Norms and the Law

Edited by

JOHN N. DROBAK
Washington University School of Law

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press
32 Avenue of the Americas, New York, NY 10013-2473, USA

www.cambridge.org
Information on this title: www.cambridge.org/9780521862257

First published 2006

Printed in the United States of America

A catalog record for this publication is available from the British Library.

Library of Congress Cataloging in Publication Data

Norms and the law / edited by John N. Drobak.
p. cm.
Includes bibliographical references and index.
ISBN-13: 978-0-521-86225-7 (hardback)
ISBN-10: 0-521-86225-6 (hardback)
ISBN-13: 978-0-521-68079-0 (pbk.)
ISBN-10: 0-521-68079-4 (pbk.)
1. Law – Social aspects. 2. Social norms. I. Drobak, John N. II. Title.
K370.N67 2006
340'.115 – dc22 2006001220

ISBN-13 978-0-521-86225-7 hardback
ISBN-10 0-521-86225-6 hardback

ISBN-13 978-0-521-68079-0 paperback
ISBN-10 0-521-68079-4 paperback

Contents

in fact is sometimes employed in contexts where domestic decision-makers number in the many dozens – for example, in a traditional village commons or a contemporary co-housing development.[112] Unalloyed majority rule among either owners or occupants, if it is ever employed, is to be expected only in the least intimate of households.

IV. CONCLUSION: TOWARD MORE HOMEWORK ON HOW THE HOME WORKS

This chapter has presented a number of potentially falsifiable propositions about the norms of households, among them that contributors of equity capital to households are empowered to make residual control decisions; that members of households rely more on gift exchange than on explicit bilateral exchange; that written contracts among housemates are rare; that privatization of spaces is common within households; and that the owners (or occupants) of an intimate household prefer to make decisions by consensus, as opposed to majority vote. More systematic empirical work on these issues and others would help reveal fundamental insights about how people cooperate (or fail to cooperate) in managing jointly owned and occupied resources. Heretofore, scholars who have investigated the commons typically have focused on fisheries, pastures, and irrigation networks owned by clusters of households. While this work plainly is valuable, scholars can learn much by lowering their sights to a smaller and more basic human institution – the household itself.

[112] See Margaret A. McKean, *Success on the Commons: Comparative Examination of Institutions for Common Property Resource Management*, 4 J. THEORETICAL POL. 247, 260–61 (1992); Fenster, *supra* note 97, at 29–30, 34–35, 43–44 (describing decision-making procedures at three co-housing projects).

5 .commons

Lawrence Lessig

I start with the words of someone famous, and then an account of the deeds of someone not quite so famous, as a way of framing an argument about the commons in cyberspace.

First the words.

In a letter written late in his life, Thomas Jefferson, the first commissioner of the patent office, commenting about the limited scope of patents, had this to say about the very idea of protecting something like an idea:

> 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 everyone, and the receiver cannot dispossess himself of it. Its peculiar character, too, is that no one possesses the less, because every other possess the whole of it. He who receives an idea from me, receives instruction himself without lessening mine; as he who lites 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 at 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.[1]

Patent law could try, Jefferson said, to make property out of inventions; but the idea was against nature. Nature had conspired to make ideas, in the words

[1] Letter from Thomas Jefferson to Isaac McPherson (Aug. 13, 1813), *in* 6 THE WRITINGS OF THOMAS JEFFERSON 175, 180 (H.A. Washington ed., 1861).

of the economist, nonrivalous and non-excludable – and nature in the end, he believed, would win.

Those are the words from someone famous. Here's a story less famous.

There's a man whom some of you will have heard of but no doubt most of you will not, who in 1984 began a movement that will, I predict, be understood some day to define the ideals of a generation. This movement is the free software movement; its founder is a MacArthur genius named Richard Stallman. And in 1984, Richard Stallman began a project to build a "free version" of something called the Unix operating system. That free version of Unix was not to be called "Unix"; it was to be called GNU – "Gnu's not Unix." This free version of Unix was to be made available to all to build upon and use as they saw fit.

What's the significance of an operating system?

Computers are boxes of chips and wires; operating systems ("OS") make them run. They make them run not in a physical sense – electricity does that. Rather, they make them run in a logical sense. Operating systems provide the basic platform that enables computers to function; they are a complex of computer software – ordinarily software – that links programs to the machines they run upon. An operating system makes it simple, for example, for a program to display characters on a screen. Without the operating system, every programmer who wanted to display characters on a screen would have to write code to display characters on a screen. Obviously, there are a lot of programmers, and if all programmers had to write that code, that would be a lot of redundant code. So an OS helps eliminate that redundancy by providing a common set of code that others can call upon. Programmers writing code to run on a particular OS platform know about this code; they simply invoke it to achieve the ends they seek. In this sense, an OS is a language for those writing for a particular computer platform; its conventions replace a lot of work that others would have to do.

Unix was the breakthrough OS in the history of computing. Before Unix, an OS was something written for particular machines. IBM built computers; it built an OS that would run on its machines, but this OS would not run on anyone else's machines. If you wrote programs for this OS, then you were writing programs for IBM machines.

Many didn't like this inherent, technological tie linking programs to particular machines. Many wanted to write programs that might run on a number of different machines. And, in particular, companies that didn't want to be linked to a particular computer company wanted the freedom to write programs that would run on more than one machine.

AT&T was such a company. At the time computers were taking off, AT&T was tied up. Regulations forbade it from becoming a computer company. It

was a telecommunications company – a state sanctioned monopoly to boot – so regulators were eager to keep its power out of the nascent computer market. So AT&T was forbidden to sell computers or the software to run computers. But it nonetheless needed software to run computers, and it wanted to write the same software to run many computers. So its engineers gave birth to this wonder called Unix – an OS that would cross many computer platforms.

But here was the crucial catch: because Unix couldn't be sold, these engineers were able to convince AT&T that it wouldn't hurt for AT&T to give Unix away. More importantly, these engineers were able to convince AT&T that this OS should be given away *with* its source. That is, not only should people be able to use Unix for free, they should be able to modify and adapt Unix freely. And they should be able to do this by having not just a program called Unix, but also the "source code" that made Unix possible. In other words, they urged AT&T to give away not just an OS, but the code that made the OS run.

If there is list of "most significant decisions" in the history of computing, this decision by a few engineers at AT&T to give the Unix source away is high on that list, for it made real a phenomenon that we may not have otherwise noticed. Very quickly Unix spread to become the foundational operating system for computing in the world: universities adopted it, computer science departments taught it, and millions tinkered with it. A generation of engineers was raised on it. Unix became the language within which computing was understood. And it became all of this because people were free to open, and tinker with, the code. Its source code was given away; people were free to take that source apart; and because they could take that source apart, they could come to understand – they could learn – how it, and computers functioned.

And so here is the last technical idea that is crucial to understanding the argument that I am making: source code. Computers run programs. Programs are code. For code to run, the computer must be able to read the code; for them to be written, humans must be able to read the code. It is in the nature of computers and humans that we can't both read the same thing well. What we can read efficiently is too cumbersome and confused for computers; what computers read efficiently is too complex and arcane for humans to understand. So code gets written first in a language that humans can understand – that's called source code. It is then translated into a language that computers can understand – that's called object code. And then this object code is loaded onto a computer, executed by the machine.

AT&T released not just the object code, it also released the source code. By releasing the source code, more than machines could understand it. By inviting more than machines to understand it, many millions came to understand it, and it became a standard upon which an industry was built.

In 1984, all that changed. In 1984, AT&T was broken up; as a consolation, the computing-related limitations the law had placed on AT&T were removed. AT&T was free to sell computers, and sell software. AT&T thus decided that's exactly what it would do. And so in 1984, AT&T announced that no longer would Unix be free. No longer would its source code be available to any one at all. Any access would have to be licensed; and every license would demand a fee. The enclosure movement came to this fundamental concept in computer science and produced something of a shock. A generation was writing its code for the language of Unix; a company now claimed rights to control the terms of this language.

Richard Stallman was, like many others, angry about this event. He was angry that so much creative energy had been devoted to a platform that now restricted rights to its expression. And so in 1985, Stallman created the Free Software Foundation to espouse a new philosophy in computer science, a philosophy that he believed expressed the implicit beliefs of those who had lived for many years in the land of Unix. This philosophy he expressed in the ideals of the free software movement.[2] Code, Stallman said, should be free: people should be free to share it. Coders should be free to take apart the code built by others, to understand it, to modify it, to improve it.

As I report this idea of "free software," many of you will immediately jump to a reading of these words that is wrong, but usefully wrong. You will hear Stallman saying that software should be gratis; that people should be allowed to take it at no cost; that coders should be permitted to get the resource from others without any compensation to the others. But this is not what Stallman means, even if this mistake may well be squarely intended by his language. For by leading you to this mistaken understanding of his words, Stallman has the chance to correct you. Free, as he says, not in the sense of free beer, but free in the sense of free speech. Software should be free in just the same way that speech is free; even if, as with some speech, to get it you may have to pay a price.

What could such a distinction mean?

If there's no such thing as a free lunch, then there's no such thing as free code. People write code; these people have to eat lunch; so somehow these code writers need money to pay for lunch. Somehow, this code writing has to pay.

[2] *See* LAWRENCE LESSIG, THE FUTURE OF IDEAS: THE FATE OF THE COMMONS IN A CONNECTED WORLD 52–54 (2001); PETER WAYNER, FREE FOR ALL: HOW LINUX AND THE FREE SOFTWARE MOVEMENT UNDERCUT THE HIGH-TECH TITANS 9, 34–36, 67, 68 (2000).

Even Stallman gets this, and so there's nothing inconsistent with coders getting paid, and with them selling their code. But what the Free Software Foundation did was support the building of a free version of Unix that could be sold by any distributor; but a version that must carry with it its source code. And by carrying its source code, that meant others were always free to use this software critically. As free speech protects the right to question, others could build upon and, at the same time, question this free Unix; they could modify the source, tinker with the source, that right was always reserved to them. But what they built on this source, or what they did to this source, they were free to sell, and did – so long as they did not bottle up what they originally received for free. For the license that one receives with this free software restricts the rights of the recipient in one important way – however free this software is, one is not free to remove the initial source.

It wasn't Stallman in the end, however, who built this free version of Unix. Like Moses and the promised land, Stallman got just to the edge of completing a free software Unix when a crippling carpal tunnel syndrome made it impossible for him to type. The project, though supported by many, quickly stalled. It took another sort of genius, Linus Torvalds, to supply the missing part to the puzzle – a kernel that plugged into the parts of the operating system Stallman had built. And as ginger to a ginger cake, this tiny part of the overall system then defined the name for the system as a whole. The Linux operating system, as most of you know it – or the GNU/Linux operating system, as those who want to be reminded of the past will call it – now flows freely across the world as a free and open version of Unix. At any moment, from any number of places, one is free to download the source of the Linux OS; but at any moment, from any number of different vendors, one also can buy compiled and functioning code called the Linux operating system. The only rule that governs that code is the rule that governs all free code, and the rule that governed Unix itself before AT&T was free to change the rule – that the source code is delivered as well as the object code, and that the source code must never be locked up. You can change GNU/Linux as you want; you can sell the changes; but you must include the source to the changes you've made.

This is a long introduction to a definition, I realize. But this story establishes all that my definition of a commons will need. For my claim is that the GNU/Linux operating system is a commons in all the ways that a commons is relevant in cyberspace. And my hope is that, with this account vivid in the background, we can see the relevance of these separate elements.

In my view, we should understand a commons as a resource that others may draw upon – as a first cut we could say, without requiring the permission of

someone else, or, as I want to refine that idea, without requiring the permission of a *certain kind* of someone else. But let's start with the first cut first.

A commons is a resource that others may draw upon without the permission of someone else. A park, in this sense, is a commons. I am permitted to enter the park, to enjoy its peace, without getting the permission of someone else. An idea in the public domain is a commons in this sense. I need not register with anyone to take and use the theory of relativity. A speech by a member of Congress is a commons in this sense. I don't need a politician's permission to use his speech and ridicule it. The speech is free for anyone to take and do with as he or she wants. A report of the Supreme Court deciding a certain case is a commons in just this sense. One need not ask the permission of a court to cite the holding of a case as authority. It exists out there for anyone to use; it is free of the control of an author or institution. A public road is a commons in just this sense. I can get in my car in San Francisco and drive to my office in Palo Alto without registering with the local authorities, without requesting the permission of any central dispatcher. And finally Unix (before 1984) and GNU/Linux now is a commons in just this sense. I can, if I can, take a version of GNU/Linux and change it; I can take the OS and explore and modify it; I am free to use it to build something different; I am free to use it to criticize it, or ridicule it. I can do this without the permission of anyone else because the essential element that gives me such power – its source – is there for me to use.

I said this definition was a first cut, and so let me now add an important qualification: It may well be, in all these cases, that in fact to use a commons in the sense I would call a commons, I do need the permission of someone else. It may well be that I must ask before I can take. But if that is so, then the resource can still be considered a commons if the reasons for denying access are of a special sort: if they are, as First Amendment lawyers would describe them, *content neutral*. If the reasons are neutral with respect to the particular use, nonetheless they may restrict access for certain reasons. So, for example, a park is still a commons even if it is closed in the evening, or if a nominal admission is charged. Those are restrictions, no doubt; they in a sense require permission. But these restrictions are unrelated to one's ideas, or purpose. I am not rejected if I am a Republican, or if my purpose is to study in the park rather than relax. So too could a toll road still be considered a commons, so long as the toll was neutral, and relatively low. So long as access was not contingent upon holding certain views, or upon membership in the party. And so too could GNU/Linux still be a commons, even if one must agree to a license to get it. The condition is neutral; it restricts, but not to a particular end. It is not picking sides in a debate, or winners in an argument.

In this sense of a commons, my claim is this: that the most important features that of cyberspace so far – the features that more than any other explain its extraordinary growth, and the extraordinary innovation that cyberspace has produced – are the features that most resemble a commons. That without a commons in this sense – without a number of distinct yet overlapping structures that all satisfy the conditions of the definition I have offered – there would have been nothing remotely like the Internet for us to rave about. That all that is unexpected and extraordinary about this space are the parts of this space that are commons.

We can be more specific, however, about the place of the commons within the structure we call the Internet. We can distinguish three "layers" that comprise a communication system. As NYU Professor Yochai Benkler describes it, at the bottom we could describe a physical layer – the wires that connect the phones or the computers or the cable across which television might be broadcast. Above that, the logical layer – the system that controls who gets access to what or what gets to run where. And above that, the content layer – the stuff that gets said or written within any given system of communication.[3]

Each of these layers in principle could be controlled or free. They would be "free" if they were organized in a commons – organized so that anyone could get access on equal terms, whether they had to pay (a fixed and neutral charge) or not. They would be controlled if they were the property of someone else – someone who had a right to exclude, or to grant access or not, based on his or her own subjective reasons.

And the communications system built differs depending on whether these layers are free or are controlled.

Consider four possibilities as we vary whether each of these layers is owned or free.

Speakers' Corner: Orators and loons gather every Sunday in Hyde Park's Speakers' Corner to rage about something or nothing at all. It has become a London tradition. It is a communication system organized in a specific way. The physical layer of this communication system (the park) is a commons; the logical layer (the language used) is also a commons. And the content layer (what these nuts say) is their own creation. It too is unowned. All three layers in this context are free; no one can exercise control over the kinds of communications that might happen here.

Madison Square Garden: Madison Square Garden is another place that people give speeches. But Madison Square Garden is owned. Only those who

[3] Yochai Benkler, *From Consumers to Users: Shifting the Deeper Structures of Regulation*, 52 FED. COMM. L.J. 561, 562–63 (2000).

pay get to use the auditorium; and the Garden is not obligated to take all comers. The physical layer is therefore controlled. But like Speakers' Corner, both the logical layer of the language and the content that gets uttered are not controlled in the context of the Garden. They too remain free.

The Telephone System: Before the breakup, the telephone system was a single-unitary system. The physical infrastructure of this system was owned by AT&T; so too was the logical infrastructure – determining how and who you could connect to – controlled by AT&T. But what you said on an AT&T phone (within limits at least) was free: The content of the telephone conversations was not controlled, even if the physical and logical layer underneath were.

Cable TV: Finally, think of cable TV. Here the physical layer is owned – the wires that run the content into your house. The logical layer is owned – only the cable companies get to decide what runs into your house. And the content layer is owned – the shows that get broadcast are copyrighted shows. All three layers are within the control of the cable TV company; no communications layer, in Benkler's sense, remains free.

This range of free and controlled layers constructs very different communication environments. Consider then within this range one more significant communication environment: the Internet.

The Internet is a communication system. It too has three layers. At the bottom, the physical layer, are wires and computers, and wires linking computers. These resources are owned. The owners have complete control over what they do with their wires or computers, or wires linking computers. Property governs this layer.

On top of the physical layer is a logical layer – the protocols that make the net run. These protocols are many, all chucked into a single box called TCP/IP. Their essence is a system for exchanging datagrams, but we miss something important about the system if we focus exclusively on the essence.

For at the core of this logical layer is a principle of network design. At the core of the Internet's design is an ideal called "end-to-end." First articulated by network architects Jerome Saltzer, David Reed, and David Clark, end-to-end or "e2e"says build the network so that intelligence rests in the ends, and the network itself remains simple.[4] Simple networks, smart applications.

The reason for this design was clear. With e2e, innovation on the Internet didn't depend upon the network. New content or new applications could

[4] J. H. Saltzer et al., *End-to-End Arguments in System Design* (Apr. 1981), *available at* <http://web.mit.edu/saltzer/www/publications/endtoend/endtoend.pdf>. *See also* David P. Reed et al., *Active Networking in End-to-End Arguments* (May 1998), *available at* <http://<web.mit.edu/saltzer/www/publications/endtoend/ ANe2ecomment.html>.

run regardless of whether the network knew about them. New content or new applications would run because the network simply took packets of data and moved them along. The fundamental feature of this network design was neutrality among packets. The network was simple, or "stupid" in Bell Labs researcher David Isenberg's words,[5] and the consequence of stupidity, at least among computers, is the inability to discriminate. Innovators thus knew that if their ideas were wanted, the network would run it. This network was designed never to allow anyone to decide what would be allowed.

That means that this layer of the network – the feature of the network that distinguished it from all that had been built before – built this network into a commons. One was free to get access to this network, and share its resources. The protocols were designed for sharing, not exclusive use. Discrimination, at the heart of a property system, was not possible at the heart of this system. This system was coded to be free. That was its nature.

This feature is something new in network design. It contrasts, for example, with the design of the original telephone network. The original telephone network was created not to permit uses other than those it allowed; if you had a different use – if you wanted to connect a modem, for example – then you needed the permission of the network owner, AT&T. No doubt sometimes that permission would be granted – when, for example, the technology advanced the business model of AT&T; but equally certain was that permission would be denied when the technology did not advance the model of AT&T. So, for example, when the design of the Internet was first presented to AT&T – not in 1990, or 1984, or 1976, but in 1964 – said AT&T about this technology, "it can't possibly work, but if it did, damned if we are going to allow the creation of a competitor to ourselves."[6]

Allow. This was the essence of the network that predated the Internet: a network owner got to decide how the network would be used. Not allowing anyone the power to say "allowed" is the design of the network that has replaced the telephone network – the Internet. The essence of this design, the essence of how the original architecture was built, was to deny to the network owner this power to "allow."

Thus, on top of a physical layer that is controlled rests a logical layer that is free. And then on top of this free layer is a content layer that is both free and controlled.

The free part is all the content that effectively rests in the public domain. The facts, data, abandoned property, undiscovered theft – this is the content

[5] LESSIG, THE FUTURE OF IDEAS, *supra* note 2, at 38.

[6] JOHN NAUGHTON, A BRIEF HISTORY OF THE FUTURE: THE ORIGINS OF THE INTERNET 107 (1999).

that is open for the taking and that is taken openly. But it also includes a part dedicated to be open: open source or free software, dedicated to be free.

This free resource does more than entertain or build culture; this free resource teaches the world about how this aspect of the net functions, or is free – as in every web page that both displays and carries its source, so that its source can be copied and modified for different displays.

This free content coexists with content that is controlled: software that is sold; digital content – music, movies, greeting cards – that is controlled. You can link to mp3.com and listen to music that is free; you can link to amazon.com and read a book that is controlled. The network doesn't care much about what linking occurs. It is neutral about the linking, and the result of this neutrality is a mix.

This is a picture of the complexity we call the Internet. At the bottom is a physical layer that is controlled; on top of it is a logical layer that is free; and on top of both is a content layer that mixes free and controlled.

This complexity builds an *innovation commons*. And this commons has been the location of some of the most extraordinary creativity that we have seen. Not innovation in just the .com sense, but innovation in the ways humans interact; innovation in the ways that culture is spread; and most importantly, innovation in the ways in which culture gets built. The innovation of the Internet – built into its architecture – is an innovation in the ways in which culture gets created. Let the .com flame; it won't matter to this innovation one bit. The crucial feature of this new space is the low cost of digital creation, and the low cost of delivering what gets created.

We can say more about the commons built at the logical layer. I've pointed to one – to free or open source software – that defined computing before the Internet was really born, and that continues to define all of the major software that now makes the Internet run. Not just the Linux OS, which fuels most of the servers that run on the net, but also open source server software called Apache, or the programs that distribute most of your mail – Sendmail. All this is open source, or free software; all of this is software that gets distributed with its source; all of this source is open for others to change.

But let me point to a few others. Think of the world wide web itself. The world wide web rides on top of a bit of code called html; html is the source that defines how web pages will be displayed. It is a language for building web pages. This language is very recent. It didn't exist before 1991. But in 1991, Tim Berners-Lee and Robert Calloux, working in a research lab in Switzerland, released to the world the specs for this language, and a protocol that would ride on top of the Internet, http; and this protocol and language together make up the world wide web. But that's not what built the world wide web. Instead,

what built the world wide web was a single command built into every web browser – a command called "reveal source." Even today, on every major browser, you can go to the edit bar, and invoke this command, reveal source. The source that makes the web page you are looking at tick is then revealed to the user. The user is free to take and copy this code – free in the sense that the technology enables it; no doubt most of this code is copyrighted, so not free in the sense that the law permits it. But free in that people could do it. And they did it. The first generation of the world wide web was built on the stolen copies of other people's web pages; the language of html was learned by this stealing and changing. No one sat down at a book to learn html. People took what was out there, and modified it as they wanted.

The code of the web, too, like free software, in this sense, existed in a commons; anyone anywhere could take this code and use it without the permission of anyone else.

Let me draw together these different kinds of commons that mark the space called the Internet, and suggest what it is about them that makes them so important to the Internet. All of them define resources that anyone can draw upon. That anyone can draw upon without the content based permission of someone else. There is a cost to get access, but the access once granted is not conditioned on content-based, or strategy-based, terms such as, "Does this use compete with our existing business model?" No one in this environment – of open source software, or end-to-end design – is in a position to do anything if a new innovation competes with a dominant design. No response is possible because the space creates no structure sufficient to enforce a response.

This design feature has two important consequences for innovation. First, it flattens the field of potential contributors to the innovation of this space. A broader range of innovators can innovate for the platform; innovation and development are not reserved to a select few. To develop technologies for the Internet, one need only a connection to the Internet; to develop technologies for AT&T's monopoly phone company, one needed to work for, or license to, AT&T. Both requirements impose costs on the innovator – not everyone wants to work for a telephone company; not every good idea is a good idea for a telephone company – and these costs are not present in the Internet. Anyone can play, meaning many more do, meaning many more ideas about how best to develop this space get realized.

Second, this design builds into the design a right to revolt. The network is a platform; the platform is built to be unable to resist revolution. If a new idea comes along, even if this new idea destroys the dominant use of the network, there is no structure, no power, that can resist this new idea. Nothing in this

original design gives anyone the power to check one idea over another; nothing in this original design empowers anyone to say "allow."

We can link this design back to the optimism of a revolutionary, Thomas Jefferson. Recall how Jefferson reveled in the conspiracy that nature had launched against the concept of property rights in an idea. How nature had made it impossible for ideas to become property. Ideas would flow freely, this was nature's design; man could do little about this design.

That romanticism about ideas in the Enlightenment is how most now speak about innovation in the Internet. That here, now, the nature of the space is such that innovation can always flourish; that new ideas will always prevail; that old ideas cannot defend themselves. This is the nature of the space, and this nature cannot change.

But it is here, when we put these stories together, that we should see the trouble with this modern Jefferson – and it is with this trouble that I want to end this essay.

For if there is one thing we should know about cyberspace, it is this: nature did not build it. If anything is socially constructed, cyberspace is. How it is constructed is simply a function of its code; its code here, and elsewhere, could be different. If it is built now to enable these two fundamental commons – over platforms, over networks – then it could be built differently to take these commons away. And to the extent you agree with me that these commons contribute to the innovations of the net, how future changes might remove these commons should concern you as well.

Let me end then with three accounts of changes in these commons – actually just one real account, but three indications of where a fuller account could be made. One is from the past (or one wishes it were the past) – Microsoft; one is about the present, and the current struggle over the architecture of broadband Internet access; and one is about the future, about how the future architecture won't be. All three are about changes to the environment of innovation in cyberspace, and all three, I suggest, are examples that, to preserve this environment, we must begin to account.

First the past: Microsoft owned a platform; that platform is the Windows operating system. This operating system has inspired extraordinary innovation upon its platform. By leaving open most of the application program interfaces ("APIs"), by encouraging developers to code to the system, by supporting these projects, and by evangelizing its product, the corporation has done much to assure that the world can build on the platform that Microsoft owns.

But owning the platform means something. In particular, it means the ability to control how the code will evolve. Not perfect control: if Microsoft decided to dump the graphical user interface ("GUI") as the operating system interface on the desktop, and in some radical retro moment, decided to return to the

command line, consumers would react; we would flee the platform, but this fleeing and fury would have an effect on the platform. It would reform it, for at the extremes the customers have that power.

Within the extremes, however, within the detail within which a system gets built, customers don't have much power. Within the extremes, the owner of the code gets to decide how the code will evolve. It gets to decide, that is, whether a browser remains a separable product – whether an application or a system service doesn't matter for the moment. It gets to decide whether other products get to run well on the platform. It gets to decide all this because the owner owns the code and the code keeps itself secret. The code is closed, not open. If someone doesn't like how it is developing, what they can do is limited.

This architecture of a closed platform also has an effect on innovation. Or, at least it has an effect on innovation that threatens the underlying platform – that threatens to weaken its power as a dominant force in the network. For if an innovation develops that the platform doesn't like, then a closed code platform can chose to cancel that innovation. It can choose to refuse it, or confuse it, or embrace it and digest it; it can bundle or bind an alternative; it can displace the competitor; it can play many games to make the competing application have to compete more strongly.

This was the argument, at least, of the U.S. government in its recent action against Microsoft. No one could doubt that in a significant way, Microsoft had fueled innovation. But the charges against the company were based on the ability of the company to target innovations it didn't like. Anytime an innovation threatened its control over the platform – over the APIs to which developers wrote – Microsoft, the government claimed, would intervene to kill that innovation. To capture it. To control it. To displace it. To, as a Microsoft executive said to Apple about QuickTime, "knife the baby".[7]

The platform thus chose which innovations were allowed. And it was empowered to choose because the owners control the code. The platform could behave strategically. It had strategic power because it controls its code. To the extent it exercises that strategic power, it undermines the innovation commons built by neutral, common platforms.

Second, the present: I said that the commons that fuels innovation is the commons that exists at the logical layer of the net. This is the commons constituted by the principle of end-to-end; it is the commons that gets built by a set of protocols that don't discriminate. It is the neutral platform upon which innovation happens. And this neutrality is neutrality built into the code.

[7] *See* Declan McCullagh, *Knifing the Baby*, Wired News (Nov. 5, 1998), *available at* <http://wired-vig.wired.com/news/politics/0,1283,16082,00.html>.

But this code is not given. The code governing a network is not fixed. The code that governs at one time could be replaced by different code later on. And more importantly, there is nothing that forces people who connect to the net to obey the neutrality of the net. There is no brand called "the Internet" that carries with it a set of assumptions about openness and balance; there is instead a basic set of protocols that anyone is free to supplement by adding other protocols on top.

Anyone is free to change the open commons of the Internet, and some important people are changing it. For example, providers of broadband services.

As the Internet moves from the telephone – from modems and 28.8 or 56k connections – to broadband – to fast, always-on connections, the physical layer across which the Internet travels is different. The dominant technology today for serving this broadband content is cable.

As cable converts itself to make itself open to the Internet, it is modifying the architecture of the Internet in an important way. While the essence of the commons of the Internet is neutrality and simplicity, the essence of what the broadband cable Internet will be is the power to discriminate in content and services. The aim of this form of Internet access will not be openness and neutral platforms; the aim of this form of Internet access will be control over the content that gets played.

For example: Cable companies make a great deal of money streaming video to television sets. That is the core of their legacy monopoly power. Some think it would be useful to stream video to computers. Cable companies are not eager to see this form of competition. So they imposed rules on broadband users – no more than ten minutes of streaming video could be contracted for at any time. When they were smart, they said they were worried about congestion. But when they were honest they said something different. Said Daniel Somers, of AT&T, "we didn't spend 56 billion on a cable system to have the blood sucked from our veins."[8]

Broadband providers will insist that this control is their right – that nothing should interfere with their right to layer onto the free logical layer a system of control. And a budding line of First Amendment doctrine strongly supports this claim.[9]

[8] David Lieberman, *Media Giants' Net Change: Major Companies Establish Strong Foothold Online*, USA TODAY, Dec. 14, 1999, at B2.

[9] *See* Robert Corn-Revere, *Broadband Internet Access Debate Heightened by Agencies, Courts, the First Amendment and the Media* (2002), *available at* <http://www.mediainstitute.org ONLINE/FAM2002/BCTV_B.html>.

These cases are Blade-Runner-esque. Remember one of the million amazing puzzles in that extraordinary film is the slow recognition that the machines are human. Well, here too, with a cable system, it is the increasing recognition that these systems to deliver electricity are in fact First Amendment speakers. Wires plus a certain logic entail the press; and then into the mix comes courts eager to bestow on this press long-standing First Amendment power.

And hence we should expect, as the Internet moves to broadband service, that the rules governing the providers will be different. Unlike the telephone company, these providers will be allowed to discriminate, and discriminate they will. When they do, the open feature of the Internet commons will be removed, with the consequence that innovation will be different.

The two changes that I've described so far are changes at the logical layer of the Internet. The third is a change at the physical layer – or more precisely, a change in how the physical layer gets allocated.

The radio spectrum is a resource. For most of the last century, it was not a resource organized as a commons. Early in the last century, the federal government claimed the right to allocate, through license, the right to use the spectrum. Very quickly, Nobel-quality economists like Ronald Coase noted this was a silly way to allocate a scarce resource, that property rights would better allocate the resource of spectrum.[10]

Both of these models for allocating radio spectrum presume something about the nature of this resource. They both presume that spectrum is the sort of thing that must be allocated. Whether by governments or by the market, the presumption is that radio spectrum is like land; that the only way efficiently to use it is to assure clean and clear rights to use it; and the only issue then is who sets the rights.

Behind this picture is a very crude notion about how radio spectrum works. This notion is that radio waves function like jet-liners, destined to crash and fall to the ground if someone doesn't direct how they should fly.

But this is not at all the nature of radio spectrum. And indeed, as technologists demonstrated more than thirty years ago, there is another way we might allocate spectrum such that it is shared, and any "tragedy" from this sharing would be negotiated by the receivers. Rather than allocating the resource *ex ante*, the resource would be allocated by machines on an "as needed" basis.

The technologies for this are many, and increasingly available; were they deployed, they would enable a fast and efficient use of spectrum. But these technologies would permit a use of radio spectrum that was unallocated, uncontrolled. No property regime would govern who got what, when; no

[10] Ronald H. Coase, *The Federal Communications Commission*, 2 J. L. & Econ. 1 (1959).

rules would limit access based on who had accumulated what set of rights. One's right to broadcast using spectrum would be just as one's right to send email across the Internet; broadcasting and receiving would not require the permission of someone else.

This alternative way of allocating and using radio spectrum is in competition with existing models; it is in conflict with the model of Coase that is increasingly being deployed across the world. Spectrum is being sold, and rights are being created and allocated, with no clear understanding of whether such an allocation system is needed.[11]

The result of this property system will be to remove radio spectrum from a commons; it will require the permission of someone else before one can use the spectrum of this space; and this requirement will thus channel and direct the kinds of uses that can be made. Not controlled by a government, no doubt, but neither free in the sense the Internet initially was. Rather, it is more in line with the structures of access of the world before the Internet, producing the kind of speech that was produced in the world before the Internet.

In all three areas – the past, if true, about Microsoft; the present, if allowed, with cable; the future, if we continue, with wireless – I have described changes in a commons that currently define the environment of the Internet. These changes are occurring because, Jefferson notwithstanding, there is no inherent nature of the net that will preserve the commons of its founding. Yet we are allowing these changes to occur without considering the effect the loss of the commons will have on what is most surprising, and extraordinary, about this space: the reality of an explosion in creativity and innovation that is induced by an environment where property is only imperfectly protected. We will lose this reality if we fail to understand its environment and fail to preserve the commons that is at its core.

[11] *See* LESSIG, THE FUTURE OF IDEAS, *supra* note 2, at 225–27, 231–33.

6 How Norms Help Reduce the Tragedy of the Commons: A Multi-Layer Framework for Analyzing Field Experiments

Juan-Camilo Cárdenas and Elinor Ostrom

I. INTRODUCTION

Contemporary economic theory is one of the more successful, empirically verified social science theories to explain human behavior. It does best, however, in the settings for which it was developed – the exchange of private goods and services in an open, competitive market. The theory is based on a theory of goods, a set of rules for social exchange, and a model of human behavior. When the goods involved are easily excludable and rivalrous, and individuals are interacting in a competitive market, theoretical predictions have strong empirical support. When the goods involved are not easy to exclude – such as public goods or common-pool resources (CPRs) – conventional theoretical predictions receive much less empirical support.[1] In a static setting, the conventional predictions are that individuals will not produce public goods and that they will overharvest common-pool resources. The evidence for both predictions is mixed.[2]

[1] *See* SAMUEL BOWLES, MICROECONOMICS: BEHAVIOR, INSTITUTIONS, AND EVOLUTION (2004); COLIN F. CAMERER, BEHAVIORAL GAME THEORY: EXPERIMENTS IN STRATEGIC INTERACTION (2003); Colin F. Camerer, *Bounded Rationality in Individual Decision Making*, 1 EXPERIMENTAL ECON. 163–83 (1998); Colin F. Camerer, *Progress in Behavioral Game Theory*, 11 J. ECON. PERSP. 167 (1997); Herbert Gintis, *Beyond* Homo Economicus: *Evidence from Experimental Economics*, 35 ECOLOGICAL ECON. 311 (2000); John O. Ledyard, *Public Goods: A Survey of Experimental Research*, *in* HANDBOOK OF EXPERIMENTAL ECONOMICS (John Kagel & Alvin Roth eds., 1995).

[2] *See* Elinor Ostrom, *A Behavioral Approach to the Rational Choice Theory of Collective Action*, 92 AM. POL. SCI. REV. 1 (1998).

A portion of an earlier version of this chapter has been published as *What Do People Bring into the Game? Experiments in the Field about Cooperation in the Commons*, 82 AGRIS. SYS. 307 (2004). Support received from the National Science Foundation (Grant SBR-9521918) is gratefully acknowledged. Extensive comments by Marten Beckenkamp, Bruno Frey, Mike McGinnis, and James Walker on past versions helped improve this chapter as well as the excellent editing by Patty Lezotte.