

BOOK NOTE

SOFTWARES: THE LEGAL BATTLES FOR CONTROL OF THE GLOBAL SOFTWARE INDUSTRY

By Anthony Lawrence Clapes.¹

Westport, Connecticut: Quorum Books. 1993.

Pp. 317. \$45.00 (hard).

Much has been said in both high-tech and legal circles about how much protection the law should accord computer software. In particular, many computer scientists are apprehensive about the so-called "look and feel" cases, such as *Lotus Development Corp. v. Paperback Software*,² that appear to give monopoly power to companies that get the public dependent upon their particular user interfaces (elements of a person's interaction with a program, such as the screen displays and possible inputs). Collateral to this fear is the suspicion that the people who are deciding these issues are lawyers and judges who have no understanding of the computer industry or the needs of its professionals. In *Softwares: The Legal Battles for Control of the Global Software Industry (Softwares)*, Anthony Lawrence Clapes guides the reader through the history of American legal software protection and argues that much of the current law, which applies traditional principles of copyright and other intellectual property law to software, is both predictable and correct.

Clapes aims to write "a book about intellectual property protection for computer programs that is designed to reach, and satisfy, a diverse audience of computer programmers, lawyers, judges, policy-makers, and interested users" (pp. 5-6). On that score, *Softwares* is definitely a success. It provides a comprehensive explanation of both technical and legal concepts that need to be explained to the nonspecialist. Clapes most thoroughly discusses copyright law, the branch of intellectual property law most applicable to software, however he also explores the use of patent and trade secret protection for computer programs.

1. Assistant General Counsel at IBM.

2. 740 F. Supp. 37 (D. Mass. 1990).

The book is divided into five parts spanning nineteen chapters. In Part I, Clapes lays out the landscape of his tour. First, he argues that success in the computer industry has come to mean success in the software industry (pp. 4, 23). A corollary to this is that the outcome of legal battles regarding intellectual property rights in software is crucial to determining such success (p. 5). Who are the players in these battles that Clapes calls the "softwars"? He explains, "the softwars represent the collision of two fundamentally different paradigms of industrial competition: innovation and imitation" (p. 6). The two sides of the softwars, according to Clapes, are the "originators," those who compete by developing advanced products, and the "copiers," those who compete by producing similar products at a lower cost (pp. 6, 8). Later, Clapes explains why: Software is easy to copy; therefore, the more strongly intellectual property law protects software, the better the position of the originators in their search for profits and capital at the expense of the copiers (pp. 22-23). On the other hand, the weaker the legal protection, the better the position of the copiers, who incur less liability through their business activities.

Clapes gives a persuasive explanation of the axiom in software protection law that computer programs are a form of creative expression (pp. 9-16). This implies that copyright is generally the proper type of intellectual property protection for software, just as it is for other forms of expression, such as books or music (p. 16). Clapes argues that it is the law's recognition of the expressive nature of software and the consequent application of traditional copyright law by the courts that make software law predictable and sensible, contrary to widespread belief (p. 19).

Before moving on to Part II, Clapes gives the reader a brief history of software protection law up to 1990, the year *Lotus v. Paperback* was decided. Along the way, he gives a short, helpful description of copyright, patent, and trade secret protection, and the differences between them (pp. 26-29). In Part II, Clapes takes on the difficult task of explaining *Lotus v. Paperback*, the first software "look and feel" case. Lotus owned the rights to the revolutionary spreadsheet program Lotus 1-2-3. Consumers liked Lotus 1-2-3 because it had an easily understood, perhaps even intuitive, format and menu of commands; its popularity at the time practically raised these features to the level of a common standard. Paperback Software International marketed a competing program, VP-Planner, that was meant to allow the user to operate as if

working with Lotus 1-2-3 "keystroke for keystroke" (p. 43). Lotus sued for copyright infringement. Both sides agreed that while Paperback's business strategy was to produce a program that was as similar to Lotus 1-2-3 as possible, that strategy did not involve any copying of the latter's source code, which would have clearly violated Lotus' copyright. The question to be decided was whether or not the nonliteral elements of Lotus' program, such as the screen displays, were copyrightable. If so, Paperback's strategy constituted infringement. Lotus claimed that nonliteral elements were protected; Paperback vigorously contested this claim. In a long and detailed opinion, Judge Keeton found for Lotus.

To many people who have only read about this decision in the general or high-tech press, *Lotus v. Paperback* was a bombshell standing for the proposition that a company can monopolize the format of a program and its interaction with the user, its "look and feel." In the most dire scenario, if a program's look and feel becomes popular enough to become a common standard, then consumers will be at the mercy of the program's producer, since no one else will be allowed to compete. Clapes argues that such a view is a total misunderstanding of the *Lotus* holding and the law.

Clapes does an admirable job of dissecting the *Lotus* opinion. He notes, "although the case is widely known as a 'look and feel' case, Judge Keeton did not find the term . . . particularly helpful and did not use it" (p. 43). Clapes correctly explains that the focus of the case was on the more precise problem of which elements of Lotus 1-2-3 were only ideas (and thus not protected by copyright law) and which elements constituted expression (which is protected). Under traditional copyright law, nonliteral elements of a work may constitute expression and thus be protectable. Judge Keeton reasoned that, in the context of computer programs, not only was the written code protectable, as both parties agreed, but manifestations of the program's expression that were unnecessary to its ideas were also protected. The crucial question thus became: Which nonliteral elements of Lotus 1-2-3 were ideas, and which were protectable expression?

Clapes frankly writes, "the court's exposition . . . becomes somewhat cryptic at this point, reflecting the surpassing difficulty of putting into words the subjective and intuitive judgments by which idea is parsed from expression" (p. 47). He then goes on to explain the four factors Judge Keeton found relevant in determining whether something was an idea or an expression of an idea (p. 47) and the judge's application of the law as

deduced to the dispute between Lotus and Paperback (pp. 48-52).

Clapes closes this chapter by pointing out that he had published views about how he thought the law should treat software protection, in light of current law (pp. 53-54). These views turned out to be close to the conclusions drawn in *Lotus v. Paperback*. This is evidence, argues Clapes, that the law is predictable (p. 54).

In the remainder of Part II, Clapes discusses the copiers' reaction to Judge Keeton's opinion, highlighting events such as the formation of the League for Programming Freedom, "the radical fringe of the programming community: those who believe that the way to end the softwars is to deprive software of virtually all legal protection" (p. 65). He also spends a chapter on the case of *Feist Publications, Inc. v. Rural Telephone Service Company, Inc.*,³ a 1991 Supreme Court case which essentially held that a white pages telephone book is not copyrightable. Clapes writes that after this decision, "[a]lmost immediately, the weak protectionists began bootstrapping the argument against protectability of white pages into arguments against protectability of computer programs" (p. 72). However, he argues that "the analogy is unconvincing" (p. 72) and the case has "[n]othing, really" (p. 76) to do with copyright protection of software user interfaces. Finally, he explains two other "look and feel" cases, *Ashton-Tate Corp. v. Fox Software, Inc.*⁴ and *Apple Computer Corp. v. Microsoft*,⁵ and examines the issues related to patent protection of software.

In Part III, Clapes discusses the practice of "reverse engineering," the technique of analyzing a program's object code and then deducing its source code in order to be able to write the same or similar program. Here, Clapes' tour of software law takes on an international scope through the story of a 1989-90 softwars battle for favorable treatment in the European Community's proposed uniform software copyright law. Clapes, telling the tale of a certain lawyer for IBM on a high-speed train to Brussels, argues that reverse engineering does not provide benefits that would justify its exemption from copyright law, and that attempts by companies such as Japan's Fujitsu to distinguish such activities from illegal outright copying are flawed. Clapes attacks contentions that present copyright protections grant an artificial monopoly to software

3. 499 U.S. 340 (1991).

4. 760 F. Supp. 831 (C.D. Cal. 1990).

5. 1992 U.S. Dist. LEXIS 5986 (N.D. Cal. April 14, 1992).

innovators and that permitting reverse engineering would promote progress in the software industry (pp. 130-31). In subsequent chapters, Clapes becomes less adversarial and tells the dramatic stories behind the landmark Australian case of *Autodesk, Inc. and ANOR v. Martin Peter Dyason and ORS*⁶ and an interesting Japanese case, *Microsoft Corp. v. Shuuwa System Trading, K.K.*⁷ Finally, Clapes recounts the history of the litigation between IBM and Allen-Myland Inc.⁸

In Part IV, Clapes ventures from "the mainstream issues of the softwars" examined thus far, "look and feel" and reverse engineering, "to walk along . . . the fringe of the disputed territory" (p 205). In effect, he takes the reader on a whirlwind tour of various topics, such as a thought-provoking chapter on open systems, which are attempted standards for operating systems, that, as Clapes describes the issue, are easier visualized than actualized. Clapes also gives good reading with his discussions of *Computer Associates International, Inc. v. Altai, Inc.*,⁹ *Lasercomb America, Inc. v. Reynolds*,¹⁰ and *Atari Games Corp. v. Nintendo of America, Inc.*¹¹ Rounding out Part IV is a chapter of entertaining anecdotes about some notable hackers and some of the people who caught them.

Finally, in Part V, Clapes makes his conclusions and predictions. He observes that "software will be a particularly fertile source for litigation and alternate dispute resolution" for a number of reasons (p. 278). First, the fact that software is difficult to write, but easy to copy and modify, will encourage copiers to test the boundaries of software protection law (p. 278). Although "the legal regimes that define those boundaries have centuries of development that can guide companies in determining right from wrong," continuing litigation is guaranteed, because "there is an unshakable tendency in the industry simply to ignore precedent that doesn't specifically involve software" (p. 278). Moreover, the digitalization of visual and audio media will cause the movie, publishing, broadcast, and music industries, "each more litigious than the software

6. (1989) 15 IPR 1, appeal allowed, No. VG 300 (September 14, 1990), appeal dismissed, No. 92/001 (February 12, 1992).

7. 1219 Hanrei Jihoo 48 (Tokyo D.Ct. 1987).

8. *Allen-Myland Inc. v. International Business Machines Corp.*, 693 F. Supp. 262 (E.D. Pa. 1988) and *Allen-Myland Inc. v. International Business Machines Corp.*, 746 F. Supp. 520 (E.D. Pa. 1990).

9. 775 F. Supp. 554 (E.D.N.Y. 1991).

10. 911 F.2d 970 (4th Cir. 1990).

11. 19 U.S.P.Q. 2d (BNA) 1935 (N.D. Cal. 1991).

industry insofar as intellectual property disputes are concerned," to take stronger interests in the protection of computer programs, which "is bound to increase the number of software-related cases on court calendars everywhere" (p. 279).

The winner of the softwares will depend, according to Clapes, "on the varying comparative advantages of suppliers in different parts of the world and on the evolution of the rules of war in the major countries" (p. 279). As he discussed in the chapter on the formation of an EC directive regarding reverse engineering, both sides have been trying to influence the evolution of those rules and will continue to do so. Clapes notes that a battle similar to the EC debate has recently played itself out in Australia, and others may be imminent in forums both within countries and in international negotiations (p. 287). Although so far it seems that American law is more conducive to a successful software industry than European or Japanese law, by virtue of a more protective regime that nurtures innovation, Clapes suggests there are potential weaknesses in the United States, such as Wall Street shortsightedness and recent trends in the courts (pp. 291-93).

Clapes's conclusion, unsurprisingly, is that the success of a country's software industry depends on a legal regime that gives its innovators the protection they need to reap a fair return. Moreover, since software cannot be distinguished sufficiently from more traditional forms of expression as to warrant different treatment, no attack on software protection can succeed without threatening intellectual property law generally, which "would attract defenders from outside the software industry in such numbers as to preclude its success" (p. 293). In the end, the copiers should lose because the logical conclusion of their argument is "untenable" (p. 294).

Clapes's writing style reminds one of a smart older uncle: he has lots of good stories that he knows he tells well and likes to show off, and he enjoys sprinkling his commentary with references and analogies, some of which are rather obscure. Clapes also has a penchant for puns and plays on words, as even the term "softwars" indicates. Some of these are quite tortured. The best (or perhaps the worst) example occurs in chapter three. First, Clapes explains how Buddhism divided into two schools of thought, the Mahayana and the Theravada (p. 39). The Theravadins believed that Buddhists should be ascetics who achieve Nirvana through their own efforts. The Mahayanists believed that Buddhists should help others achieve enlightenment. One of the great works of the Mahayanists

is the *Lotus Sutra*, which encourages its students to copy it and give it to others. Just as the reader is wondering what any of this has to do with the Silicon Valley, Clapes analogizes the Theravadins with the software originators and the Mahayanists with the software copiers (pp. 39-40). Big deal, the reader shrugs, and forgets the whole thing as Clapes cuts into his explanation of *Lotus v. Paperback* in earnest. Fifteen pages later, after Clapes has shown that he was able, for the most part, to predict the principles announced in *Lotus v. Paperback* he says:

Those principles, written before the *Lotus* decision, also illustrate how predictable the law is. . . . There is no real uncertainty about copyright protection for software. Don't let anyone tell you differently. The *Lotus* case is a long *sutra* that allows those with interests in the computer industry to achieve *satori*, the inner enlightenment that comes—in this case—from realizing that which was already within one's capacity to know. (p. 54)

Ergo, *Lotus Sutra*.

In chapter 13, entitled "The Lady Vanishes: An Academic Ventures into the Real World and Retreats in Dismay," Clapes' playful storytelling goes a bit too far. The title refers to an Alfred Hitchcock movie about a woman named Miss Froy who leads a double life as a governess and a spy and whose disappearance is central to the plot. Clapes uses this reference as a springboard to his story about a law professor who he dubs "Professor Froy." Although the professor is identified in the notes at the end of the book, her real name is never used in the chapter; Clapes refers to her as Professor Froy throughout the story. The professor has vocally opposed the prevailing interpretation of copyright law as to software (p. 193). In 1989, this professor agreed to testify on behalf of a company, Allen-Myland, Inc., in connection with a copyright suit by IBM. It seems that AMI's explanation to her of its activities was less than candid, or the professor, according to Clapes, would have easily seen that AMI had violated copyright law, and she would have refused to testify. As pre-trial discovery and the trial itself unfolded, the truth became more and more apparent. Just before the professor was to testify, she withdrew as a witness, citing scheduling conflicts.

While it seems that the professor acted foolishly at best and the story makes a good yarn, calling the scholar Professor Froy served no useful

purpose. Clapes was not writing a satire with indirect references; the name comes across as a mean-spirited attack on a peer in the legal community that was inappropriate in the context of the book's informational purposes.

Softwars has lots of nifty features to amuse the reader and make it extremely user-friendly. Notes and references have been discreetly tucked away at the back of the book, and there are no footnote numbers littering the text. Each chapter begins with a quotation, often of literary origin, that Clapes uses as a springboard for resuming commentary, including a marvelous 1748 poem by Thomas Gray called "Ode on the death of a favorite cat, drowned in a tub of goldfish" (pp. 245-46). More seriously, at the end of chapters three through nine, Clapes has inserted bite-sized installments of what he terms a "correspondence course" (p. 54) in the economics of intellectual property. Not surprisingly, the thrust of the arguments advanced is that traditional copyright protection is important, even vital, to progress in the software industry and continued benefits to society from software innovation.

As Assistant General Counsel of IBM, Clapes is not unbiased on the subject of software protection. Moreover, Clapes is an advocate—a very, very good advocate—for traditional copyright protection of software. Although he readily acknowledges this bias and overall avoids being dogmatic, the reader seeking information about the software protection debate needs to take *Softwars* with a grain of salt. For example, Clapes never discusses the length of time that copyright protection is generally in force, which is much longer than the term for patent protection and is a nontrivial factor in the economics of the fast-paced software industry.

At the same time, it is obvious that Clapes has been in a position to observe first-hand the evolution of the law of software protection. *Softwars* allows Clapes to share a wealth of experience and observation, including several memorable stories about various people and companies who have played pivotal roles in the softwars. Moreover, readers are fortunate that Clapes is able to deliver this information with an accessible and colorful style. While *Softwars* should not be taken as the final word on software protection, it is a terrific primer for the curious and concerned.

Karen H. Kim