Software Lock-In and Antitrust Tying Arrangements: The Lessons of Data General, 5 Computer L.J. 329 (1984)

Charles H. Helein

Recommended Citation

SOFTWARE LOCK-IN AND ANTITRUST TYING ARRANGEMENTS: THE LESSONS OF *DATA GENERAL*

by CHARLES H. HELEIN*

TABLE OF CONTENTS

Analysts of the computer industry have for some time acknowledged that software is the "new driving force" of the high tech revolution.[1] No longer is hardware considered to be "where the action is."[2] "Now," according to one national journal, "the computer wars are being fought on a new battleground: software—the instructions that tell computers how to do everything from processing payrolls to playing video games."[3] In addition to analysts, members of the computer industry

---

also have long recognized software's enormous potential market power.[4]

Judicial sensitivity to this power has now been reflected dramatically in decisions of the Ninth Circuit Court of Appeals. On June 7, 1984, that court held in *Digidyne Corp. v. Data General Corp.*[5] that the defendant violated antitrust laws by limiting the licensing of its software to purchasers of its hardware.

The purpose of this Article is to discuss the technological advances in software which made this power possible, and to critique the one case in which an appellate court has dealt with the problem of applying traditional legal principles to this aspect of technological innovation. At the same time, it must be recognized that technological advances in software development suggest that the software-hardware marketing strategy confronting the court in the *Data General* litigation may soon disappear. The key issue raised by the *Data General* case then, is whether the application of traditional antitrust principles governing tying arrangements are sufficiently flexible to permit effective competition, or whether the rapidly changing developments in computer software will be impeded by an overly cautious reliance on traditional analysis of antitrust tying arrangements.

I.   THE *DATA GENERAL* CASE

A.   FACTUAL BACKGROUND

Data General, a manufacturer of minicomputers and microcomputers, designed, manufactured and marketed a computer system that included a central processing unit ("CPU") under the trademark "NOVA."   Data General also developed machine-specific operating software called "RDOS" to run the NOVA CPU. Several corporations competed with Data General by manufacturing CPUs, modeled after the NOVA CPU, which were also compatible with the RDOS operating software.

Aware of the competition's hardware, and unwilling to give its competitors what it considered to be a "free ride" on its software development costs,[6] Data General refused to license its RDOS to anyone using a CPU other than its NOVA hardware.[7]   It did so by making RDOS available only under a "Program License Agreement" that limited the use of its supplied software almost entirely to CPUs produced by Data

---

4. "Hewlett-Packard Co. was one of the first major computer makers to recognize that in order to keep its equipment sales growing, it would have to move strongly into software. 'As a hardware maker alone, we couldn't survive,' says Edward R. McCracken, general manager of HP's business development group." *Id.* at 82.

5. 743 F.2d 1336, 1347 (9th Cir. 1984).

6. *In re* Data Gen. Corp. Antitrust Litig., 490 F.Supp. 1089, 1122 (N.D. Cal. 1980).

7. *Id.*

General. Data General further required its software licensees either to purchase a minimum amount of Data General's hardware (a "minimum equipment configuration" or "MEC") or to pay a license charge. Data General's competitors alleged that these policies—the imposition of little or no charge for the software license when a minimum amount of hardware was purchased, and the prohibition of use of the licensed software on non-Data General products—violated federal antitrust laws.

B. PROCEDURAL BACKGROUND

For over six years, the events surrounding *Data General* have spawned a host of vigorously litigated antitrust cases involving the computer industry.[8] In March, 1977, Data General sued Ampex Corporation in New Jersey, alleging that Ampex had used trade secrets—disclosed by Data General to enable Ampex to manufacture memory units for Data General—to allow a third party and Ampex to compete against Data General.[9] In June, 1978, Digidyne Corporation filed a suit in California, charging that Data General misrepresented to Ampex and other Digidyne customers that Digidyne had used trade secrets of Data General in connection with the design and manufacture of a Digidyne computer.[10] By subsequently amending this complaint to allege violations of the federal antitrust laws (specifically, unlawful tying arrangements in violation of section 1 of the Sherman Act and section 3 of the Clayton Act),[11] Digidyne created the framework within which complicated liti-

---

8. *Id.* at 1097. The court described these actions as follows:

Three actions were filed originally in the Northern District of California: Fairchild Camera & Instrument Corp. v. Data Gen. Corp., No. C-78-2418 (N.D. Cal., filed Oct. 1978); SCI Sys., Inc. v. Data Gen. Corp., No. C-78-2417 (N.D. Cal., filed Oct. 1978); Digidyne Corp. v. Data Gen. Corp., No. C-78-1261 (N.D. Cal., filed June 1978).

During 1979 and 1980, the Judicial Panel on Multidistrict Litigation transferred an additional eight actions to the Northern District of California for coordinated pretrial proceedings: Bytronix Corp. v. Data Gen. Corp., No. 78-3832-RF (C.D. Cal. filed Oct. 1978); Data Gen. Corp. v. Ampex Corp., No. 79-1247 (D.N.J., filed Aug. 1978); Data Gen. Corp. v. Ampex Corp., No. 77-0636 (D.N.J., filed Mar. 1977); Data Compass Corp. v. Data Gen. Corp., No. 79-1784-AA (C.D. Cal., filed Feb. 1979); Data Gen. Corp. v. Data Nat'l Corp., No. 78-2869-K (D. Mass., filed Nov. 1978); Data Gen. Corp. v. Ampex Corp., No. 79-1342-MRP (C.D., Cal. filed Apr., 1979); Data Gen. Corp v. Ampex Corp., No. 79-193-T (D. Mass., filed Jan. 1979); Keronix v. Data Gen. Corp., No. 2-80-1217 WHO. *See generally In re* Data Gen. Corp. Antitrust Litig., 470 F.Supp. 855 (J.P.M.D.L. 1979) and 510 F.Supp. 1220 (J.P.M.D.L. 1979).

9. Data Gen. Corp. v. Ampex Corp., No. 77-0636 (D.N.J., filed Mar. 1977).

10. Digidyne Corp. v. Data Gen. Corp., No. C-78-1261 (N.D. Cal., filed June 1978).

11. *In re* Data Gen. Corp. Antitrust Litig., 490 F.Supp. 1089, 1099 (N.D. Cal. 1980). Section 1 of the Sherman Act, 15 U.S.C. § 1 (1982), states in pertinent part:

§1. Every contract, combination . . . or conspiracy, in restraint of trade or commerce . . . is declared to be illegal. . . .

Section 3 of the Clayton Act, 15 U.S.C. § 14 (1982), states in relevant part:

§3. It shall be unlawful for any person engaged in commerce, in the course of

gation developed involving eleven separate actions.[12] A simple delineation of this multi-party litigation shows Data General, on one side, separately charging Digidyne, Ampex, Fairchild Camera & Instrument Corp., and SCI Systems, Inc. with misappropriation of trade secrets and related state law violations. On the other side, those companies, joined by Bytronix Corp., Data National Corp. and Data Compass Corp., claimed that Data General violated federal antitrust law by using unlawful tying practices.[13]

After extensive discovery on the antitrust issue, the parties filed cross-motions for summary judgment.[14] The court denied the motions but found certain facts to be uncontroverted under section 56(1) of the Federal Rules of Civil Procedure,[15] and ordered the trial to be limited

---

such commerce, to lease or make a sale or contract for sale of goods . . . for use, consumption or resale . . . on the condition, agreement or understanding that the lessee or purchaser thereof shall not use or deal in goods . . . of a competitor . . . where the effect of such lease, sale or contract for sale or such condition, agreement, or understanding may be to substantially lessen competition. . . .

Tying arrangements may also implicate § 2 of the Sherman Act, 15 U.S.C. § 2 (1982), on the basis of a claim of monopolization or an attempt to monopolize. In the tying context, a § 2 claim would most likely arise where two products (such as memory and CPU) are "bundled" (sold as a pair), and priced so that purchasing only one of the paired products is more expensive than purchasing both, making the bundled package the customer's only viable option. Such a package would result in a prohibitive leveraging from one market into the other. *In re* IBM EDP Devices Antitrust Litig., 481 F.Supp. 965, 974 (N.D. Cal. 1979), *aff'd sub nom.* Transamerica Computer Corp. v. IBM, 698 F.2d 1377 (9th Cir. 1983), *cert. denied,* 104 S. Ct. 370 (1983). Greyhound Computer Corp. v. IBM, 559 F.2d 488 (9th Cir. 1977), *cert. denied,* 434 U.S. 1040 (1978).

12. *In re* Data Gen. Antitrust Litig., 490 F.Supp. at 1099. *See supra* note 8.

13. *In re* Data Gen. Corp. Antitrust Litig., 490 F.Supp at 1099. These actions also raised a variety of other allegations. For example, the *Data Compass* filing charged that Data General had breached a computer purchase contract between the parties. Data General alleged, *inter alia,* that Data National had breached its Original Equipment Manufacturer (O.E.M.) agreement with Data General by misrepresenting that it was in compliance with Data General's certifications.

The California federal district court that heard this complex case broke the litigation into component parts. Investigation of the trade secrets and related issues was postponed in order to focus only on pretrial preparation of the antitrust issues. The antitrust issues were further divided for pretrial and trial purposes into separate determinations of liability and damages. *Id.*

14. *Id.* at 1097.

15. With regard to the plaintiffs contention that Data General's tying arrangements possessed each of the elements of a *per se* tying violation, the court found the following material facts existed without substantial controversy:

a. Data General's operating systems software and its central processing units were separate products subject to a tie-in imposed by Data General's Program License Agreement. *Id.* at 1104-06.

b. Data General's software-CPU tie-in affected a "not insubstantial" amount of commerce in the CPU market. *Id.* at 1116-17.

c. All plaintiffs except Data Compass suffered actual injury of which a material cause was Data General's software-CPU tie-in. *Id.* at 1117-19.

to the issue of Data General's economic power to restrain competition.[16] Fairchild and Digidyne were the only plaintiffs who actually proceeded to trial; the remaining actions were settled prior to its commencement.[17]

When the jury found in favor of the remaining plaintiffs, Data General moved for judgment *non obstante veredicto* or, in the alternative, for a new trial.[18] The court granted the motion, finding that the evidence could not reasonably support the jury's finding.[19] Plaintiffs successfully appealed this holding.[20]

C.   TYING ARRANGEMENTS UNDER THE SHERMAN AND CLAYTON ACTS

The complaints filed against Data General characterized Data General's marketing practice as an unlawful tying arrangement, i.e., an agreement by a seller to sell a product "only on the condition that the buyer also purchase a different (or tied) product, or at least agrees that he will not purchase that product from any other supplier."[21] Tying arrangements in the computer field often have unique characteristics. In *Response of Carolina, Inc. v. Leasco Response, Inc.*,[22] the court noted that two computer products "might be illegally tied through the technological relationship between them." However, such a tying violation

---

d.   There were no legitimate business justifications for the software-CPU tie-in because less restrictive alternatives were available. *Id.* at 1120-23.

16.   The court found that an actual and good faith controversy surrounded the material question of whether Data General possessed sufficient economic power in the operating systems software market to appreciably restrain competition in the CPU market. *Id.* at 1111-15.

17.   *In re* Data Gen. Corp. Antitrust Litig., 529 F. Supp. at 804 n.1.

18.   *Id.* at 804.

19.   *Id.* at 821. The court identified the "threshold legal issue" presented as the degree of market analysis necessary to prove economic power in a tying case. *Id.* at 807. While recognizing that to require market evidence identical to that appropriate in a rule of reason case was to erase the distinction between *per se* and rule of reason analysis, the court nonetheless placed upon the plaintiffs the burden of introducing "sufficient evidence of the tying product market to permit the jury to find that Data General's competitors were prevented from developing competitive software, and sufficient evidence of the tied market to permit the jury to find an appreciable restraint of trade." *Id.* at 809. The court reasoned that such evidence was needed to make the determination required by a recent Supreme Court statement on market definition in tying cases in Fortner Enters., Inc. v. United States Steel Corp., 394 U.S. 495 (1969) ("Fortner I"); United States Steel Corp. v. Fortner Enters., Inc., 429 U.S. 610 (1977) ("Fortner II"), *viz.* whether the seller has some advantage not shared by his competitor in the market for the tying product. *In re* Data Gen. Corp. Antitrust Litig., 529 F. Supp. at 808, 809.

20.   Digidyne Corp. v. Data Gen. Corp., 734 F.2d 1336 (9th Cir. 1984).

21.   *In re* Data Gen. Corp. Antitrust Litig., 490 F. Supp. at 1100. Tying arrangements may also implicate § 2 of the Sherman Act, 15 U.S.C. § 2 (1982), on the basis of a claim of monopolization or an attempt to monopolize. *See supra* note 11.

22.   537 F.2d 1307 (5th Cir. 1976).

would be limited to situations where the "technological factor" which effectively compels the purchaser to take both components "has been designed for the purpose of tying the products, rather than to achieve some technologically beneficial result."[23]

Since 1958, tying arrangements have generally been deemed illegal *per se*.[24] However, unlike other *per se* categories, tying arrangements which establish some business justification may still be judged under the rule of reason approach. That approach allows tying arrangements to exist, at least for a reasonable period of time, where there is sufficiently compelling business reason for the arrangement.[25]

In 1982, the Ninth Circuit reaffirmed the traditional view that a tying arrangement exists only when four elements are present.[26] The first element is the sale of two separate products, where buyer's purchase of one product is conditioned on buyer's purchase of the other.[27] Second, the seller uses this market power (or "leverage") by acting with "some modicum" of coercive behavior towards the buyer.[28] Third, the seller has sufficient economic power in the relevant product market for the tying product.[29] The final element is that the alleged tying arrangement affect an amount of commerce that is "not insubstantial."[30]

In *Data General*, the district court found that the first three of

---

23. *Id.* at 1330.

24. Northern Pac. Ry. Co. v. United States, 356 U.S. 1 (1958).

25. United States v. Jerrold Elec. Corp., 187 F. Supp. 545 (E.D. Penn. 1960), *aff'd per curiam*, 365 U.S. 567 (1961), *reh'g denied*, 365 U.S. 890 (1961). The tying *per se* rule, however, "is exceptional in that it permits the defendant to offer justifications for undertaking the tie . . . . A tie-in may be justified if it is implemented for a legitimate purpose and if no less restrictive alternative is available." Betaseed, Inc. v. U&I, Inc., 681 F.2d 1203, 1215 (9th Cir. 1982), *quoting*, Phonetele, Inc. v. American Tel. & Tel. Co., 664 F.2d 716, 738-39 (9th Cir. 1981), *cert. denied*, 459 U.S. 1145 (1983). *But see*, Metrix v. Daimler-Benz Akg., 1982-2 Trade Cas. (CCH) § 64, 861 (D. Md. 1982) (rejecting a rule of reason approach in fourth circuit tying cases), *rev'd in part*, 716 F.2d 245. (4th Cir. 1983).

26. Hirsch v. Martindale-Hubbell, Inc., 674 F.2d 1343, 1346 (9th Cir. 1982), *cert. denied*, 459 U.S. 973 (1982).

27. *Id.* at 1346-47.

28. *Id.* at 1347.

29. *Id.*

30. *Id.* An important substantive gloss has been added to these traditional tying arrangement elements by the recent Supreme Court decision in Jefferson Parish Hosp. Dist. No. 2 v. Hyde, 104 S. Ct. 1551, (1984). This case held that a tying arrangement is illegal *per se* if the plaintiff can demonstrate that the tying product accords the seller some form of economic leverage to compel the purchase of a tied product which might not otherwise be sold. The extent to which *Jefferson Parish* modifies or undercuts the anlysis of the Ninth Circuit's holding in *Data General* is unclear, particularly in light of the Ninth Circuit's emphasis upon the economic power conferred upon the defendant as a result of the desirability of the RDOS operating system. The Court of Appeals in the *Data General* case cited *Jefferson Parish* in reaching its conclusion.

these elements existed.[31] The only issue was whether Data General possessed sufficient economic power in the relevant tying product market.[32] The Court of Appeals agreed with this determination.[33]

D. General Legal Analysis of Economic Power

A tying arrangement will not be found absent a demonstration of sufficient economic power in the relevant product market. The tasks of defining the product and the geographic market definition are extremely difficult and are often dispositive of the outcome of the trial. In the tying context, usually one of two tests are used to evaluate the sufficiency of economic power in the relevant product market: either the traditional market share analysis employed under section 2 of the Sherman Act, or a determination of the "uniqueness" of the tying product.

1. *Traditional Market Share Analysis*

In *Phillips v. Crown Central Petroleum Co.*,[34] the Fourth Circuit Court of Appeals held that the seller's market power in the tying product market must approximate a ten percent minimum in order for that power to be sufficient to establish a prohibited tie-in under section 1 of the Sherman Act. However, this ten percent figure is not carved in stone; the minimum percentage probably fluctuates depending upon the type of product and the nature of the industry.

2. *Uniqueness*

A second basis for the determination of economic power deals with the purported "uniqueness" of the tying product. Uniqueness has been defined in terms of legal, economic or physical characteristics.[35] Legal uniqueness generally encompasses legislatively created monopolies such as patents or trademarks.[36] However, the presumption of economic power accorded to patents and trademarks has never been extended into the trade secret context.[37]

---

31. *In re* Data Gen. Corp. Antitrust Litig., 490 F.Supp. 1089 (N.D. Cal. 1980). *See supra* note 15.

32. 490 F.Supp. at 1111-15. *See supra* note 16.

33. Digidyne Corp. v. Data Gen. Corp., 734 F.2d 1336 (9th Cir. 1984).

34. 602 F.2d 616 (4th Cir. 1979), *cert. denied*, 444 U.S. 1074 (1980).

35. United States v. Loews, Inc., 371 U.S. 38 (1962).

36. *In re* Data Gen. Corp. Antitrust Litig., 529 F.Supp. 801, 808 (N.D. Cal. 1981).

37. Trade secret protection has never been held to create a presumption of economic power:

> While an owner of a trade secret has extensive rights with respect to those standing in a contractual or confidential relationship to him, he . . . has no rights at all against independent third parties who develop similar or identical matter . . . . Thus, if XYZ first develops program "Dynamite"—a software program—retaining it and licensing it as a trade secret, the law of trade secrets will not prevent

Uniqueness as defined for purposes of determining the existence of economic power also includes any situation where "competitors are in some way prevented from offering the distinctive product themselves."[38] Thus, uniqueness could arise either from the desirability of the specific product to consumers, the product's attributes, or the economic entry barriers to the competitor's development of a comparable product ("economic leverage").[39]

E. JUDICIAL ANALYSIS OF ECONOMIC POWER IN *DATA GENERAL*

In *Data General* the plaintiffs based their proof of market power on the concept of software "lock-in." Software lock-in allegedly occurs when a customer creates applications programs tailored to a particular operating system (such as Data General's RDOS). Because customers must spend millions of dollars to design applications software tailored to one operating system, they become committed to using that operating system; the only way customers can escape the effects of this lock-in is to convert their tailored applications software to another company's operating system's software.

The plaintiffs in *Data General* contended that these conversion costs were prohibitively expensive, effectively foreclosing this option and locking in Data General's customer base, with the result that the locked-in customers insulated Data General from the normal competitive conditions of the market place. Such insulation, it was further alleged, enabled Data General to impose discriminatory terms upon old customers and to demand uniformly higher prices from all customers, even at the cost of losing some new business.[40]

In rejecting the jury's verdict,[41] the district court determined that lock-in was not a source of market leverage. The court determined that

---

ABC—another company—from developing a program . . . serving the same function and meeting the same specifications.

Id. at 817. *See also* Kewanee Oil Co. v. Bicron Corp., 416 U.S. 470, 490 (1974).

38. Warner Management Consultants v. Data Gen. Corp., 545 F.Supp. 956 (E.D. Ill. 1982).

39. United States v. Loews, Inc., 371 U.S. 38 (1962).

40. *In re* Data Gen. Corp. Antitrust Litig., 529 F. Supp. at 814.

41. The court concurred with the jury's definition of the relevant tied product market as consisting of all general purpose minicomputers and microprocessors and with the jury's definition of the relevant tying product market as consisting of all the operating systems software which will run on such hardware. *Id.* at 813-14. The jury's definition adopted neither the defendant's nor the plaintiff's suggested market definition. *Id.* at 809, 811. The court rejected, however, the jury's finding of a submarket defined as "operating software which run with CPUs utilizing NOVA instruction set." *Id.* at 811. In determining the existence of a submarket, the jury adopted the plaintiff's definition of the relevant *general* tying market. *Id.* at 809. The plaintiff offered this definition after initially basing its definition on the principle of software lock-in, which would have included only those types of operating systems software compatible with applications programs written to run

the degree of lock-in depended on a number of variables, the most important being the way in which a customer writes its application program, and the evidence showed the technology to develop conversion aids already existed. In short, the choice between continuing to use software compatible with RDOS or converting to another company's operating system software was a business decision left in each case to the individual entrepreneur.[42] The court also pointed out the virtual impossibility of calculating the degree of actual lock-in experienced or perceived by Data General's customers.[43]

In addition, the lower court dismissed the plaintiffs' efforts to prove Data General's economic power by showing that RDOS was unique in the economic sense, and that this uniqueness gave the manufacturer an advantage over its competitors who were unable to offer the distinctive product themselves. Even assuming that RDOS was unique in that customers viewed it as desirable, and it in fact possessed various features rendering it superior to other software, the court determined that plaintiffs failed to carry their burden of proof as to whether Data General's competitors were prevented from developing "functionally equivalent" software.[44] The court observed that "uniqueness is an indicia of economic power to the extent that the manufacturer . . . may exact a premium, in the form of higher prices or burdensome terms, which could not be exacted in a completely competitive market."[45] Referring to Data General's demonstration that the lock-in did not free it from the price constraints of the general market, the court concluded that Data General lacked the requisite market power, and rejected the jury's verdict.[46]

---

on Data General's software RDOS. *Id.* at 809. The court found no substantial evidence justifying the jury's definition. *Id.* at 818-20.

42. *Id.* at 814.

43. *Id.* The court also noted that Data General's customers, who were predominately original equipment manufacturers ("OEMs"), were very price-sensitive; if Data General overcharged old customers, it would destroy its customer base by rendering those customers unable to compete against other OEMs in the market who were customers of vendors other than Data General. *Id.* at 815. The court further noted that the plaintiff offered nothing to rebut either Data General's contention that it actively sought to attract new customers or to counterbalance the "overwhelming evidence" that Data General's pricing was in fact competitive both in policy and result. *Id.*

44. *Id.* at 816. The court reasoned that legal barriers in the form of copyright notices, while creating a presumption of economic power, did not conclusively prove such power; that the plaintiff's evidence regarded the effect of copyright notices on the development of compatible—not comparable—software; and that no evidence supported a finding that Data General's trade secrets protection created a legal barrier preventing competition. *Id.* at 816-18.

45. *Id.* at 818.

46. In the court's opinion, the plaintiff's failure to prove Data General's economic power in the tying product market necessarily meant that Data General possessed no

The appellate court's reversal was based on the grounds that plaintiffs had established Data General's possession of sufficient economic power with respect to RDOS, the tying product.[47] The appellate court traced most of the district court's reasons for setting aside the jury verdict to the lower court's misconception that the legality of a tying arrangement must be tested by the seller's economic power throughout the market for the tying product, and by the relative substantiality of the restraint on competition in the tied product market considered as a whole.[48] The court further suggested that while possession of such "monopoly" power by the seller would be sufficient to establish illegality *per se*, proof of such power was not necessary to establish illegality.[49] Instead, the proper inquiry in reviewing the jury verdict was "whether the jury reasonably could have concluded [that] defendant's RDOS was sufficiently unique and desirable to an appreciable number of buyers to enable [the] defendant to force those buyers also to buy a substantial volume of [the] defendant's NOVA instruction set CPUs they would have preferred not to buy."[50] The court added:

> [T]he question is not whether other operating systems with which RDOS competed were as good as RDOS or better in the eyes of some buyers, but rather whether RDOS, available only from the defendant, was *sufficiently attractive to some customers* to enable defendant *to require* those who wished to obtain it also *to buy from defendant* NOVA instruction set CPUs they might otherwise have purchased from others.[51]

The Court of Appeals also dismissed defendant's claim because the

---

power to restrain competition in the relevant tied product market. Even had Data General held this power, the court stated, there was no evidence to support the plaintiff's allegation of appreciable restraint in the market for the tied product. *Id.* The court further concluded that, even if a relevant submarket were properly defined, there was an absence of evidence that Data General possessed economic power in that market over the tying product or that Data General's tying practices appreciably restrained competition in that submarket. *Id.* at 820-21. On the basis of this rationale, the court granted Data General's motion for judgment N.O.V. *Id.* at 821.

47. Digidyne Corp. v. Data Gen. Corp., 734 F.2d 1336 (9th Cir. 1984). The reviewing court supported many aspects of the lower court's decision, stating that the district court had properly granted summary judgment in holding that the NOVA-CPU and RDOS were separate products, that plaintiff was damaged in fact by the tie-in, and that no legitimate business considerations offered by the defendant justified the tie-in. *Id.* at 1341.

48. The appellate court explained that not all of the district court's errors could be traced to the misconception about the need for a detailed market analysis. For example, the Court of Appeals pointed out that, although correct in holding that a copyright creates only a presumption of economic power sufficient to render a tying arrangement illegal *per se*, the district court improperly neglected to see that the burden to rebut the presumption shifted to the defendant. *Id.* at 1344.

49. *Id.* at 1345.

50. *Id.* at 1341.

51. *Id* at 1345-46 (emphasis added).

customers were aware of the tie-in at the time of initial purchase, had invested no money in applications software, and could choose freely among competing systems. Instead, the court characterized the lock-in as a magnifier of initial advantage, as evidenced by statistics that revealed that, by 1979, 93% of the defendant's NOVA-CPU sales were made to locked-in customers.[52]

Using this framework in evaluating the jury verdict, the Court of Appeals found "abundant evidence" that defendant's RDOS was distinctively and desirably unique, and that Data General's marketing practice constituted a prohibited tying arrangement.[53]

II. THE NATURE OF SOFTWARE

The appellate court's focus upon the purportedly inherent advantages of RDOS, and its concommitant rejection of the district court's finding that software lock-in did not effectively confer market power upon Data General, flies in the face of the realities of the software marketplace. Software, as with other aspects of the "high tech" revolution, has undergone enormous technological advancement in recent years, advancement which the appellate court either failed to consider or deemed insufficient to justify the district court's holding.

In the antitrust context, the rationale behind, and consequences of, software lock-in require judicial understanding of the fundamental nature of software. Software may be divided into two broad categories: applications software and operating system software.[54] Applications software is the set of instructions that tell the computer to perform a particular function, such as bookkeeping, word processing or inventory control. Operating system software, on the other hand, tells the computer how to "compute." Normally, software developers write applications programs specific to a single operating system. These programs in

---

52. *Id.* at 1342-43.

53. In addition, as proof of the products' uniqueness, the appellate court noted RDOS's copyright and trade secret protection that established for the court both the distinctiveness of RDOS and legal bars to its reproduction by competitors. The appellate court also corrected the district court's conclusion that the plaintiffs failed to prove an appreciable restraint in the market for the tied product. Reiterating that a detailed analysis of competitive conditions in the tied product market is uncalled for under the *per se* analysis, the appellate court stated that "all that is required in respect to the extent of the restraint in the market for the tied product is that a 'substantial *volume* of commerce' be foreclosed, and 'substantial volume' in this context means only an amount greater than *de minimis*." The court believed that this requirement was clearly satisfied here. *Id.* at 1347.

54. Some catagorizations include a third type of software called "utility software," which comprise compilers, data base managers and other integrated environments. *Software: The New Driving Force*, BUS. WK. Feb. 27, 1984, at 74, 76.

an unmodified form are inherently incapable of working with any other operating system. This, in essence, is the concept of software lock-in.

A. THE STRUCTURE OF OPERATING SYSTEMS

A computer's operating system software is analogous to a person's autonomic nervous system which controls simple reflex actions. In simple terms, operating systems consist of a series of instructions that tell the CPU how to function as a computer, i.e., to coordinate information processing and to control peripherals.[55] Microcomputer operating systems may be divided into several categories: single user/single tasking; multi-user/multi-tasking; 8-bit systems; and 16-bit systems.[56]

Operating systems generally consist of numerous "subroutines" which control and direct such hardware-specific functions as creating the video display, reading and entering memory, and accepting input from the keyboard.[57] Conceptually, an operating system may be broken

---

55. Operating systems have been called "traffic cops," "major domos," "virtualized interfaces" or even the "personality" of your computer. They give it "soul" or handle the "housekeeping." The reason behind these metaphorical outbursts is that an operating system does too many jobs for one simple explanation; it is, in fact, easily the most important software a computer possesses.

Cook, *Special Report: Operating Systems*, POPULAR COMPUTING, Aug. 1984, at 111, 112.

56. The nature of single user/single tasking operating systems is self-explanatory. A multi-user system is one in which a single computer processes information for several users simultaneously. The operating system is required to "allocate and share system resources accordingly." Cook, *Operating Systems in Transition*, HIGH TECH., June 1984, at 65.

"Multitasking" or "concurrent" operating systems are designed to allow simultaneous multiprogram processing. "Multitasking tricks each of several programs into believing that it is running on a separate computer." *Id.* at 66. *See also* Schindler, *Mini Operating Systems Adapt to Multiuser Demands*, SYS. & SOFTWARE, May 13, 1982, at 19; McLeod, *Small Business Systems: Systems Get Multitasking, Multi-User Capability*, SYS. & SOFTWARE, June 1983, at 97; Catchpole, *Under Discussion*, BUS. COMPUTER SYS., Apr. 1984, at 13.

Eight-bit operating systems refer to a single character ("binary integer" or "bit"). Eight bits are equivalent to one byte or character of information at a time, while 16-bit systems refer to two characters of information.

Three very common operating systems are CP/M, MS/DOS and UNIX. CP/M is a "Control Program for Microcomputers." MS/DOS is the "Microsoft Disk Operating System." DOS is a generic term utilized by many CPU manufacturers to describe their particular operating system, such as Apple/DOS or Northstar/DOS. UNIX, a system designed by Bell Labs, was specifically designed for multi-users, multi-tasking and telecommunications applications.

Since the use of CP/M and MS/DOS is not limited to a single hardware manufacturer, it is considered a "third party" operating system for which a large library of applications programs have been created. *See generally* Cook, *supra* note 55, for a survey of the characteristics of these major operating systems.

57. Cook, *supra* note 55, at 111-12. *See generally*, S. KAISLER, THE DESIGN OF OPERATING SYSTEMS FOR SMALL COMPUTER SYSTEMS (1983).

down into specific layers of subroutines.

The subroutines "closest" to the hardware are commonly referred to as the BIOS (basic input/output system).[58] The BIOS coordinates and controls the machine-specific parts of the system. Because it is the software which primarily interfaces with much of the peripheral equipment running in conjunction with the CPU, the BIOS operating layer is by nature highly machine-specific and not readily transportable to another machine or machine configuration.[59]

The next layer of the typical operating system is the "kernel,"[60] which is the heart of the operating system. The kernel generally contains instruction sequences such as a "supervisor" routine for managing system operations, a "network" routine for telecommunications functions, and an "I/O" routine controlling the input/output of things such as characters, memory management routines, and graphic interface routines.[61]

The next operating system layer consists of system utilities which are instruction sets that control such functions as the copying of files or the manipulation of memory.[62] This layer also includes programming tools, such as compilers.[63]

There is no theoretical limit to the number of subroutine layers an operating system may contain. The more powerful an operating system is, the richer and more complex its layers will be.[64]

---

58. Digital Research was instrumental in developing the BIOS level by means of its CP/M operating system, which isolated machine-dependent input/output functions from the rest of the operating system. Legg, *Portable Operating Systems Create Common Program Environment*, EDN, Sept. 29, 1983, at 102; Cook, *supra* note 55, at 114.

59. KAISLER, *supra* note 57, at 43. The division of operating systems into "layers" is for illustrative purposes and presents a simplified picture of the structure of operating systems in general. In many of these systems the "layers" overlap. For example, the "device handler" routines generally perform such machine-specific functions as status tracking, device access, device control and space allocation, and thus are part of the BIOS. *Id.* at 44. However, some operating systems, such as MS/DOS and UNIX, allow modification and even removal of the device handler subroutines into another "layer." *Id.* at 43.

60. "Kernel" is a descriptive term used by Cook in his Articles, *supra* notes 55 and 56. Kaisler does not use this term, but simply refers to subroutines for Process Management, Memory Management and Input/Output Management. KAISLER, *supra* note 57, at chs. 3, 4 and 5.

61. Again, the location of many of these routines varies with different types of operating software. Networking software, for example, may be part of several different software "layers." Cook, *supra* note 56.

62. The "utilities" layer of an operating system may also be comprised of independent application programs, such as a data base management system. *Software: The New Driving Force*, BUS. WK., Feb. 27, 1984, at 74, 76.

63. A compiler is an interpreter which translates programming language into machine language.

64. Cook, *supra* note 56, at 67.

B. APPLICATIONS SOFTWARE

Applications software generally may be described as task-specific software directed towards performance of a single function. Conceptually, applications programs sit "on top" of the operating system utilities layer. It is here that the majority of user interface occurs.

Applications programs may be categorized into two general classifications, generic software and user-specific software.[65] Applications software may be characterized further in terms of the kind of information processed. Thus, an important trend in many applications programs is the creation of an "application environment" which functions as a hybrid operating system.[66] Applications software may create the application environment's function by imposing a template over the particular operating system which allows the running of different applications with a single, standardized user appearance.[67]

III. SOFTWARE LOCK-IN AND THE VIRTUAL OPERATING SYSTEM

The concept of "software lock-in" was a major element in the Ninth Circuit's finding that Data General possessed economic leverage in the tying product market. What was perhaps misunderstood by that court was that software lock-in is a simple concept referring merely to applications programs that are written for, and thus necessarily "tied" to, a single operating system. In short, most applications programs are physically unable to "run" on an operating system different from the one for which they were originally designed.[68]

"Virtual" operating systems, however, remove this barrier by allowing a user to run applications programs designed for other operating

---

65. Generic application software is software which performs a generic task, such as bookkeeping or inventory control, with no further specification of the type of business for which the function is to apply. User specific software is generally more customized, e.g., addressing or bookkeeping for a law firm, or inventory control for a doctor's office. Generally, generic applications programs require some additional customization to function properly in a user-specific environment, while user-specific programs require no further tailoring to fit the environment in which it is to be used. *Cf., Software: The New Driving Force*, BUS. WK., Feb. 27, 1984, at 82.

66. Legg, *supra* note 58, at 109-11.

67. An example of an advanced applications environment is Apple Computer's "Macintosh," which presents a series of standardized "icons" to represent a range of different applications functions.

68. Applications programs may become locked-in to a particular operating system for many reasons. Different operating systems may require different file structures, language or operating system "calls." Legg, *supra* note 58, at 104. An additional difficulty is represented by the nonportability of object code. "Most commercially available application software comes only in object form, and object code usually is not portable." *Id.* at 106.

systems.[69] Virtual operating systems place an interpretive layer of software between the machine specific operating system and the applications program running with it. The virtual operating system may work as a translator between the applications program and the operating system by converting the application commands into software understandable to the machine specific operating system with which it is being run.[70] As a result, the operating system "disappears" as far as the user is concerned, and any applications software may be used with any operating system, regardless of which applications software the operating system was designed to use.

Virtual operating systems thus have the advantage of being proprietary to the originator of the system, while at the same time enormously improving the availability of applications to the end user. As a result, by providing the user with entire libraries of alternate applications programs designed for other operating systems software, virtual operating systems eliminate software lock-in as a market factor.

CONCLUSION

Given the powerful competitive advantage which a virtual operating system can provide to a hardware manufacturer, these systems will

---

69. For example, a recently developed virtual operating system is called "Concurrent DOS." Concurrent DOS can run applications programs designed for operating systems running on machines utilizing the 16 bit 8088/8086 family of processors. Holsworth & Guzaitis, *Concurrent CP/M Drives Multiple "Virtual Consoles,"* MINI-MICRO SYS., Mar. 1984 at 231. The main drawback to virtual operating systems is their slow speed. By adding another software "layer" onto the machine, virtual operating systems consume additional computing power. This problem may be partially alleviated by taking a combined hardware/software approach to operating system interoperability. A new machine utilizing this approach was recently introduced by Micro Craft of Dallas. Named the "Dimension 68000," this machine claims to be able to run programs written for the IBM PC family CP/M based computers, the Apple II family and even UNIX operating system. *The Dimension 6800: A Machine For All Software,* POPULAR COMPUTING, Nov. 1984, at 64. Thus, for example, IBM recently announced the unveiling of its multiuser "Popcorn" microcomputer system based upon Intel's 286 microprocesser capable of running PC-DOS and UNIX. *Data Topics,* ELECTRONIC NEWS, July 2, 1984, at 21.

70. A virtual operating system thus works by creating a "virtual" or universal machine defined by the virtual operating system. Legg, *supra* note 58, at 108-09.

The Institute of Electrical Electronics Engineers has organized a standardization committee known as MOSI (Microprocesser Operating System Interface) Task Force 855, to help standardize operating systems so as to expedite the development of virtual operating systems. The MOSI task force has defined seven areas of operating system standardization: memory management, exception processing, program interfacing, process management, system-clock management, I/O, and data management. *Can Operating Systems be Standardized?,* ELECTRONIC DESIGN, May 13, 1982, at SS25; *Operating System Standards,* SYS. & SOFTWARE, May 1983, at 94.

quickly enjoy widespread commercial use.[71] Once available, this operating software will eliminate applications software lock-in as an indicia of uniqueness and hence as a source of marketing leverage. In short, technological change could quickly make much of the reasoning in *Data General* obsolete.[72]

But, to paraphase Holmes, the law is not a crystal; it is, like the skin around a living creature, capable of growth in unison with the organic matter that it shapes.[73] Technology may render the facts upon which a legal decision rests obsolete, but the principles of that decision often remain vibrant. Because of the law's capacity to accomodate change, *Data General* can teach several valuable lessons.

First, the *Data General* case demonstrates the courts' determination to apply traditional precedent in new contexts. The Supreme Court's most recent discussion of antitrust tying arrangement analysis states explicitly the premise upon which the *Data General* litigation implicitly proceeded. "It is far too late in the history of our antitrust jurisprudence to question the proposition that certain tying arrangements pose an unacceptable risk of stifling competition and therefore are unreasonable 'per se'."[74] *Data General* clearly stands for the proposition that computer products, like other industrial goods, are subject to the "cold test of competition."[75] In this context, courts apply the basic prin-

---

71. Cook, *supra* note 56, at 69. In its recent article on the software industry, BUSINESS WEEK framed the trend as follows:

> As the emphasis in the data processing industry shifts to software—and as software companies strengthen their sales, service, and distribution—the big-system makers, too, are scrambling to do more to provide their customers with software. "In the old days, our customer wrote his own application [software]," notes Jon Tempas, vice president for software products at Sperry Corp's Computer Systems operation. Today, he says, "there's an increased expectation for hardware suppliers to provide the complete solution." That means the equipment makers will need to provide more of their own software. Sperry, for example, now writes 95% of the software it sells for its computer line.

Software: The New Driving Force, BUS. WK., Feb. 27, 1984, at 82.

72. Technological change in software, rather than obviating the need for tying analysis in the computer marketplace, may make the need for such analysis more urgent and the analysis itself more complex. Although a software lock-in analysis may not in the future be a viable theory, the competitive advantages inherent in a virtual operating system may constitute product uniqueness that by itself confers a powerful and tempting marketing leverage. For example, the manufacturer who first offers the virtual operating system might "force" the purchase of its hardware by bundling that software with its hardware. Such *de facto* market power may pass scrutiny under the tying analysis because of the apparent absence of two distinct products.

73. "A word is not a crystal, transparent and unchanged, it is the skin of a living thought and may vary greatly in color and content according to the circumstances and the time in which it is used." Towne v. Eisner, 245 U.S. 418, 425 (1917).

74. Jefferson Parish Hosp. Dist. No. 2 v. Hyde, 104 S. Ct. 1551, 1556 (1984).

75. *Id.* at 1550 (citing Times-Picayune Publishing Co. v. United States, 345 U.S. 594, 605 (1953)).

ciple that the law will dismantle arrangements that restrain competition.

Moreover, *Data General* teaches that courts will not be constrained in developing legal analysis fitting their understanding of technological innovation, hence the adoption of a software lock-in theory. Such creative analysis requires that courts educate themselves about technological advances in order that they may better and more accurately evaluate such innovations under traditional legal principles. As the foregoing discussion reveals, such education may prove short-lived. Nevertheless, if the law is to reflect the growth of a high-tech society, the courts must assume the burden of continuing their own technological education.

Data General demonstrates also that even as innovation fuels the technological revolution, the law stands prepared to check abuses in the competitive marketplace brought about as a result of innovation. The success of this effort to preserve competition will ultimately depend upon the courts' capacity to understand the changes occuring, to adjust traditional legal analysis, and to apply tested principles in new contexts with educated restraint. *Data General* evidences the courts' efforts to extend protection under the law to new technology and to promote continuity even as that technology moves society forward.