

**THE ONTOLOGY OF CYBERSPACE:
LAW, PHILOSOPHY, AND THE FUTURE OF
INTELLECTUAL PROPERTY**

By David R. Koepsell

Chicago, IL: Open Court Publishing Co. 2000

Pp. 139. \$26.95 (hard). ISBN 0-8126-9423-6

*Rita F. Lin**

Much ink has been spilled and many pixels filled in arguments over the patentability and copyrightability of software. Some jurists and technologists have criticized copyright protection of source code because of its focus on literal copying. Even where copyright does bar copying of non-literal elements such as the program's structure or "look and feel," critics have argued that such protection leaves the most critical innovations in the programs' behavior unprotected and fails to provide sufficient incentive for investment in software development.¹ On the other hand, many have objected to the increasing use of software patents on the grounds that costly patent searches will raise barriers to innovation and prevent reverse engineering necessary for software compatibility.²

David Koepsell's *The Ontology of Cyberspace* is an ambitious attempt to resolve this fifteen-year-old debate over the proper role of intellectual property law in emerging technologies. Drawing on his background as a professor of philosophy, Koepsell argues that the

* J.D. 2003 (expected), Harvard Law School.

1. See, e.g., Dan L. Burk, *Patenting Speech*, 79 TEX. L. REV. 99 (2000); Andrew Chin, *Computational Complexity and the Scope of Software Patents*, 39 JURIMETRICS J. 17 (1998); Donald S. Chisum, *The Future of Software Protection: The Patentability of Algorithms*, 47 U. PITT. L. REV. 959 (1986); Joseph Robert Brown, Jr., Note, *Software Patent Dynamics: Software As Patentable Subject Matter After State Street Bank & Trust Co.*, 25 OKLA. CITY U. L. REV. 639 (2000); see also *State St. Bank & Trust Co. v. Signature Fin. Group, Inc.*, 149 F.3d 1368 (Fed. Cir. 1998), cert. denied, 119 S. Ct. 851 (1999) (extending patent protection to software).

2. See, e.g., Dennis S. Karjala, *A Coherent Theory for the Copyright Protection of Computer Software and Recent Judicial Interpretations*, 66 U. CIN. L. REV. 53 (1997); Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?*, 106 HARV. L. REV. 977 (1993); Anthony L. Clapes et al., *Silicon Epics and Binary Bards: Determining the Proper Scope of Copyright Protection for Computer Programs*, 34 UCLA L. REV. 1493 (1987); League For Programming Freedom, *Software Patents: Is This the Future of Programming?* DR. DOBB'S JOURNAL, Nov. 1990, at 56.

philosophical assumptions behind the legal categories of patent and copyright are untenable. He suggests that emerging technologies such as software and genetic engineering have revealed underlying conceptual contradictions in the distinction between machines protected by patent and writings protected by copyright. Koepsell contends that as these technologies continue to become more critical to our economy, the confusion over which products are protected by patent and which by copyright will create significant barriers to innovation. He therefore advocates a unified conception of intellectual property that will extend protection to “any man-made, intentionally produced objects,” regardless of their intended use (p. 92).

The first half of *The Ontology of Cyberspace* presents a methodological argument about the application of ontology (the study of being) to questions of law. Because ontology is focused on developing a conceptual framework for categorizing objects and understanding the relations between those objects, Koepsell argues that its techniques can be used to analyze the validity of legal categories (pp. 25–31). The first step of the ontological method is to examine the empirical assumptions behind current classifications (which he calls “naive” ontology). The next step is to identify categorizations in the naïve ontology that are logically inconsistent or fail to comport with our daily experiences (which he calls “incorrect” ontology). The last step is to develop a “correct” ontology that eliminates those contradictions (pp. 33–39).

Koepsell’s major claim in this section is fairly noncontroversial. He carefully avoids thorny metaphysical debates implicated in ontology by arguing that agreement on the categorization of objects (the focus of ontology) can be separated from deeper disagreements about the fundamental nature of those objects as “real” or “ideal” (the focus of metaphysics) (pp. 20–27). In the end, Koepsell’s point is simply that ontology makes a difference in legal systems: inconsistent or arbitrary categorizations can produce grave injustices or, at the very least, economic inefficiencies (p. 41).

The second half of the book presents a substantive argument applying Koepsell’s ontological method to copyright and patent. He begins by tracing the history of intellectual property protection of software from the early 1970s, when software was predominantly protected by copyright because courts considered algorithms too similar to mathematical formulas to be patentable,³ to contemporary times, when software is protected under both patent and copyright (pp. 59–71).

3. See, e.g., *Gottschalk v. Benson*, 409 U.S. 63 (1972) (ruling against the patentability of software for converting decimal into binary numbers because of its similarity to a mathematical formula).

Arguing that copyright and patent are mutually exclusive categories, he contends that the dual status of software — covered by both categories of intellectual property protection — is an indication that the current system is based on an incorrect ontology (pp. 85–95). For Koepsell, software is not a new challenge to a well-functioning legal framework, but rather a symptom of existing flaws in assumptions of the current system.

Koepsell posits two plausible bases for distinguishing products protected by patent and copyright and rejects both. First, he considers a framework under which copyright protects aesthetic expressions, while patent protects functional ones. Koepsell rejects this categorization on the argument that functional works always involve some element of expressiveness and aesthetic works are always functional in the sense that aesthetic pleasure is a use (pp. 97–102). Second, he considers the notion of a distinction based on the medium of expression, with copyright protecting writings and patent protecting machines. He rejects this distinction as well, insisting that the medium ought not be confused with the message. For Koepsell, the storage medium is no more than a “substrate” upon which expression is propagated; the same software instructions, for instance, could be “hardwired” into microprocessors, stored on a CD-ROM, loaded onto RAM, or written out in source code on the screen. Koepsell’s point is a straightforward one: the change in the medium ought not determine the level of protection afforded to expression (pp. 88–95).

Finally, Koepsell closes with two alternatives to the incorrect ontology of a system divided between patent and copyright. Using the popularity of freeware and shareware games as a case study, he suggests abandoning intellectual property protection altogether on the hope that voluntary secrecy will provide a sufficient first-mover advantage to reward innovation (pp. 108–110). Alternatively, he proposes a unified system of intellectual property based on a modified version of copyright. The goal would be to eliminate the “false dichotomy” of patent and copyright by offering protection for all intentionally produced works — whether aesthetic or functional, linguistic or physical, writing or machine (p. 111).

On the whole, *The Ontology of Cyberspace* is an interesting, cogently argued critique of the current intellectual property system and its treatment of information technologies. Koepsell’s application of ontology to law is highly original, and the book does a superb job of revealing how the methodologies of ontology might be useful in analyzing legal categories. The major difficulty with *The Ontology of Cyberspace* is its ambitiousness. There is quite a bit of literature on application of patent and copyright to software, and Koepsell covers too

much ground with too little attention to obvious objections that have been raised by scholars in the field.

Koepsell's failure to consider the countervailing literature is most apparent in his summary dismissal of arguments that software represents a novel challenge to the current system. The fundamental assumption of Koepsell's argument is that the dual nature of software — protected by both patent and copyright — reveals something wrong with the ontological categories of intellectual property law rather than something special about the nature of software itself. In this sense, the critical claim of his book, to which he devotes a scant five pages (pp. 85–87, 129–30), is the assertion that “cyberspace is nothing very special” (p. 1). The difficulty is that if software is a special hybrid phenomenon (i.e., both a writing and a machine at the same time), it may be no more than an exception to the generally valid rule that an intellectual product is a writing or a machine, but usually not both. Almost every legal category has borderline cases; if Koepsell is to argue that the “computer has helped establish that there are no clear or valid distinctions amongst the objects of patent law and those of copyright” (p. 102), he must show that software is more than just another borderline case.

The problem for Koepsell is that many legal scholars have persuasively argued that software defies the conceptual molds of patent and copyright precisely because it does constitute a special hybrid: a machine capable of being rewritten and reprogrammed to transform itself into another machine. Because a single computer can behave in an infinite number of ways through the use of different software, the computer combines the utilitarian functionality of a machine with the endless flexibility of written expression. As Pamela Samuelson explains: “By executing different sets of stored instructions, a general purpose digital computer could ‘become’ different machines [M]achines could become writings, and writings could become machines.”⁴

Even beyond the scope of intellectual property law, the question of whether software is a writing or a machine has been a difficult one for legal and non-legal scholars alike. Courts and First Amendment scholars have puzzled over whether source code ought to be treated as speech in the sense that it expresses ideas to those with the technical background to understand them, or conduct in the sense that its execution can carry real lasting functional consequences.⁵ At the same time, anthropologists

4. Pamela Samuelson, *Benson Revisited: The Case Against Patent Protection for Algorithms and Other Computer Program-Related Inventions*, 39 EMORY L.J. 1025, 1128–29 (1990).

5. See *Bernstein v. U.S. Dep't of Justice*, 176 F.3d 1132 (9th Cir. 1999),

and sociologists who study social interaction in cyberspace have also wondered about the dual status of software.⁶ In his ethnography of virtual communities, Julian Dibbell writes:

[T]he computer . . . operates on a principle impracticably difficult to distinguish from the pre-Enlightenment principle of the magic word: the commands you type into a computer are a kind of speech that doesn't so much communicate as *make things happen*, directly and ineluctably, the same way pulling a trigger does. They are incantations, in other words.⁷

None of this is to say that Koepsell is wrong and the multitude of scholars who interpret software as a unique hybrid phenomenon — “a machine whose medium of construction happens to be text,” as one commentator put it⁸ — are right. The real problem is that Koepsell treats software as ontologically unexceptional, and he appears unwilling to take seriously the difficult questions surrounding the novelty of software. His failure to give a considered response to those questions often undermines his claim that the computer has revealed deeper inconsistencies in the current intellectual property system. For instance, he writes: “As in languages, whose flexibility accommodate a possibly infinite number of forms of expression for any general idea, computers may be programmed a possibly infinite number of ways. This flexibility

withdrawn, reh'g granted, 192 F.3d 1308 (9th Cir. 1999); *Junger v. Daley*, 209 F.3d 481 (6th Cir. 2000); *Karn v. U.S. Dep't of State*, 925 F. Supp. 1 (D.D.C. 1996); see also Steven E. Halpern, *Harmonizing the Convergence of Medium, Expression, and Functionality: A Study of the Speech Interest in Computer Software*, 14 HARV. J.L. & TECH. (forthcoming 2000); Yvonne C. Ocrant, *A Constitutional Challenge to Encryption Export Regulations: Software is Speechless*, 48 DEPAUL L. REV. 503 (1998); Norman A. Crain, Comment, *Bernstein, Karn, and Junger: Constitutional Challenges to Cryptographic Regulations*, 50 ALA L. REV. 869 (1999); R. Polk Wagner, Note, *The Medium is the Mistake: The Law of Software for the First Amendment*, 51 STAN L. REV. 387 (1999).

6. See, e.g., DONNA J. HARAWAY, *SIMIANS, CYBORGS, AND WOMEN: THE REINVENTION OF NATURE* (1991); SHERRY TURKLE, *LIFE ON THE SCREEN: IDENTITY IN THE AGE OF THE INTERNET* (1995); Allucquere Rosanne Stone, *Will the Real Body Please Stand Up?: Boundary Stories about Virtual Culture*, in *CYBERSPACE: FIRST STEPS* 81 (Michael Benedikt ed., 1991).

7. JULIAN DIBBELL, *MY TINY LIFE: CRIME AND PASSION IN A VIRTUAL WORLD* 27 (1998) (emphasis in original).

8. Pamela Samuelson et al., *Toward a Third Intellectual Property Paradigm: A Manifesto Concerning the Legal Protection of Computer Programs*, 94 COLUM. L. REV. 2308, 2320 (1994).

has demonstrated that atoms and bits, like language, are just other media of expression" (p. 103). The critical issue that Koepsell ignores in this passage, of course, is whether computers are uniquely flexible compared to other machines. Although he cites player pianos and Jacquard looms (manual looms that allow weavers to set pre-determined patterns) as examples of machines capable of different behaviors, his implicit claim that such flexibility is the rule rather than the exception among machines remains rather dubious.

If he is to argue that all man-made intentionally produced objects ought to be treated identically for purposes of intellectual property protection, Koepsell must answer the more fundamental question of whether some valid basis exists for treating writings and machines differently — notwithstanding borderline cases such as software. Here again, Koepsell runs into a host of possible objections. One possible rationale for the distinction is that copyright and patent reflect different degrees of "merger" between the intellectual product and the underlying idea. In copyright, courts have assumed that written works offer a nearly infinite number of ways to express the same underlying idea.⁹ As Koepsell himself points out, although the plot of every murder mystery novel might be essentially the same, there is little monopoly danger in offering copyright protection to each individual author because there are so many ways to tell and retell the same old yarns. In patent, by contrast, the merger between the idea and its implementation in a useful machine is usually regarded as much tighter. If there are a limited number of processes for developing a newly discovered chemical synthetic, for instance, the danger of monopoly over the unpatentable underlying idea (i.e., the chemical substance itself in this case) is much greater. To the extent that machines generally foreclose the development of an underlying idea more than writings, it would make sense to draw distinctions between writings and machines in determining the threshold requirements for protection (i.e., minimal originality for copyright versus novelty, usefulness, and non-obviousness for patent), length of protection (i.e., author's lifetime plus seventy years for copyright versus twenty years for patent), and processes for obtaining grants of protection (i.e., none for copyright versus submitting a detailed application for patent).

9. Indeed, under the "merger doctrine," courts have refused to grant copyright protection in cases where there is only one or a limited number of ways to express the underlying idea. *See, e.g.,* *Apple Computer, Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1253 (3d Cir. 1983); *Hoehling v. Universal City Studios, Inc.*, 618 F.2d 972, 979 (2d Cir. 1980). *See generally* Jeannette Rene Busek, Comment, *Copyright Infringement: A Proposal for a New Standard for Substantial Similarity Based on the Degree of Possible Expressive Variation*, 45 UCLA L. REV. 1777 (1998).

Another argument for distinguishing between the protection of writings and machines is that they operate in different markets and contribute differently to social utility. With artistic expression, the goal tends to be to maximize the total quantity of works produced. As Paul Goldstein explains:

The method of art, after all, is to refract the individual author's unique viewpoint in language, line or music. We are thus better off with hundreds of different novels, paintings and musical compositions, each distilling the individual author's perception of reality, than we are with a single view of that reality.¹⁰

By contrast, with patented technologies, there are strong policy reasons to encourage investment only in substantial technological developments; the goal is to develop the most efficient transistors, not necessarily to encourage the development of different kinds of transistors purely for the sake of variety. Goldstein writes: "Society's interest in technological advance, although also served by differentiation, needs fewer resources devoted to variations on a single technological solution, and more to verifiable improvements on that solution."¹¹ In this sense, the distinction between writings and machines arises out of differences in the purpose of aesthetic and utilitarian items in our society, as well as discrepancies in the role of improvement in such products.

Obviously, these are not the only possible reasons for treating writings and machines differently. Writings and machines might face higher or lower enforcement costs in prosecuting infringement, faster or slower payback rates from the market, greater or lesser costs of copying, and countless other market differences. Here again, the point is not that Koepsell is wrong to reject these explanations, but rather that his argument suffers because of his failure to address them.

Nowhere is this failure more evident than in his proffered solution of a unified system of intellectual property protection. Koepsell leaves countless questions unanswered about how such a scheme might reconcile the different goals of copyright and patent. Would it require only copyright's minimal level of originality or patent's more stringent requirements of novelty, usefulness, and non-obviousness? Would it permit copyright's defenses of fair use? Would independent creation be a complete defense as in copyright or would it still constitute

10. Paul Goldstein, *Infringement of Copyright in Computer Programs*, 47 U. PITT. L. REV. 1119, 1122-23 (1986).

11. *Id.* at 1123.

infringement as in patent? Would infringement require copying every element (as in patent) or only a minimal level of copying (as in copyright)? Would owners of intellectual property have to register their products with an office in advance, as patent has required, or would protection be automatic as soon as the product is completed, as copyright has permitted? Could the court impose remedies typically limited to copyright, such as compulsory licensing?¹² The vagueness of Koepsell's solution reveals the extent to which he ignores significant disparities in the purposes of patent and copyright.

Indeed, if software is a borderline case for a generally valid distinction between writings and machines, Koepsell would be hard-pressed to defend his proposal over the current alternatives proposed by other scholars. Many commentators have suggested preserving the existing system and adopting *sui generis* intellectual property legislation tailored to special hybrids such as computer software.¹³ Congress has already begun enacting targeted protection for hybrids through laws such as the Semiconductor Chip Protection Act,¹⁴ which extends a ten-year copyright-like protection to patterns (known as "mask works") used to etch circuitry into computer chips, and the Plant Variety Protection Act,¹⁵ which extends intellectual property protection to genetically engineered plants capable of sexual reproduction. Others have suggested alternatives such as recognizing software as patentable but not copyrightable or vice versa.¹⁶ And still others have proposed reconceived versions of copyright and patent.¹⁷ Unless Koepsell can

12. For a thorough discussion of the basic differences between the scope of protection offered in patent and copyright, see Maureen A. O'Rourke, *Toward a Doctrine of Fair Use in Patent*, 100 COLUM. L. REV. 1177, 1184-85 (2000).

13. See generally John C. Phillips, *Sui Generis Intellectual Property Protection for Computer Software*, 60 GEO. WASH. L. REV. 997 (1992); J.H. Reichman, *Toward a Third Intellectual Property Paradigm: Legal Hybrids Between the Patent and Copyright Paradigms*, 94 COLUM. L. REV. 2432 (1994); see also Irwin R. Gross, *A New Framework for Software Protection: Distinguishing Between Interactive and Non-Interactive Aspects of Computer Programs*, 20 RUTGERS COMPUTER & TECH. L.J. 107, 177-86 (1994); Peter S. Menell, *Tailoring Legal Protection for Computer Software*, 39 STAN. L. REV. 1329, 1364-67, 1371 (1987).

14. 17 U.S.C. §§ 901-914 (2000). Koepsell mentions the Semiconductor Chip Protection Act (pp. 70-71), but never discusses the possibility of similar *sui generis* legislation for computer software.

15. 7 U.S.C. §§ 2321-2582 (2000).

16. See *supra* notes 2 and 3.

17. See, e.g., Pamela Samuelson, *Creating a New Kind of Intellectual Property: Applying the Lessons of the Chip Law to Computer Programs*, 70 MINN. L. REV. 471, 502-03 (1985) (arguing for replacing the distinction between writings and machines with a distinction between original works of authorship and non-obvious implementations of ideas); Chin, *supra* note 2 (suggesting modified patent protection of software); Karjala, *supra* note 3 (proposing modified copyright protection of software).

convincingly argue that the difficulties faced in the intellectual property protection of software are the norm rather than the exception, his solution will remain far less appealing than alternatives that seek to accommodate software protection within the existing framework.

One wonders whether Koepsell misses many of these objections precisely because his method of analysis blindly focuses on ontology alone. Because the distinction between copyright and patent is ostensibly based on the difference between machines and writings, Koepsell addresses the bulk of his argument to suggesting that expression ought to be protected regardless of the medium of expression. However, the distinction between writings and machines often serves as mere shorthand for a host of associated characteristics. Machines are offered patent protection not just because they are machines, but also because they generally tend to apply scientific ideas and pose greater dangers of closing off avenues of innovation. Koepsell misses the underlying reasons behind the categorization because he takes the superficial basis of categorization (i.e., based on whether a work is a machine or a writing) at face value.

Of course, Koepsell has an easy response: if the real basis for distinguishing patent and copyright is the danger of foreclosing innovation, then the legal system ought to differentiate its protection based on the degree of such danger rather than based on the distinction between machines and writings. The objection to this response cuts to the heart of the problem with the ontological method. Although it might be more ontologically correct to distinguish protection based on the danger of foreclosing innovation, the costs of administering such a categorization would be enormous. Because no one — not judges, business experts, or even hopeful owners of intellectual products — can know in advance which works would be most likely to foreclose innovation, the obvious solution is to find a relatively clearer distinction that roughly maps onto the desired characteristics. Although intellectual property protection of machines might not always pose a greater monopoly danger more than the protection of writings, the writing-machine distinction is relatively straightforward and will usually ensure that avenues to innovation stay open more often. Ontological correctness is sacrificed to the practical goals of creating workable distinctions.

In this sense, although Koepsell is right to suggest ontology as a tool for understanding legal categories, he is wrong to insist on its primacy. Whereas philosophy seeks to develop ontological categories that accurately reflect the objects we perceive in the world around us, the law categorizes not for the sake of accurate ontology, but as a means of furthering social ends. Despite the existence of difficult hybrids such as

software, we might well decide that we are satisfied with the current distinction between writings and machines if it hits the mark more often than not. In this sense, the costs of adopting a correct ontology could be prohibitive in law where they would not be in philosophy.

In the end, the greatest strength of *The Ontology of Cyberspace* is Koepsell's courage in venturing into uncharted waters. Although many commentators have written on patent, copyright, and the intellectual property protection of software, Koepsell is the first to suggest ontology as a method for analyzing the issue. Unfortunately, perhaps because he is exploring the unknown, Koepsell glosses over difficult points in his argument. He never convincingly demonstrates that software is more than just a borderline case for a generally valid distinction between patent and copyright; he never responds to prevailing arguments in support of distinguishing machines and writings; he never articulates how his solution of a unified intellectual property regime will reconcile the differences between patent and copyright; and, most importantly for his methodological argument, he simply never considers the countervailing considerations that might weigh against ontological correctness.

Ultimately, the flaws of Koepsell's substantive legal argument undermine the credibility of his primary philosophical project—namely, the methodological importance of ontology in law. The major claim of *The Ontology of Cyberspace* is well-taken: ontology does matter in law. But it seems to matter less than Koepsell would have us believe.