Blurring the Line between Creator and Consumer

By Andrew Katz

We are reaching the end of a great historical experiment. Printing (starting with Gutenberg-style presses, and leading to huge industrial Heidelberg printing machines), radio broadcasting, records (shellac 78s and vinyl), CDs, cinema, television, DVDs and Blu-Rays were the technological backdrop for this experiment. All are (or were) media based on the principle of one-to-many distribution. To understand how this experiment was initiated, and how it is reaching its end, we need to understand a little of the nature of the businesses involved in these activities, and how the law enabled them to attain, and retain, that nature.

The one-to-many broadcast distribution model distorted our perception of creativity. A key characteristic of one-to-many distribution is the role of the gatekeeper: the corporation which decides what we, the public, get to read, hear, watch or listen to. The roles of creator and consumer are starkly defined and contrasted. The public becomes used to the idea of passive consumption, and creativity, in those areas covered by copyright becomes increasingly marginalised: perceived as capable of flourishing only through the patronage of the movie studios, the record companies or the TV stations.

The industrial technology behind printing, broadcasting and vinyl duplication is expensive. Copyright law grants a monopoly which enables the distributors of content to invest in the capital infrastructure required for its packaging and distribution. These are the businesses which grew fat on the monopolies so granted, and they succeeded in convincing the public that it was the corporations' role to provide, and the public's role to pay and consume.

The original social approach to creativity did not become extinct as the dominant producer/consumer mode become established, even for media (like music, for example) where it applied. Andrew Douglas's excellent film *Searching for the Wrong-Eyed Jesus* shows that a visitor to the late 20th century Appalachians of the American South, may well be asked "What instrument do you play?". If the visitor answers "I don't", the questioner will go on to say "Ok, so you must sing".

Steven Johnson, in *Where Good Ideas Come From* makes the convincing case, based on a mass of evidence, that this social mode is more effective at maximising creativity that relying on lone inventors and creators sitting in their garrets and sheds. Lone creators make a good central figure in a compelling narrative (which is one reason why this meme is so popular). However, examining the truth behind the narrative often reveals that any creative work has much broader parentage than the story suggests. James Boyle in *The Public Domain* reveals the story behind the Ray Charles song *I Got a Woman*, tracing it backwards to Gospel roots, and forwards to the YouTube mashup *George Bush Doesn't Care About Black People,* which sprung to prominence in the aftermath of hurricane Katrina. To be sure, companies sometimes tried to foster a social model *within* the organisation, but as Johnson points out, the benefits of social creation increase very dramatically with the size of the pool of participants, due to network effects (Metcalfe's law – the number of connections increases with the square of the number of participants): until company silos are able to combine, the beneficial effects are relatively small.

The internet proved hugely disruptive. The sharing and social nature of Web 2.0 has enabled the rediscovery of the natural, human, social mode of creative endeavour. The social side of the internet is dominated by individuals acting in their private capacity, outside the scope of businesses. Businesses, initially wary of losing control over the activities of their staff, and which regarded internet social activities as, at best, time wasting, and, at worst, providing the potential to leak the company's "valuable intellectual property", were often slow to see the benefits of social interaction in terms of benefits to their creativity. However, as they have seen the benefits accrue to their competitors, they are starting to embrace a more open mode of business.

A return to the social mode is not without setbacks. The internet radically lowered the barrier to entry for collaborative participation, and consequently increased the number of potential contacts that an entrant can make. This immensely powerful engine of creativity is also subject to a brake: the effect of unfit-for-purpose copyright laws.

The copyright laws of the broadcast era do more to assist the incumbent gatekeepers (the film companies, music companies and so-on) than to promote the social mode of collaboration. A side-effect of the digital world is that almost every form of digital interaction involves copying of some sort. Whereas copyright law has nothing to say about sharing a book with a friend by lending it to her, in the digital realm, lending her a digital copy of *Nineteen Eighty-Four* to read on her e-book reader or computer involves and of copying which can potentially violate copyright law.

The broadcast-model gatekeepers have relied on this unintended side-effect of copyright law to their advantage, taking action against private individuals who had no intention of monetary gain, including mash-up artists, home video enthusiasts and slash fiction authors.

Incumbent rights-holders, fearful of losing their profitable monopoly-based businesses, have sought to extend their rights ever further, by (frequently successfully lobbying) governments to legislate for new and increased intellectual property rights, far beyond their original purpose and intention.

To put the issue in context, it is necessary to ask the fundamental question: "What is copyright for?"

Thomas Jefferson was one of the most lucid writers on the topic. He understood well the unique nature of knowledge:

> *"If nature has made any one thing less susceptible than all others of exclusive property, it is the action of the thinking power called an idea, which an individual may exclusively possess as long as he keeps it to himself; but the moment it is divulged, it forces itself into the possession of every one, and the receiver cannot dispossess himself of it. Its peculiar character, too, is that no one possesses the less, because every other possesses the whole of it. He who receives an idea from me, receives inſtruction himself without lessening mine; as he who lights his taper at mine, receives light without darkening me. That ideas should freely spread from one to another over the globe, for the moral and mutual inſtruction of man, and improvement of his condition, seems to have been peculiarly and benevolently designed by nature, when she made them, like fire, expansible over all space, without lessening their density in any point, and like the air in which we breathe, move, and have our physical being, incapable of confinement or exclusive appropriation. Inventions then cannot, in nature, be a subject of property."*

But Jefferson conceded that creative people should be given a limited right of exclusive control over their creations. Even though a monopoly is inherently a bad thing (as was recognised in the late 18th century just as it is today), nonetheless, a monopoly of control, in the form of copyright or a patent, was the most convenient way of enabling the creators to be remunerated for their work. And once the monopoly expired the, idea would be freely available for all and become part of the common heritage of mankind, to be used without restriction by anyone. The necessary (but limited) monopoly includes “copyright”, and the principle that the restrictions should be the minimum possible to achieve that aim should be copyright's golden rule.

The golden rule has been repeatedly ignored. The scope of protection has steadily increased over the last three hundred years, to the extent that, in Europe, the protection granted to the author of a novel, for example, lasts for seventy years after his or her death.

What isn't restricted by intellectual property is called “the public domain”.

Commentators have become increasingly strident that the public domain is a public good (an idea with which Jefferson would have agreed). In the same way that a common land is open to all to graze their animals, the public domain has been described as a commons of knowledge, where potentially anyone can graze on the intellectual creations of others. The public domain has one crucial difference from a commons in the tangible world: a meadow open to all can easily be over-grazed and ruined, so that it becomes of use to no one (the so-called “tragedy of the commons”). It is impossible to exhaust the commons of knowledge and ideas.

The modern “tragedy of the commons” is that, just as the internet makes it easier to pass ideas and knowledge from one person to another (for “the moral and mutual instruction of man, and improvement of his condition”) it seems that legislation and the more extreme activities of the rights holders are making it more difficult for those ideas and knowledge to enter the commons in the first place. This is because the term of intellectual property is forever extending (will the early Mickey Mouse films ever enter the public domain?), and so is its scope (for example, the patenting of genes or plants).

Increasingly, people are becoming aware of the value of the commons, and are seeking to protect it. At the same time, we are becoming aware that the monopoly granted by intellectual property laws is a blunt instrument, and that people are prepared to create for reasons other than the expectation of payment for the use of their creation.

Copyright law does not always have to work against the commons.

In the late 1980s, Richard Stallman, a computer programmer, realised that copyright law could be turned inside out to create a commons of computer software. The method he proposed was simple, but brilliant:

Software is protected by copyright. The existing software business model involved granting customers permission (the licence) to use a specific piece of software. This licence was conditional on the customer not only paying the software publisher fee, but also adhering to a number of other restrictions (such as only using the software on one computer, for example). Why not, Stallman reasoned, make it a condition of the licence, that if you took his software and passed it on (which he was happy for people to do), then they had to pass it on, together with any changes they made, under the same licence? He called this sort of software “free software”: once a piece of software has been released under this sort of licence, it can be passed on freely to other people, with only one restriction: that if they pass it on, in turn, they must also ensure the people they pass it on in a way that guarantees and honours that freedom.

In time, he reasoned, more and more software would be released under this licence, and a commons of freely available software would flourish.

In the two decades since the most widely used version of the licence (called the GNU General Public License version 2 – called the GPL) was issued it has become the most commonly used software licence. It is the licence at the core of Linux, the computer operating system which powers Google. Amazon, Facebook and which enabled Red Hat to generate revenue in excess of $1.5Bn in financial year 2013-14.

The GPL software commons not only exists: by any measure it is an overwhelming success, whether it is in terms of number of participants creating software for it; whether it is the number of items of GPL software in use; whether it is in the environments in which GPL software can be found (from running over 90% of the worlds 100 most powerful computers, to mobile phones and in-car entertainment systems); whether it is at the core of the business offerings of businesses like IBM and Red Hat.

The success of free software is not just down to the GPL. The GPL extracts a price for using the commons. To analogise possibly too far, a landowner adjoining the GPL commons who wants to use it also has to add his own land to the commons (although, remember that this is the magical land of ideas which cannot be ruined by over-grazing). This will have the effect of increasing the size of the commons as more and more adjoining landowners want to make use of the commons and donate their own land in the process. But many of them may not want to join this scheme: either because they don't want to add their own land to the commons, or because they have already pledged their land to another commons.

Is it possible to generate a commons of ideas without forcing participants to pay the price of entry: that they add their own adjoining land to the commons? Is the compulsion of the GPL necessary, or is the social and community dynamic powerful enough to allow a similar commons of ideas to spring up on its own?

The software industry has given us several outstanding examples of this. Apache, the most popular web server software in the world (and used by many of the busiest web sites) is issued under a licence which doesn't ask users to pay the GPL price. Anyone can take the Apache code, and modify it and combine it with their other software, and release it without having to release any sources to anyone else. In contrast to the GPL, there is no compulsion to add your software to the Apache commons if you build on Apache software and distribute your developments, but many people choose to contribute back without this compulsion. FreeBSD, to take another example, is an operating system with similarities to GNU/Linux which is licensed under a very liberal licence allowing its use, amendment and distribution without the *requirement* to contribute improvements back: nonetheless, many people *choose* to do so.

The GPL tackles an issue called the *free rider* problem. Because BSD does not compel people to contribute back to the commons, those who take advantage without contributing back are called *free riders*. The question is whether free riders really are a problem (as the GPL band would maintain), or they are (as the BSD band would maintain) at worst a cost-free irritant, or at best, a cadre of people who will eventually see the light and start to contribute, once they recognise the benefits.

Supporters of both the GPL and BSD models of licensing have similar aims: the production of a software commons which will enable the social mode of creativity to flourish.

While the BSD model could subsist in the absence of copyright, GPL relies (perhaps ironically) on copyright law to enforce its compulsion to share. It still remains an open question as to whether the better model is to use licensing to compel people to participate in the software commons, and reduce the free rider problem (as with GPL), or whether voluntary engagement will result in a more active community (as with Apache). As we will see below, designers working outside the digital domain will rarely have the ability to choose a GPL-style option.

The undoubted success of free and open source software (Gartner confidently states that all businesses today use at least some free software in their systems. The Linux Foundation estimated that free software underpinned a $50 billion economy in 2011) means that this model has been considered for its applicability in other contexts. Can designers in other fields benefit from this model?

One of the most prominent of these has been the Creative Commons movement. Founded in 2001, Creative Commons has written a suite of licences which were inspired by the GNU GPL, but which are intended for use in relation to a broad range of media, including music, literature, images and movies. The licences are drafted to be simple to understand and are modular, in that the rights owner can choose from a selection of options. The *attribution* option requires that anyone making use of the work makes fair attribution to the author; the *share alike* option is akin to the GPL, in that if a licensee takes the work and redistributes it (whether amended or not), then the redistribution needs to be on the same form of licence; the *no derivatives* option means that work may be passed on freely, but not modified, and the *non commercial* option means that the work can only be used and distributed in a non-commercial context.

There are now millions of different works available under on of the various creative commons licences: Flickr is just one search engine which has enabled Creative Commons licensing as a search option. There are, at the time of writing (2014), over 300,000,000 Creative Commons licensed images available for use on Flickr alone (up from just under 200,000,000 in 2010, when the first version of this article was written). Similar sites provide music and literary works under a Creative Commons licence.

The Creative Commons provides designers and other creatives operating within the digital domain the legal infrastructure to adopt this model. There is also an effective choice as to whether an appropriate model is GPL-style share-alike, or BSD style. Where designers' work moves into the physical world, all is not so straightforward.

The movement of hardware design into the commons has been difficult. The issues are fundamentally:

1. In the digital world, the creator has the choice of whether a GPL- or BSD- model is appropriate. This choice does not translate well to the analogue world.
2. Digital works are relatively easy to create and test on low-cost equipment. Analogue works are more difficult to create, test and copy, and this creates barrier-to-entry problems.
3. Digital goods are easy to transport. Analogue ones frequently aren't. This creates a barrier to the communication necessary to get the most benefit out of network effects.

The barrier to entry for any participant in a digital project is remarkably low. A low-cost computer and basic internet access are all that is required to have a system capable of running the (free) GNU/Linux operating system and accessing project hosting sites like github.com (which is free of charge to public projects).

A vast range of tools required to develop software (such as GCC – the GNU Compiler Collection) is also available as free software. Copying purely digital works is trivially easy. Physical objects are a different matter.

Hardware development is likely to require more intensive investment in equipment (including premises), not just for development, but for testing. Electronic digital hardware is probably closest to software in terms of low barrier to entry: for example, the open-source Arduino micro-controller project enables an experimenter to get started with as little as $30 for a basic USB controller board (or less, if the experimenter is prepared to build the board). Arduino's schematics, board layouts and prototyping software are all open source. However, Arduino-like projects represent the lowest barrier to entry in the hardware world.

The Arduino-style project is essentially a hybrid of the analogue and the digital domains. Prototyping software makes it possible to develop Arduino-based hardware in the digital domain, where it retains all the characteristics of the digital world: ease of copying, the ability to upload prototypes to fellow-contributors for commentary, assistance and showing off. These are characteristics which enable network effects, and which make the open source model so powerful. It is only when the project is implemented as a physical circuit board that these characteristics are lost.

The analogue world is not always so simple. One of the most ambitious open source projects is the 40 Fires/Riversimple hydrogen car project, which has developed a small urban car (the Hyrban) powered by hydrogen, using a fuel cell/electric drivetrain. Elements of the design (for example power control software, or the dashboard user interface) can be developed largely in the digital domain, but the development of motors, brakes, the body shell and so-on are strictly analogue only, and not only present a large barrier-to-entry for interested tinkerers, but also tend to restrict the ability to participate in the development community: a necessity if network effects are to work. It is, clearly, difficult to upload a car to a development site and say "can you tell me why the windscreen leaks?".

Another significant issue is the lack of design software at a low cost. Software developers have access to high quality tools like development environments and tools available for free under free software licences. There is no similar suite of CAD software, and proprietary CAD software is notoriously expensive. The barrier to entry is raised once again.

Many of these issues are surmountable, in time. Ever-improving simulation software means that more and more testing and prototyping can be undertaken in the digital domain. The introduction of 3-d printers, such as the RepRap, means that printing physical objects, such as gears, for example, out a variety of plastics, is becoming increasingly affordable and feasible. The lack of suitable CAD software is being addressed by a number of projects.

For designers, progress in open source tools, increased connectivity and so-on makes the establishment of open-source communities ever more feasible. The legal issues are, however, not so straightforward.

So far, we have concentrated on copyright issues. Other forms of intellectual property pose, in some ways, greater challenges. Copyright protects the expression of an idea. Retaining the same idea, but recasting the expression of it in a different form does not infringe the copyright. The story of two people from warring tribes meeting, falling and love, and dying in tragic circumstances can be told in a myriad different ways, each with their own independent copyright, and without infringing anyone else's copyright. This has two practical consequences. The first is that if a creator creates something which he or she has not copied from something else, then the creator will not be in breach of copyright, even if their creation turns out to be very similar, or even identical, to someone else's. The second is that if a component of something is found to be infringing copyright, it is possible to rewrite it by recasting the same idea in a different expression.

Copyright also has the advantage of being (reasonably well) harmonised worldwide, and has also proved amenable to hacking (by Richard Stallman) so that it can be used to guarantee freedom in the code it covers.

There are other forms of intellectual property protection, and, for designers, these are more problematic.

This issue is linked to the distinction between the analogue and digital domains. Designs will almost invariably start with some sort of drawing or description. This will be protected as a literary or artistic work by copyright. Often, this material will be digital in nature. At this point, it's similar to software. Licensing options include the suite of Creative Commons licences. Once an item is created in the physical world, a different set of legal considerations applies.

The most obvious is design right. Unfortunately, design right is complex and uncoordinated. There are many different types of design right. In the UK, for example, there are four separate design right regimes operating simultaneously, covering (depending on the right in question) aspects such as shape, texture, colour, materials used, contours and ornamentation. Registered designs are in many ways similar to patents (and are sometimes called petty patents or design patents). Infringement can be unintentional, and independent creation is irrelevant. Unregistered designs are more in the nature of copyrights, and are capable of infringement only where copying has taken place. The very fact that registration is required (in the case of registered design rights, clearly), itself provides a barrier to entry for collaborative projects: whereas copyright arises automatically and without the necessity of registration, who will pay for the preparation of a design registration, and who will make the application and maintain it?

Patents provide a particular problem for both programmers and designers, as they can impinge on both the digital realm and the analogue realm. Patents are a protection on the idea itself. However that idea is expressed, the patent will be infringed. Independent invention does not excuse patent infringement. The only way to be sure that an invention does not infringe a patent is to do an exhaustive check in patent offices worldwide: something that is very rarely done (the expense is enormous and creates a vast barrier to entry for small businesses, and US law in particular applies a positive disincentive to search: if a search is undertaken, then the searcher can be deemed to have knowingly infringed a patent – even if their reasonable determination was that the patent was not infringed – and will be liable to triple damages as a consequence).

Pressure groups are lobbying worldwide for a reform of the patent system and process, but at present it is clear that the system benefits incumbent large companies with an existing patent portfolio.

The upshot of the intellectual property issues is that whereas those operating wholly in the digital domain (which includes programmers, but which can also extend to digital creatives such as filmmakers, novelists or graphic designers) have the ability to choose whether they prefer the GPL model to the BSD model, for a number of reasons, the BSD model is often a more viable option in the hardware, analogue world. The main reasons are, briefly, that (1) copyright, being largely universal, automatic, unregistered and long-lasting, is better suited to the development of a copyleft model that other forms of intellectual property; and (2) that the difference in cost between copying something and reverse engineering (which is vast in digital world, but much smaller in the analogue world), makes the copyleft a less compelling point).

These reasons probably need some explanation.

For a GPL-model to apply to hardware designs, to be effective, it would need to impinge on the ideas underlying the design (meaning patents), or on the visual characteristics of the design (meaning design rights). A GPL-style model based on patents would likely fail (at least when any of the participants are not large corporations) because of the cost, complexity, and time involved in applying for the patents (and the necessity to keep the invention secret prior to its publication as part of the application process squares badly with the open source ethos). If the model were based on design rights, this would fail in relation to registered design rights, for the same reasons as for patents, and for unregistered design rights would be unlikely to work because the scope and length of protection would be too short, and because the rights are insufficiently universal (although there is some scope for a limited GPL-style model in relation to unregistered design rights).

There is also an economic argument why a GPL model may cause problems in the field of hardware. The reasoning is as follows: the digital world makes things extremely easy to copy. Imagine a programmer wants to create some software based on a program with similar functionality to a word processor, for example, released under the GPL. The options are (1) take the original GPL program, modify it, and release the result under the GPL; or (2) take the GPL program, reverse engineer it, and rewrite a whole new program from scratch, which will be unencumbered by copyright restrictions. The difference in the amount of work involved in the two scenarios is vast, and any programmer is likely to consider very seriously adopting the easier, cheaper and quicker option (1), where the “cost” is out-licensing under the GPL. However, even if there were a functioning mechanism for applying share-alike to, for example, a mechanical assembly, in this case, an engineer wishing to reproduce the mechanical assembly would, in effect, have to reverse engineer it, in order set up the equipment needed to reproduce it. Copying a digital artefact is as simple as typing:

```
cp old.one new.one
```

Copying an analogue artefact is vastly more difficult, and therefore there is little difference between slavish copying (which would invoke GPL-like restrictions), or reverse-engineering and re-manufacturing (which wouldn’t). In this case, it’s much more likely that the “cost” of GPL-like compliance would be greater than the benefits of having a GPL-free object.

So even if GPL-like licences are legally effective in the physical world, economics would tend to disfavour their use.

Designers, therefore, operating in the analogue realm may choose an openness model more akin to BSD that to GPL. Their challenge is to make this model work, and discourage free riders with a combination of moral pressure and a demonstration that playing by the community norms will be beneficial both to them, and to the community as a whole.

Designers and creators are increasingly able to benefit from the promise of the connected, social mode of creativity. The way was paved by free software pioneers, who skilfully hacked the copyright system to generate a commons which has not only, generated a huge global business, but also provided the software which runs devices from mobile phones through to the most powerful supercomputers. It provides the software which gives the developing world access to education, medical information, micro finance loans and enables them to participate in the knowledge economy on similar terms to the developed nations.

The challenge for designers and creators in other fields is to adapt the model of software development to their own field of work, and to counter the extensive efforts of incumbent beneficiaries of the broadcast era to use ever more draconian legislation to prop up the outmoded business models. Ultimately, the social mode will win: it takes one of humanity's defining characteristics, the fact that as animals we are highly social and community oriented, and uses it as the foundation of the entire structure. One-to-many works against this fundamental trait. Nature will ultimately triumph.

Open Design: Definition

An open design is:

0. The freedom to use the design, including making items based on it, for any purpose (freedom 0).

1. The freedom to study how the design works, and change it to make it do what you wish (freedom 1). Access to the underlying design documents is a precondition for this.

2. The freedom to redistribute copies of the design so you can help your neighbour (freedom 2).

3. The freedom to distribute copies of your modified versions of the design to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the underlying design documents is a precondition for this.

(A version of the Free Software Foundation's "four freedoms" repurposed for designs by OHANDA – the Open Source Hardware and Design Alliance).

Rights And Licensing Schemes

The re-use of designs is governed mainly by copyright, design rights and patents. Traditional open licensing schemes have been based on copyright, as this is the main intellectual property right which impinges on software, the most fertile ground for openness.

Software licensing schemes include the GPL (which enforces copyleft) and BSD (which doesn't). A more comprehensive list of licences can be found at

http://www.opensource.org/licenses/index.html.

Software licences rarely work properly when applied to other works. For literary, graphic and musical works, the creative commons suite is more effective: www.creativecommons.org. They allow both copyleft (*share alike*) and non-copyleft options. They may work well when applied to underlying design documents, which are covered by copyright, and control the distribution of those documents, and the creation of physical objects from them, but their protection is unlikely to extend (depending on the jurisdiction) to the copying the physical object itself.

Creative Commons And Design Rights

Creative Commons licensing is fundamentally based on copyright, and there is little clarity or consensus on how they would operate in relation to design rights across the myriad different jurisdictions and types of right. Those designers operating purely in the realm of copyright will find that there is already an existing structure of support in terms of Creative Commons licences and associated communities. Where other forms of intellectual property impinge, the world is less developed. The Creative Commons licences are arguably sufficiently broadly drafted to cover unregistered design in certain circumstances, but because they were not drafted with design rights in mind, it cannot be assumed that the copying of a three-dimensional object will automatically fall within its scope.

Patents are specifically excluded from Creative Commons licenses: this, a designer can appear to be offering a design on an open basis using a CC license, but still withhold patents rights necessary for its manufacture or sale.

Andrew Katz has practised technology law for 20 years, and has been at Moorcrofts in the UK's Thames Valley for 14 of those. He is a Fellow of the Open Forum Academy and Free Software Foundation Europe and visiting lecturer on Free and Open Source Software at Queen Mary, University of London. He has lectured on open issues in London, Paris, New York, Boston, Seoul, Helsinki, Stockholm, Mangalia (Romania), Brussels, Amsterdam, Barcelona, Edinburgh, Oxford and Cambridge. He is on the core drafting team of the CERN Open Hardware licence. He acts for some of the world's leading free and open source software companies and projects. Andrew has more recently been involved in the rapidly expanding fields of open data and open hardware, and as well as speaking on legal issues at the Open Hardware Summit in New York in 2012, he co-opened the 2014 Open Hardware and Data Conference in Barcelona (the world's first legal conference specialising in open hardware and data).

Andrew is married with two children and lives in Oxfordshire. He's heavily involved in both the Oxford tech and music scenes and is passionate about live music: the grottier the venue, the better. He still codes occasionally, mainly Javascript and Python, although his favourite computer language will always be Pascal.