: Law and the Construction of the Information Society*. 2004, Harvard, p. 128.
- x Ibid, p. 129.
- xi Free Software Foundation. ‘FSF-funded Call for White Papers on Philosophical and Legal Questions Around Copilot’. 28th July 2021, <https://www.fsf.org/blogs/licensing/fsf-funded-call-for-white-papers-on-philosophical-and-legal-questions-around-copilot> (retrieved 22nd August 2021).
- xii See <https://www.wipo.int/tk/en/> for more information.
- xiii Brunk, Conrad G. ‘Appropriation of Traditional Knowledge: Ethics in the Context of Ethnobiology (Part II)’. In James O. Young & Conrad G. Brunk (eds), *The Ethics of Cultural Appropriation*. 2009, Blackwell, p. 166.
- xiv Drahos, Peter. *A Philosophy of Intellectual Property*. 1996, Dartmouth, pp. 57–69.
- xv Ibid, pp. 66–7.
- xvi Lessig, Lawrence. *Remix: Making Art and Commerce Thrive in the Hybrid Economy*. 2008, Bloomsbury, p. 28ff.