Santa Clara High Technology Law Journal

Volume 14 | Issue 2 Article 9

January 1998

Rescuing Reverse Engineering

Barak D. Jolish

RESCUING REVERSE ENGINEERING*

Barak D. Jolish†

I. INTRODUCTION

A good scientist is a person with original ideas. A good engineer is a person who makes a design that works with as few original ideas as possible. There are no prima donnas in engineering.[1]

Reverse engineering is the art of the follower, the free-rider, the copycat, and, sometimes, the thief. In reverse engineering one disassembles another person's product to learn how it works and to appropriate its ideas. It is a shortcut around the countless hours and substantial costs of product development and testing otherwise required to achieve a market-ready good. Yet, reverse engineering is also an accelerator of progress, allowing competitors to produce compatible goods or improve upon existing designs. This interaction rewards the public with a constant barrage of newer, better and cheaper products.

In weighing these considerations, the Supreme Court has historically regarded reverse engineering as a positive force that is integral to U.S. intellectual property law and policy. In recent years, however, the courts have struggled with cases involving the reverse engineering of software, for two reasons: (1) such engineering necessarily requires the production of an intermediate copy of the program, a *prima facie* copyright violation and (2) software producers often impose license-based prohibitions on reverse engineering. Though the courts have yet to definitively resolve these issues, it appears from cases like *Sega v. Accolade*[2] and *ProCD, Inc. v. Zeidenberg*[3] that software manufacturers may well be able to prevent the reverse engineering of their products. Such a result would undermine tradi-

* Copyright © 1998 Barak D. Jolish.

† All errors and omissions remain those of the author.

1. Freeman Dyson, physicist and author, in DISTURBING THE UNIVERSE § 1 (1979).

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

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

tional copyright and patent law, and harm the market and the public interest. It can and should be avoided by an act of Congress, the Supreme Court or the Executive Branch.

II. THE REVERSE ENGI NEERING OF SOFTWARE

Software is big business — in 1996 U.S. personal computer application sales alone exceeded $10 billion.[4] In this fast-moving and competitive market, software developers often use reverse engineering to study their rivals' products.[5] These developers then use this information to create compatible products, or to develop their own software.[6] Many in the industry disagree, of course, as to whether the practice of reverse engineering is a positive phenomenon or instead a form of piracy that discourages research and innovation. Ultimately, however, most software publishers have tried to use legal obstacles to prevent the reverse engineering of their products; for example, claiming copyright and license protections.

A. The Sega *Case and Copyright Law*

The first case to directly address the collision of reverse engineering and copyright law was *Sega Enterprises Ltd. v. Accolade, Inc.*[7] Plaintiff, Sega, manufactured and marketed a popular video game console and proprietary game software. Embedded in each game's object code was a "lock and key" device which caused the console to recognize and play only Sega cartridges. Defendant Accolade purchased several publicly available Sega cartridges, and reverse engineered their code in order to find the "key." Accolade then employed this key in its own game cartridges for the Sega console. Sega sued, alleging that Accolade's intermediate copying constituted copyright infringement. The Ninth Circuit held that, though interme-

4. *See* Software Publisher Association Press Release, *1996 Personal Computer Application Software Sales Pass $10 Billion for the First Time* (visited February 8, 1998) <http://www.spa.org/research/releases/1996NA.htm> (representing a 8.3% rise over the 1995 total). (The content located at this URL is available at the JOURNAL office).

5. *See* Robert R. Lech, *Protecting Computer Software Against Reverse Engineering*, 73 MICH. BUS. L.J. 526, 527 (1994).

6. The use of reverse engineering does not protect a developer from an action for copyright or patent infringement *in his or her end product.* RAYMOND T. NIMMER, THE LAW OF COMPUTER TECHNOLOGY §1.18[3] (2d. ed. 1996). This essay will not, therefore, deal with questions of substantial similarity and infringement in products which competitors produced after reverse engineering.

7. Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510 (9th Cir. 1992).

diate copying could infringe a software copyright, Accolade's actions constituted a protected fair use.

The *Sega* case analysis, as to market effect, is subtle. The court ruled that since Accolade's copying was intended to discover the code required for compatibility with Sega consoles, the relevant area of competition for statutory analysis was only the "lock and key" mechanism. The court explained, "it is the characteristics of the game program as experienced by the user that determine the program's commercial success [and] there is nothing in the record to suggest that Accolade copied any of those elements."[8] This analysis raises the most serious questions as to the scope of the fair use exception pronounced in the case. If Accolade had reverse engineered to learn how Sega games achieve their pacing, the court might well have ruled that the relevant area of competition was "the game program as experienced by the user." Thus, Accolade's reverse engineering would be aimed at a factor in which it competes with Sega. Would the court see public benefit in allowing others to emulate the stylistic elements of Sega's games?

The court does, however, finally note that underlying its entire analysis are the public policy goals of copyright law. The law strives to achieve a balance "between the benefit the public will derive if the use is permitted and the personal gain the copyright owner will receive if the use is denied . . . [t]he less adverse effect that an alleged infringing use has on the copyright owner's expectation of gain, the less public benefit need be shown."[9] Following this logic, the Second Circuit found that in attempting to monopolize the game cartridge market, Sega was itself undermining the "statutory purpose of promoting expression."[10]

III. Impacts of Prohibitions on Reverse Engineering of Software

Restrictions on reverse engineering extend the scope of trade secret protection beyond its traditional bounds. No longer a tool to protect closely held, well guarded secrets, the doctrine today applies to literally millions of software programs circulating freely around the world. Copyright, a doctrine meant to protect only expression, is similarly stretched to cover the ideas in program code. This hybrid of protection threatens the carefully crafted balance which the

8. *Id.* at 1523.
9. MCA, Inc. v. Wilson, 677 F.2d 180, 183 (2d Cir. 1981) (internal citations omitted).
10. Sega v. Accolade, 977 F.2d at 1523-24.

authors of federal copyright, patent and trade secret protection worked to forge.

A. Harm to the Market

Prohibitions on reverse engineering would harm the market by raising the cost of products and reducing the number of choices available to consumers. In examining this question "a Chicago school economist might suggest that the original creator will grant . . . a license [to reverse engineer] to the competitor if and only if the value of the new use exceeds the cost, and therefore that a failure to license is conclusive evidence that the competitor's proposed use would be inefficient."[11] This approach is inaccurate for two reasons. First, countless non-economic factors play a role in the decision to license a product. Secondly, this approach fails to account for the externalities of an enhanced consumer surplus scenario.[12]

Ultimately, prohibiting reverse engineering would stifle the proliferation of ideas, bounty of products, and hearty competition — all of which are the very reasons courts have supported the practice of reverse engineering.

IV. STEPS TO RESOLVING SOFTWARE REVERSE ENGINEERING OBSTACLES

A. Legislation

The simplest way to eliminate the confusion regarding reverse engineering would be to pass legislation granting it specific protection. In fact, Congress included just such, *sua sponte*, via an explicit reverse engineering defense in Section 906(a) of the Semiconductor Chip Protection Act of 1984.[13] Alternatively, as some have suggested, Congress could codify reverse engineering as a fair use exception.[14] Congress should explicitly assert that reverse engineering

11. Mark A. Lemley, *The Economics of Improvement in Intellectual Property Law*, 75 TEX. L. REV. 989, 1057-58 (1997).

12. *Id.*

13. 17 U.S.C. § 906(a) (1994). The Act expressly permits copying a protected maskwork for the purposes of teaching, analysis, or evaluation.

14. *See* Allan M. Soobert, *Legitimizing Decompilation of Computer Software Under Copyright Law: A Square Peg in Search of a Square Hole*, 28 J. MARSHALL L. REV. 105-107 (1994) (proposing statutory exception to copyright law that permits limited reverse engineering of computer software). Congress, however, seems reluctant to make such a modification to the Copyright Act, 17 U.S.C. § 101-1101 (1994), even though it is aware of the uncertainty surrounding reverse engineering. The legislative history of 1992 amendments to criminal copyright penalties under the Copyright Act states that "[t]his legislation should not be construed

is a right that cannot be preempted by contract or state law in general, and thus eliminate the existing state of confusion through legislative action.

B. Judicial Action

In addition, courts could use fair use principles to disallow reverse engineering restrictions. Since the public interest is a factor that continually informs this area of the law, it would be well within the Supreme Court's discretion to uphold a wide interpretation of *Sega*, thereby extending a more robust protection to the reverse engineering of software.

C. Regulatory Action

Furthermore, the Antitrust Guidelines for the Licensing of Intellectual Property of the U.S. Department of Justice and the Federal Trade Commission strive to "promote innovation and consumer welfare by prohibiting certain actions that may harm competition with respect to either existing or new ways of serving consumers."[15] As the *Sega* court noted, if reverse engineering was automatically branded an unfair use, the copyright owner would hold a virtual monopoly on the functional aspects of a work.[16] Thus, restrictions on reverse engineering should raise the ire of government regulators, especially when the software in question already holds a large share of the market.

V. CONCLUSION

The reverse engineering of software is essentially similar in purpose to the reverse engineering of toasters, tanks or boats. It is, therefore, unreasonable to apply to software an entirely different set of rules with regard to the practice — especially if these rules undermine the purpose and function of traditional copyright and patent law.

---

by the courts as expressing Congressional intent on the question of whether reverse engineering is or is not a civil violation of copyright law." H.R. REP. NO. 997, 102d Cong., 2d Sess. 5 n.15 (1992).

15. U.S. DEPT. OF JUSTICE AND THE FEDERAL TRADE COMMISSION, ANTITRUST GUIDELINES FOR THE LICENSING OF INTELLECTUAL PROPERTY 2 (1995).

16. Sega v. Accolade, 977 F.2d at 1522.