

YOU SAY MISUSE, I SAY FAIR USE . . .

By Jonathan Band and Noah Levine¹

Long before the emergence of the Law and Economics movement at the University of Chicago Law School or Critical Legal Theory at Harvard Law School, professors at the Yale Law School taught Legal Realism. The Legal Realists argued that judges typically reached a result in a case based on their idiosyncratic notion of which side deserved to win, and then found a legal rationale justifying that result.² Indeed, Critical Legal Theory arguably is just a refinement — or a perversion — of Legal Realism.

Three recent software copyright decisions — *Lotus Dev. Corp. v. Borland Int'l*, 49 F.3d 807 (1st Cir. 1995), *aff'd by equally divided Court*, 116 S. Ct. 804 (1996), *Bateman v. Mnemonics, Inc.*, 79 F.3d 1532 (11th Cir. 1996), and *DSC Communications Corp. v. DGI Technologies, Inc.*, 81 F.3d 597 (5th Cir. 1996) — can be best explained by Legal Realism. In all three decisions, the courts concluded that the defendant software developers should be permitted to make copies of the plaintiffs' programs to the extent necessary to achieve interoperability. Once they reached that same conclusion, they selected different legal theories justifying how they got there.

By this interpretation, we by no means suggest that the First, Fifth or Eleventh Circuits were unprincipled in any way. Rather, we simply acknowledge that until the law in a particular area is well settled, almost any result can be justified by several plausible legal theories. Under those circumstances, there is nothing unprincipled about a court selecting a victor based on notions of fairness and consumer welfare, and then finding the strongest legal basis for reaching that result. In *Borland*, *Bateman*, and *DSC Communications*, we believe that the courts recognized the importance of interoperability to competition within the software industry, and therefore interpreted the Copyright Act in a manner which did not conflict with interoperability.

¹ Jonathan Band is a partner in the Washington, D.C. office of Morrison & Foerster. Noah Levine is a summer associate at Morrison & Foerster and a third year law student at Columbia Law School.

² See Karin Wentz, *1987 Survey of Books Relating to the Law*, 85 Mich. L. Rev. 1105 (1987); Joseph W. Singer, *Legal Realism Now*, 76 Cal. L. Rev. 465, 471-72 (1988) (both reviewing Laura Kalman, *Legal Realism at Yale, 1927-1960* (1986)).

I. LOTUS v. BORLAND

In *Borland*, the First Circuit held that Lotus' 1-2-3 command structure constituted a method of operation unprotected under 17 U.S.C. § 102(b).³ Judge Boudin's concurring opinion is a text book example of Legal Realism.⁴ At the outset, he expressed the uneasiness with which federal judges approach questions of copyright protection for functional elements of computer software: "to assume that computer programs are just one more new means of expression, like a filmed play, may be quite wrong. . . . Applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit."⁵ Because software had "mechanical utility," protection under copyright could have the effect of patent protection, *i.e.*, limiting the ability of individuals to carry out a task in the most efficient manner.⁶

Turning to the facts before him, Judge Boudin remarked that "the present case is an unattractive one for copyright protection" *Id.* at 821. Judge Boudin noted that "the menu commands . . . are largely for standard procedures that Lotus did not invent and are common words that Lotus cannot monopolize." *Id.* Further, if Lotus could receive a copyright in its command structure, users who invested in learning Lotus 1-2-3 and writing 1-2-3 compatible macros would be "locked" into Lotus; the copyright would preclude competitors from developing products which could interoperate with the installed base. After explaining the economic inefficiencies of users remaining "captives of Lotus," Judge Boudin concluded that for him, "the question is not whether Borland should prevail, but on what basis."

Judge Boudin's concurrence is a self-conscious demonstration of Legal Realism in action. Unfettered by the traditional modes of legal analysis, he first reached a conclusion concerning the undesirability of protecting certain elements of a computer program. Then, recognizing the necessity of identifying legal theories supporting the outcome, he proposed two alternatives sufficient for the task. First, he observed that the basis adopted by the Court—treating the command structure as an unprotected method of operation—was "a defensible position." Second, he stated that Borland's use should be "privileged" because "Borland is merely trying to give former Lotus users an option to exploit their own prior investment in

³ The *Borland* decision is discussed in greater detail in Jonathan Band, *Lotus v. Borland Through the Lens of Interoperability*, *The Computer Lawyer* (June 1995), at 1.

⁴ *Borland*, 49 F.3d at 819 (Boudin, J., concurring).

⁵ *Id.* at 820.

⁶ *Id.* at 821.

learning or in macros.” Judge Boudin suggested that the fair use doctrine could serve as the source of this privilege. As between the two alternatives, Judge Boudin candidly says that the majority’s “formulation is as good, if not better, than any other that occurs to me now as within the reach of courts.” *Id.* at 822.

In sum, based on his understanding of the goals of the Copyright Act and the economics of the software market, Judge Boudin decided that Borland should prevail. Judge Boudin then searched—and found—plausible theories supporting that conclusion.

II. *BATEMAN v. MNEMONICS*

The Eleventh Circuit in *Bateman v. Mnemonics* was less obvious about its approach than Judge Boudin, but just as pragmatic. In *Bateman*, the court considered what the trial court should have instructed the jury concerning the copying of elements dictated by compatibility requirements.⁷ Defendant Mnemonics on appeal argued that the trial court erred in not directing the jury to filter out those portions of the Bateman operating system dictated by the interface with Mnemonics’ application program.

Judge Birch, writing for the panel, noted at the outset the critical significance of separating idea from expression in the context of computer programs:

It is particularly important to exclude methods of operation and processes from the scope of copyright in computer programs because much of the contents of computer programs is patentable. Were we to permit an author to claim copyright protection for those elements of the work that should be the province of patent law, we would be undermining the competitive principles that are fundamental to the patent system.⁸

Notwithstanding this view, Judge Birch stated that if Mnemonics was arguing that interface specifications are not copyrightable as a matter of law, it was wrong: “[i]t is an incorrect statement of the law that interface specifications are not copyrightable as a matter of law.” *Id.* Judge Birch hastened to add, however, that the trial court had erred “in not instructing the jury on the legal consequences of copying dictated by compatibility requirements.” *Id.* Judge Birch explained that other circuits had found “that external factors such as compatibility may work to deny copyright protection to certain portions of a computer program.” *Id.* He then stated that “[w]hether the protection is unavailable because these factors render the expression

⁷ The *Bateman* decision is discussed in greater detail in Jonathan Band, *Interoperability After Lotus v. Borland: The Ball is in the Lower Courts Again*, *The Computer Lawyer* (March 1996), at 11.

⁸ *Bateman*, 79 F.3d at ____.

unoriginal, nonexpressive per 17 U.S.C. § 102(b), or whether these factors compel a finding of fair use, copyright estoppel, or misuse, the result is to deny copyright protection to portions of the computer program.” *Id.*

In other words, Judge Birch, like Judge Boudin before him, first identified his concern — that the copying necessary to the development of interoperable computer software should not necessarily be precluded by copyright law — and only then proposed various alternatives for reaching such a conclusion through accepted forms of legal reasoning. And by listing five alternatives, Judge Birch made clear that he did not much care which legal theory the lower court used, so long as it reached the right result.

III. DSC COMMUNICATIONS CORP. v. DGI TECHNOLOGIES, INC.

Judge Birch’s list of legal theories included both theories mentioned by Judge Boudin — an idea or method of operation unprotected under 17 U.S.C. § 102(b) and fair use. The list also includes the theory relied upon by the Fifth Circuit in *DSC Communications*: copyright misuse.

Both DSC and DGI manufacture various forms of telecommunications equipment, including microprocessor cards for phone switching systems. The conflict between the companies arose when DGI began developing a microprocessor card which would be compatible with the DSC-manufactured phone switch, thereby making it possible to substitute for the card produced by DSC.

There were actually three forms of copying at issue in the case as presented to the district court. First, DGI purchased some of DSC’s microprocessor cards on the open market and disassembled the programs embedded on those chips (“firmware”) into human readable form. Relying on *Sega v. Accolade*, the district court found this copying to be a fair use because the reverse engineering was necessary to discover the unprotectable elements within the firmware.⁹ Second, DGI made removable, identical copies of the operating system in the DSC phone switch and brought the copies back to the DGI laboratories for study. The district court found this to be actionable infringement and granted DSC’s motion for a preliminary injunction enjoining such copying.¹⁰

Finally, in order for any microprocessor card to actually operate in the DSC phone switch, it was necessary that the card download DSC’s copyrighted operating system. Thus, for

⁹ See *DSC Communications Corp. v. DGI Technologies, Inc.*, 898 F. Supp. 1183, 1188–92 (N.D. Tex. 1995), *aff’d*, 81 F.3d 597 (5th Cir. 1996).

¹⁰ See *id.* at 1193–96.

DGI to successfully test and run its product, it was necessary to place its microprocessor cards in the DSC phone switch and thereby cause a RAM copy of the DSC operating system to be made. The district court implicitly found this form of copying to be permitted. It did so by limiting the preliminary injunctive order such that it did not prohibit “downloading into dynamic RAM on a microprocessor or test microprocessor card which is incidental to the testing or operating of a compatible [microprocessor] card so long as the copy is not capable of being removed from the customer location and transported to any other location.”¹¹

The issue on appeal to the Fifth Circuit concerned only the last form of copying. DSC argued that the preliminary injunction was too narrowly drawn because it did not also prohibit the copying incidental to testing or operating of DGI’s newly created microprocessor cards. Because DGI conceded that a RAM copy of the DSC operating system software was made whenever the microprocessor cards were booted up, DGI relied on the copyright misuse defense.¹²

Judge Garza, writing for the Fifth Circuit, expressed the same concerns as Judges Boudin and Birch with the patent-like protection copyright law might be construed to bestow when applied in the computer software context. He noted that:

DSC seems to be attempting to use its copyright to obtain a patent-like monopoly over unpatented microprocessor cards. Any competing microprocessor card developed for use on DSC phone switches must be compatible with DSC’s copyrighted operating system software. In order to ensure that its card is compatible, a competitor such as DGI must test the card on a DSC phone switch. Such a test necessarily involves making a copy of DSC’s copyrighted operating system, which copy is downloaded into the card’s memory when the card is booted up. If DSC is allowed to prevent such copying, then it can prevent anyone from developing a competing microprocessor card, even though it has not patented the card.¹³

Seeking to prevent this result, Judge Garza latched on to the copyright misuse theory, arguing that it “forbids the use of the copyright to secure an exclusive right or limited monopoly not

¹¹ *Id.* at 1197.

¹² The Fifth Circuit appears to accept the Ninth Circuit’s ruling in *MAI Sys. Corp. v. Peak Computer, Inc.*, 991 F.2d 511 (9th Cir. 1993), *cert. dismissed*, 510 U.S. 1033 (1994), that a RAM copy is a “copy” within the meaning of the Copyright Act.

¹³ *DSC Communications*, 81 F.3d at 601. For similar concerns expressed by Judges Birch and Boudin, *see Bateman*, 79 F.3d at 1547 n.33 (“In no case, however, should copyright protection be extended to functional results obtained when program instructions are executed and such results are processes of the type better left to patent and trade secret protection.”); *Borland*, 49 F.3d at 819 (Boudin, J., concurring) (“Granting protection, in other words, can have some of the consequences of *patent* protection in limiting other people’s ability to perform a task in the most efficient manner.”).

granted by the Copyright Office.” *Id.*, quoting *Lasercomb America v. Reynolds*, 911 F.2d 970, 977 (4th Cir. 1990).

What is distinctive about *DSC Communications* is that Judge Garza chose to act on his concern about *de facto* patent protection through the affirmative defense of copyright misuse. In *Lasercomb*, the Fourth Circuit based its misuse finding on the fact that Lasercomb — by the terms of its copyright license agreement — was attempting to monopolize something which clearly was not part of the bundle of rights granted by copyright: the right to develop competing software utilizing the same ideas. Lasercomb could not make even a colorable argument that a copyright holder could effectively prevent other parties from creating a computer program which would be similar only in that it would accomplish the same task.

Although DSC prohibited customers by license from booting up its operating system software on non-DSC equipment,¹⁴ Judge Garza did not rest his decision on that license agreement. He instead found that the attempt by DSC to prevent actual, identical copies from being made by another party to constitute copyright misuse. This clearly extends the reasoning of *Lasercomb*. The right to create a program implementing only the same ideas—the activity Lasercomb sought to control—is not one of the copyright holder’s exclusive rights under the Act. Conversely, the right to reproduce the actual program in question—which DSC sought to control—is an exclusive right of the copyright holder.¹⁵

What is even more remarkable about Judge Garza’s choice of copyright misuse as the basis for his decision is that he preferred the new, not widely accepted misuse defense over the venerable fair use defense, which also was available to him. Indeed, the district court upheld the fair use defense with respect to a different form of copying: it found DGI’s disassembly of DSC’s firmware to constitute fair use under *Sega Enterprises Ltd. v. Accolade, Inc.*,¹⁶ because the disassembly was “the only way to understand the functional elements of DSC’s firmware.”¹⁷ Nonetheless, the Fifth Circuit became the first court since *Lasercomb* to find for the defendant on the basis of copyright misuse.

¹⁴ See *DSC Communications*, 81 F.3d at 600.

¹⁵ See 17 U.S.C. § 106 (1994).

¹⁶ *Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510 (9th Cir. 1992).

¹⁷ *DSC Communications*, 898 F. Supp. at 1190.

CONCLUSION

From a strict Legal Realist perspective, it makes little difference whether Judge Garza selected misuse or fair use. The result is the same: DGI's copying is excused. But the selection does make a difference to the extent that the decision sends a signal to other courts and to the software industry. The fair use defense focuses on the actions of the defendant: was her use justified? Similarly, a defense predicated on Section 102(b) focuses on whether the defendant is entitled to copy the work at issue. The misuse defense, by contrast, focuses on the actions of the plaintiff: did he overreach? In *Borland* and *Bateman*, as well as the Twin Peaks of software copyright cases, *Computer Associates v. Altai* and *Sega v. Accolade*, the court in essence asked whether the *defendant* was permitted to make the copies necessary to achieve interoperability. In *DSC*, Judge Garza asked an altogether different question: was the *plaintiff* permitted to use his copyright to frustrate interoperability? Judge Garza's selection of misuse, and the resulting focus on the actions of the plaintiff, reflect a deepening commitment of the judiciary toward software interoperability.

This deepened commitment is all the more important in the wake of the Seventh Circuit's recent decision in *ProCD v. Zeidenberg*.¹⁸ There, Judge Easterbrook suggested that a contractual restriction on the copying incidental to software reverse engineering would not be preempted by the copyright laws. The *DSC* decision offsets *ProCD* by intimating that enforcing such a restriction would constitute copyright misuse.

¹⁸ *ProCD v. Zeidenberg*, 86 F.3d 1447 (7th Cir. 1996).