Chapter 3

Antitrust in Software Markets

Michael L. Katz and Carl Shapiro
University of California at Berkeley

1. INTRODUCTION*

The computer and software sector is a tremendously important and visible part of the economy. It is also a sector in which there have long been concerns about monopolization. In the past, these concerns centered on monopolization by IBM. Today, the concerns are with Microsoft, but in many ways they are the same. IBM was accused of attempting to sabotage industry standards in Fortran; Microsoft is accused of sabotaging JAVA. IBM was accused of predatory product pre-announcements; Microsoft has been accused of employing "vaporware" — the tactic of announcing products before they are ready in order to preempt the market — to undercut its competitors. IBM was accused of bundling functionality into its CPUs to reduce the value of peripheral equipment; Microsoft is battling government lawyers over the bundling of Internet Explorer with Windows 95. IBM was accused of manipulating interfaces and refusing to reveal them to competitors; Microsoft is accused of refusing to reveal interfaces to competitors. Both companies entered into consent decrees with the Department of Justice to settle antitrust charges.

Despite the fact that these issues have been around for decades, some commentators have opined that existing federal antitrust policy is based on outdated economic theory which is inapplicable to modern software markets.

* We would like to thank Ed Dale, Timothy Muris, Steven Salop, and Hal Varian for useful comments and suggestions

Self-styled crusader Gary Reback, for example, has asserted that "[r]ight now the antitrust division is being held hostage by economists,"[1] which he apparently believes is a bad thing. He apparently believes that antitrust economists are trapped in an out-of-date mindset that is valid only for traditional manufacturing and service industries.

We disagree with those who say that antitrust enforcers lack the economic tools to understand software markets.[2] In part this is due to the fact that many of the issues that arise in software markets also arise in other areas. In the first section of this paper, we identify those features of software markets that tend to distinguish them from other markets. While there is no single feature that is unique to software markets, these markets do possess a number of characteristics that collectively make the application of antitrust policy particularly subtle.

The most notable of these characteristic is that software markets often are subject to network effects, whereby the value of a piece of software (e.g., an operating system) rises with the number of other end users who run that same software. These effects arise both because the ability to communicate and share data with others will be greater, and because it is more likely that complementary hardware, software, and wetware (i.e., brain cells) will be available, when there is a large base of users of the software.

Network effects are a form of demand-side economies of scale that lead to positive feedback. The more widely adopted a piece of software is, the more valuable it becomes, and the more users want to adopt it. Likewise, if a computer program is regarded as unpopular, this perception can feed on itself and spell the demise of that product. With positive feedback, the strong get stronger and the weak get weaker. The end result may be the leading product's becoming dominant.

Because they can lead to tipping to monopoly, network effects are important to antitrust analysis. Concern with tipping plays a central role in policy toward mergers, tying, exclusive dealing, and several other practices. The nature of network effects also gives rise to additional dimensions of competitive behavior that must be taken into account: standard setting and compatibility.

While network effects raise additional concerns, we believe antitrust authorities are up to the task. First, there is a large and growing literature on network economics to which we ourselves have been contributors for the

[1] Daly, James, "The Robin Hood of the Rich," *Wired*, August 1997, p. 112.

[2] Whether the legal tools are there, we leave to others.

past fifteen years.[3] We suspect that many critics who decry the inability of "old economics" to guide policy in the "new economy" are not familiar with this literature. More important, antitrust enforcers have experience dealing with network issues in a number of industries, including credit cards, ATMs, floral delivery networks, computerized reservation systems, railroads, airlines, health care, as well as computer hardware and software. We identify a number of antitrust cases involving networks in our discussion below. We find that the application of antitrust economics in those cases largely mirrors its application in other markets. For instance merger policy is fairly coherent, while policy toward tying is not.

Although the theoretical tools to analyze software markets exist, and antitrust enforcers do have a track record, it is also true that economists and lawyers still are learning how to analyze many of these issues.[4] The final question that we address is whether this fact implies that the antitrust authorities should sit on the sidelines while they perfect their craft, or whether the threat of irreversible tipping to monopoly is so severe that action must be taken today.

2. THE FUNDAMENTAL ECONOMICS OF SOFTWARE

Software markets possess several economic characteristics that must be taken into account by an antitrust analysis. In this section, we identify these features in turn and discuss why each poses challenges to the application of antitrust policy.

[3] For early papers on the strategic analysis of networks see Dybvig and Spatt (1983), Farrell and Saloner (1985) and (1986), Katz and Shapiro (1985) and (1986a) and (1986b). For a non-strategic analysis, see Arthur (1989). For a more recent coverage of network effects, see Katz and Shapiro (1992) and (1994). Rohlfs (1974) provides an insightful analysis of network effects in a monopoly setting. Economists have studied bandwagon effects, which are a close cousin to network effects, at least as far back as Veblen (1899).

[4] For one statement of the Justice Department's views towards network effects, see Shapiro (1996a). For a more recent and broader statement of DOJ policy towards high-tech industries, see Klein (1998). See also Economides and White (1994) for an analysis of antitrust and network effects. For a more comprehensive discussion of how network effects affect the law, see Lemley and McGowan (1997).

2.1 Systems and Network Effects

In most instances, a single piece of software is not valuable by itself; it has to be used with other components, including hardware, user training, and other software. Together, these components constitute a system.

In discussing the economics of competing systems, it is often helpful to distinguish between two types of systems. In communications networks, each user owns a single component[5] and these components make up a system that allows the users to communicate with one another. Compatible fax machines are one example. Users of word processing programs who wish to communicate with one another by sharing files constitute another example. In this example, two users are on the same network if their programs can share files, and they are on different networks if their programs cannot share files. To the extent that users wish to share files, the greater the number of users on a given network, the greater will be the benefits of belonging to that network. This positive feedback is what is known as a network effect.

It is important to recognize that network effects can arise even in the absence of any communications network. These effects can arise when a system consists of two distinct components, A and B, both of which are purchased by a single user.[6] For example, A may be the operating system needed to make word processing program B work. Positive feedback arises when an increase in the number of users who adopt component A leads to an increase in the benefits that consumers can enjoy from the purchase of component B. The greater the number of users who adopt a given operating system, for example, the greater the number and variety of applications programs that are likely to be available that can run on that platform. There also may be greater competition in the supply of those application programs. These effects arise when there are economies of scale in the provision of component B, so that a larger market makes additional entry profitable.

Because these situations give rise to positive feedback effects similar to those that arise in communications networks, we refer to these situations as virtual networks or hardware-software networks. We say that two users are on the same network if they adopt variants of component A that can make use of the same component B. Two users of Windows 95 are on the same virtual network, while a user of Windows 95 and an Apple Macintosh user are not.

[5] A component may itself be a system, such as a modem or a fax machine, comprising sub-components.

[6] In a communications network, a user with component *A* wishes to communicate with another user who also owns a component *A*. The two *A* components constitute a system.

The systems and network aspects of software raise a large number of competitive issues. First and foremost is the threat of tipping to a single dominant vendor or technology, and thus the heightened danger of monopolization. Tipping is a direct consequence of positive feedback. Less popular brands can fight tipping through differentiation, but strong network effects may overwhelm these efforts. When tipping is likely, practices like tying, predation, or exclusive dealing that disadvantage a rival can be more pernicious than usual. For example, under some conditions these practices can be used by a firm that is dominant in one market to tip the market for a related product in its direction, even if its variant of that product is an inferior one.

Network effects also result in compatibility being a critical dimension of industry structure and conduct. When two programs can communicate with one another and/or be used with the same complementary system components, they are said to be compatible. With compatibility, there is one big network and tipping to a single variant is impossible.[7] In the absence of compatibility, markets may tip.[8] Hence, the very nature of competition is fundamentally affected. For precisely this reason, the leading supplier of software in a given category has incentives to prevent others from offering compatible products, setting up a natural tension over the control of interfaces. Often, these disputes implicate intellectual property, as in the Lotus v. Borland copyright dispute over spreadsheet programs that went all the way up to the Supreme Court. Antitrust can potentially play a major role in defining the limits on the ability of a dominant firm to block compatibility or extend its power by controlling interfaces.

In the presence of network effects, expectations can become critical because rational buyers will base their choices in part on expected network sizes. Consequently, the drivers of expectations can also play significant roles. These drivers may include installed bases, current product attributes, producer reputations generated in other markets (e.g., IBM and the introduction of the PC), or financial staying power. The tactic of preemptively announcing products before they are ready, "vaporware," is fundamentally about managing expectations.

[7] This is not to say that one brand cannot have a very high market share; it simply states that network effects are not a driver of market shares.

[8] Incompatibility does not have to be complete. As a practical matter, the issue may be the magnitude of porting costs, the speed of an emulation program, or the amount of information lost in transferring data from one format to another.

Network effects also imply that the degree of concentration on the buying side of the market can strongly affect market performance.[9] This influence derives from the fact that network effects can constitute a significant barrier to entry and lead to collective lock-in of an established technology. Consider, for example, entry by a new brand of electronic spreadsheet that is incompatible with existing programs. Each individual user faces switching costs in adopting the new brand (e.g., the costs of learning the new program and the imperfections in transferring data to a new format). Moreover, because of network effects, the attractiveness of the new program hinges on its popularity, presenting a chicken-and-egg problem. A single large user, or a coordinated group of users, can take control and move the market to the new product if it is superior for their needs. Thus, buyer concentration can erode seller power in network industries generally and software specifically. By the same token, uncoordinated buyers can be extremely vulnerable. Even though users are neither directly connected to one another, nor do they communicate, their actions do affect one another. Small users acting alone cannot protect themselves from harm by exercising their own consumer sovereignty. Collective action may be needed, and it may be difficult to organize. We will refer to this as a problem of coordination costs.

Having identified the possibility of lock-in, it is important to recognize that market power can be fleeting in software markets. The list of once-dominant products whose market shares have plummeted is too large to ignore: WordStar, WordPerfect, Lotus 1-2-3, dBase, Paradox, and more.[10]

2.2 Low Marginal Costs Relative to Average Costs

Like many other goods with a large intellectual property component, software may be subject to very large set up costs (i.e., the first-copy costs of writing the code) coupled with relatively low replication and distribution costs. For example, a new software program might cost several million dollars to develop, whether to make one copy or millions. Once the original code has been written, it may cost only a few dollars each to make additional copies, or almost nothing at all to distribute these copies over the Internet. As a result, software production typically is characterized by a situation in

[9] This is an application of the general economic principle that efficiency is enhanced if parties responsible for causing externalities can deal with each other without bearing prohibitive transactions costs.

[10] Even hardware platforms are subject to such reversals, as Apple, Digital Equipment, Silicon Graphics, Atari, and Sega know all too well. Indeed, Liebowitz and Margolis, (1990) question whether network effects ever give rise to serious problems of inefficient lock in.

which the marginal cost of production — the incremental cost of making an additional copy — is very low relative to the average cost of production, which includes the development costs amortized over the number of copies made. This cost structure has three important implications.

First, lawyers and policy makers must grapple with the question of what a competitive market would look like in the presence of these economies of scale. In many areas of antitrust, the competitive equilibrium is held up as the standard against which to judge a proposed practice. To give this standard some bite, one has to define a competitive price. Most commentators apply the notion of price set at cost for this purpose. But which cost, marginal or average?

In the standard textbook model of perfect competition, this is not an issue. In equilibrium, price is equal to both marginal and average cost. But how should this standard be interpreted in markets where marginal cost is always below average cost? [11] There are at least two candidates. One is to regard the competitive price to be marginal cost, and the other is to view competition as forcing price down to average cost. We believe that price equaling average cost is the more sensible interpretation. In the short run, pricing at marginal cost yields prices that guide efficient purchase decisions.[12] But pricing below average cost provides no long-run incentive for a firm to remain in business. And the firm has no incentives to innovate or invest. This is why regulators set prices that allow regulated firms to earn reasonable rates of return even if the resulting prices are above marginal costs. Of course, measuring average costs is not easy, since these include the costs of originally developing and marketing a program, with an appropriate risk-adjusted rate of return on these sunk costs.

A second important consequence of the low marginal cost of software is that it may be efficient to provide additional features and functions to everyone, even those consumers who do not specifically demand them. As we will discuss below, this observation affects the analysis of tying and bundling.

The third major consequence of the relationship between marginal and average costs is that markets will tend to be concentrated. Moreover, when there are large first-copy costs, entry may be difficult both because the

[11] We note that this is a common problem in the application of antitrust to oligopolistic markets, which is of course the typical market structure in which antitrust concerns arise. The starkest examples are perhaps technology markets, where the marginal costs of sharing technical know-how or licensing patents tend to be very low.

[12] For purposes of this discussion, we are abstracting from the issue of whether network effects are in fact network externalities.

entrant has to be concerned about the post-entry equilibrium and because the first-copy costs typically are sunk and thus represent entry risk.

2.3 Software as a Durable Good

Software is a durable good. Software is hardly the first durable product encountered in antitrust, but it is arguably "more" durable than an aircraft or an automobile. Software "wears out" only due to technological change or planned obsolescence, not based on normal wear and tear such as applies for durable equipment. The durability of software is a major factor in the software business and has significant implications for antitrust analysis.

There is the theoretical possibility that a monopoly market for a durable good will behave much like a competitive market. This proposition is known in economics as the "Coase Conjecture."[13] The idea behind the Coase Conjecture is that the monopolist is tempted over time to offer prices closer and closer to cost in order to garner additional sales from those who have not yet made purchases. In its strongest form, the Coase Conjecture says that the monopolist will immediately offer its software at marginal cost. The monopolist is driven to do this because consumers anticipate the monopolist's attempt to work its way down the demand curve, first selling to the highest-value users and eventually offering its product at marginal cost. Even consumers who place a high value on the software will be unwilling to pay high prices if they anticipate rapid price reductions. We are not suggesting that software monopolists in fact sell their products at marginal cost. But the demand facing a software publisher at any point in time is influenced by the sales it has made so far.

There are at least three ways for a software vendor to avoid the Coase "trap" of marginal cost pricing despite the fact that its product is extremely durable. First, there may be a flow of new customers willing to pay relatively high prices for the program. The existence of these consumers reduces the supplier's incentives to cut prices. Second, the software vendor can rent rather than sell its software. Under a rental policy, the supplier's incentives to cut price are again reduced: any reduction in rental rates to attract new consumers will be costly to the supplier because it will have to cut rents to its existing customers. Third, the vendor can continue to improve its product, which can create demand on the part of old customers.

The reliance of successful software publishers on new versions of their programs to generate ongoing revenues implies that software companies, even dominant ones, have strong incentives to improve their products. To some, this fact implies that little antitrust intervention is needed to ensure

[13] See Coase (1972) and Bulow (1982).

that technological progress proceeds apace. Another interpretation is that software companies have incentives to fuel future demand for their products by engaging in planned obsolescence.

Historically, the term "planned obsolescence" has meant the design of products to wear out and thus require replacement sooner than would be technically necessary. For computer software, the concern is not that the software physically wears out, but rather that it no longer functions properly as part of the overall system. Planned obsolescence in the software industry thus revolves around intergenerational compatibility.

Two fundamental patterns arise. The first occurs when a user upgrades one component of her system and thus degrades the performance of another component. An example of this would a user who upgraded to Windows and found that one of her old DOS programs did not function properly with her new operating system. The second pattern is outside the control of a single user and arises when other users upgrade their software so that it is no longer compatible with the single user's software. For example, it is our understanding that Word97 files cannot be read by Word95 without the use of a separate translator. As more and more consumers adopt Word97, Word95 becomes less valuable, pushing Word95 users either to download the (for now) free translator or upgrade to Word97.

2.4 Rapid Technological Change

The personal computer software industry is widely seen as dynamic, fluid, and driven by rapid technological change. Some commentators emphasize the presence of rapid technological change to make two points. First, they assert that, in the face of rapid technological change, current market positions, such as market shares, mean little. Second, rapid technological change is thought to make intrusive governmental policies particularly likely to do harm.

We are sympathetic to the view that the software industry has been driven by impressive advances in technology, and we certainly agree that enormous value has been generated in the software industry over the past twenty years, both for shareholders and for customers. However, we do not believe that technological or structural change is so great that the software industry should be immune from antitrust enforcement.

The personal computer software industry is no longer in its infancy. Personal computer software markets during the 1980s can accurately be characterized as fluid, with new software categories being established and leadership in several important categories changing hands one or more times. In contrast, personal computer software markets in the 1990s have tended to follow a rather different pattern: Microsoft as an absorbing barrier.

This pattern suggests — but does not prove — that network effects are large enough that the lock-in they create may be greater than the force of technological change.

Looking beyond Microsoft, the central role of innovation in software markets focuses attention on the difficult problem of measuring "competitive significance" in dynamic markets. This problem is far broader than the software industry and has been addressed repeatedly in antitrust. As we discuss below, we believe that the resulting principles can be applied directly to software markets.

We also note that long-run industry performance will be driven more by the pace of innovation than by short-run pricing policies, so monopoly power is primarily of concern if it stifles innovation or product variety. This point underlies our belief that it is a mistake to assert that government intervention is unwarranted in markets with rapid technological progress because the costs of mistakes are too large. By the same token, private actions that stifle innovation can lead to significant welfare losses as well.

3. ASSESSING MONOPOLY POWER IN SOFTWARE MARKETS

The assessment of market power plays a key role in the application of antitrust policy. How does one determine the presence or magnitude of market power in software markets in the light of their fundamental economic characteristics?

The prevailing method by which the Courts and the enforcement agencies have assessed market power is primarily structural: define relevant markets, measure market shares, and judge entry conditions into those markets. We adopt this approach as well, but remark below on additional factors worthy of consideration.

3.1 Market Definition

The first step in most antitrust cases is to define the relevant product market.[14] The 1992 U.S. Department of Justice/Federal Trade Commission

[14] Geographic market definition is usually straightforward in computer software cases: since software can easily be transported, the geographic market is at least the entire United States, and often worldwide. Even if different versions are created in different languages, this versioning is usually relatively inexpensive. In any event, software development has largely been an English-language activity to date. In none of the software merger cases reported below was geographic market definition an important factor. We note, however, that software markets may well be national or regional if usage patterns differ across regions and if the network of users generating network effects is primarily regional rather than global.

"Horizontal Merger Guidelines" (the Guidelines) explain how the agencies define relevant markets for the purpose of evaluating horizontal mergers. We follow the Guidelines here, indicating how their market-definition procedures apply to computer software markets.

The agencies place a given product in a relevant product market consisting of the smallest collection of products for which a hypothetical monopolist controlling the entire collection could profitably impose a "small but significant and non-transitory increase in price" (SSNIP) above pre-merger level — in the case of a merger — or the competitive level, for a non-merger analysis.[15] At the heart of the inquiry is the extent to which consumers would substitute other products, were a group of some products to become more expensive.

Several features of software markets systematically affect this market definition process.

First, because software products tend to be highly differentiated, markets typically include "similar" programs, but may exclude more "distant" programs with only a partial overlap in functionality. For example, in the Borland/Ashton-Tate merger, the Justice Department found a market for relational database software, rather than all database software. And the fact that some customers use database programs only for relatively simple operations that can be performed using a spreadsheet did not mean that spreadsheets were in the relevant market. Of course, differentiation is hardly unique to software; breakfast cereals are also highly differentiated.

Second, consumers use applications software in conjunction with specific configurations of hardware and/or operating systems, and often will not purchase new hardware or a new operating system in response to modest shifts in application software prices. In part, this results from the fact that consumers have already purchased the hardware and operating system prior to buying specific pieces of software. But even when consumers concurrently buy hardware and some software, they often have other durable software that is platform-specific, making it costly to switch platforms in response to modest increases in software prices. Thus, application software markets tend to be defined for a given hardware and operating system configuration, or "platform." For example, in the Autodesk/Softdesk deal, the FTC found a market for computer-aided design engines running on Windows-based computers.

[15] Of course comparison with the "competitive price" runs into the problem we addressed earlier: one must be careful how the concept is defined.

Third, due to significant network effects, consumers tend to be reluctant to select programs that are not widely used, even if these rival programs offer comparable, or even superior, functionality to that offered by the leading program(s). Thus, network effects give rise to entry barriers that are central to market definition and the assessment of market power for software.

Fourth, since software is a durable good, software publishers typically must compete against the installed bases of their own users. Were Microsoft to raise the price of Word97, we expect that the main source of elasticity of demand would come from people who would simply continue to use an earlier version of Word instead of buying the new version. In the light of technological progress, the old and new generations represent differentiated products, and the extent to which the old version limits the exercise of market power in the sale of the new version depends on the rate of innovation. Despite the competition between the two, we know of no case in which the agencies included in-place, or "used" software in the market with new software.

Fifth, the combination of durable goods and switching costs creates the possibility of distinct markets for certain classes of customers. In studying the demand for a given software program, it is useful to divide customers into four groups: (a) the installed base of users of that program, i.e., customers who have already purchased some version of the program; (b) the installed base of users of rival programs; (c) "old" consumers who have considered the options available to them and have declined to buy in the past; and (d) "new" consumers who have recently entered the market and have not yet purchased any program in this category

We distinguish among these groups because they present the seller with different challenges. Consider group (a). Once a customer has a well-functioning version of, say, presentation software, that customer will be motivated to buy another version only if the new version offers enhanced functionality, or if the capabilities of the existing version are somehow undercut by changes in other components of the overall system. Consumers in groups (b) and (c) are somewhat like those in group (a), in that the firm will have to offer customers increased value to attract their patronage. An important difference, however, is that price cuts can be used to induce sales to consumers in groups (b) and (c), but not group (a). For group (b), the firm faces the additional challenge of overcoming consumers' switching costs and brand preferences. Consumers in group (d) have not yet locked themselves in to a particular brand, and the firm's task is to offer the best option to the buyer on a going-forward basis.

Because of differences in the demands of these different groups of consumers, a software vendor has clear incentives to differentiate its

offerings to them.[16] These price discrimination tactics can affect market definition, as specifically addressed in the Guidelines under the rubric of "price discrimination markets." For example, if the consumers who already have purchased a firm's software can be targeted with special upgrade prices, then there can be a separate market for sales to the installed base of that program. Or, in the context of merger analysis, there may be a separate market for sales to current non-users, in which the merging firms compete directly, even if they have little ability to capture sales from each other's installed base, due to high switching costs.

Sixth, the very high gross margins of 80 percent or 90 percent that are common in computer software affect the calculus of market definition in a subtle way. With a high gross margin, each unit sale lost due to a price increase entails a significant reduction in profits. This consideration alone leads to relatively broad markets, since even modest lost sales will make a SSNIP unprofitable. However, these same high gross margins tend to go hand-in-hand with inelastic demand, which implies that few unit sales would in fact be lost were prices to rise.

Finally, the dynamic nature of software markets means that the market definition exercise itself must be forward-looking. For example, some would argue that markets for applications software were transformed quite quickly from a series of markets for word processing, spreadsheet, database, and presentation software into a new market for suites. Market boundaries shift over time along with technology and marketing practices.

3.2 Market Shares

Once the relevant market has been determined, the next step in antitrust cases typically involves identifying the market participants and measuring market shares.

The list of market participants can in principle include firms not currently offering any product in the market. Under the Guidelines, such firms can still be included in the market as "uncommitted entrants" if they would likely offer the relevant product within a year, and without significant sunk costs of entry or exit, in response to a SSNIP. Such supply responses have not played a major role in the software mergers we discuss below. For example, in the Microsoft/Intuit deal, while it seemed entirely possible for companies

[16] Customers may be further segmented depending upon their use of complements, including computer hardware and other software programs. Note that there are reasons to discount to each group, making it impossible to state in general which group will be offered the lowest price: current users can simply refrain from upgrading, users of a rival program must be induced to bear switching costs, and current non-users have signaled a low willingness to pay for the product category overall.

not then selling personal financial/checkbook software to develop and offer products within one year, doing so would have involved non-trivial sunk costs. It appears to be rare for the supplier of one piece of software to be able to modify it easily to perform the functions of software in another distinct category. Consequently, only firms currently offering products, or about to introduce products already developed, typically have been counted as market participants.[17]

With markets defined and suppliers identified, market share can be measured. It is necessary to select a basis on which to do this. There are at least two dimensions of choice. First, in measuring shipments of computer software, one can use unit shipments or dollar sales. To the extent that a firm's share is greater using the dollar sales measure, this indicates that the firm's products are positioned towards the high end of the market. Since users typically desire a single copy of software per computer, unit sales tend to be a better measure of share than dollar sales.

Second, there is a choice of cumulative or current sales. The use of products' installed bases (i.e., number of users of each brand of software) is somewhat like looking in the rear-view mirror: it measures previous sales. We believe, however, that this measure can be highly informative when network effects are pronounced, because the size of the installed base directly affects the future attractiveness of each piece of software. We note in this regard that, if possible, it is desirable to measure "active installed base," i.e., the number of buyers actually using each product. Thus, it may be sensible to remove from the installed base customers who purchased an earlier version years before and have never upgraded, if such customers generally are thought not to be using the program any more. Similarly, we favor removing customers who have a current version, but are not actually using it.[18] This case can arise when consumers receive the software for free as part of a promotion or if it is included as part of a package.

An alternative to measuring the installed base is to measure each program's current shipments. Current shipments directly reflect the current market attractiveness of each program. Thus, market shares based on current shipments tend to be more forward-looking than the installed-base

[17] This is not to say that entry barriers into software markets necessarily are high. Firms that can and would enter within two years in response to a SSNIP are included in the entry portion of the analysis, rather than the "supply-side substitution" portion.

[18] Browsers present an interesting example in which it is possible to measure "active installed base" by tracking how many "hits" at Web sites involve each brand of browser. Direct measures of usage are highly pertinent in assessing the significance of network effects, especially those arising in communications networks.

measure, and thus more informative for merger analysis.[19] A discrepancy between a program's share of the installed base and its share of current shipments reflects a shift in the market.

How are we to evaluate market shares in the software industry? According to the Guidelines, a Herfindahl-Hirschman Index (HHI) above 1800 is "highly concentrated."[20] Using this label, one would conclude that most software markets are highly concentrated.[21] After all, a software market with four equal-sized firms, and thus an HHI of 2500, would be so classified. An open question is whether, at least for mergers, the software industry should be thought of like the automobile and chemical industries, where mergers between two companies with shares of 15 percent to 20 percent each are borderline antitrust violations, or the defense industry, where mergers between two of three remaining suppliers are both commonplace and often encouraged by the leading customer, the Pentagon.

Critics say that market shares tell us little, especially in the presence of rapid technological change. We believe that market shares can be informative, but agree that current market shares should be given less weight if shares tend to shift markedly over time. A large historical market share is less meaningful if predictable future events in the industry will open up competition to new players or neutralize the advantages of the incumbent. What really matters in assessing competition in dynamic markets are the assets that various firms bring to future competition, and market shares "matter" only to the extent that they reflect control over such assets.

In any event, both the agencies and the Courts recognize that market shares are the starting point for the analysis, not the end of the story. Indeed, the Guidelines quite explicitly state that historical market shares can fail to measure future competitive significance if market conditions are changing, e.g., due to shifts in technology. A company with a recently acquired 70 percent share in a software category where lock-in and network effects are minimal has far less market power than a company that has enjoyed this same share for several years in a market subject to strong network externalities and high consumer switching costs. Moreover, in dynamic

[19] The size of the installed base enters more directly in the analysis when looking at conduct designed to exploit the installed base as an asset, *e.g.*, by controlling interfaces necessary to access the installed base.

[20] The HHI is calculated by adding up the squared value of each firm's market share. The HHI ranges from zero, with many small firms, to 10,000 under pure monopoly. An HHI below 1000 is considered "unconcentrated."

[21] This should not be so surprising given the nature of costs and the presence of network effects.

markets, the presence or absence of entry barriers is at least as important as any current snapshot of market concentration.

3.3 Entry Conditions

In our experience, the assessment of entry conditions tends to be the most critical and contentious aspect of software antitrust cases. This is especially true in software mergers. To offer a caricature: The merging parties argue that markets are incredibly fluid, and any attempt to impose supracompetitive post-merger price increases would be self-defeating due to massive, rapid entry by hoards of programmers eager to introduce their powerful products into the market. In contrast, the enforcement agencies point out that market shares have in fact stabilized in recent years, that some products have dropped out after failing to attract a following, that pricing is driven by direct competition among the active brands rather than the supposed threat of entry, and that entry barriers are in fact quite large, based on consumer switching costs, intellectual property rights (IPRs) over interfaces, and network effects.

We reject the categorical statement that entry is so easy into software that antitrust should stay away. Today's software markets involve large, valuable, and entrenched installed-bases of users, extensive data that may be difficult to transfer to work with a new brand of software, and integration among various types of software. If entry were so easy, and installed bases so insignificant, it is unlikely we would see such large market capitalization of today's software companies.

We do not mean to suggest that entry barriers into software market are impenetrable. Our point is more modest: these entry barriers are non-trivial, and worthy of evaluation just as are barriers into other differentiated-product markets, from breakfast cereals to consumer electronics to medical equipment. The software industry should not regard itself as unique is this respect.

3.4 Rules for Assessing Software Monopoly Power

What does all this imply for antitrust in the software industry? We can distill several principles to apply as we turn next to specific areas of antitrust policy. The key lessons are these:

1. The traditional process of defining relevant antitrust markets can be successfully applied to the software industry.
2. Historical market shares are of limited value if they are prone to shift rapidly over time.

3. Price-cost margins are of limited usefulness as a measure of monopoly power in software markets.
4. Because software is a durable good, even software monopolists have some competition — themselves.
5. Due to the presence of switching and coordination costs, and network effects, barriers to entry into software markets can be high despite rapid technological change.

4. SOFTWARE MERGERS

The ongoing dispute between the U.S. Department of Justice and Microsoft over the bundling of the operating system and web browser has generated intense public interest. The fact is, however, that the vast majority of government antitrust enforcement actions in the software industry have involved mergers, not unilateral conduct such as that addressed in the Microsoft consent decree. To show how the underlying economic characteristics of software enter into the analysis, and to illustrate how competition in software markets is assessed in practice, we now explore in some detail a number of software mergers that have been challenged by the Federal Trade Commission and the Justice Department in the last several years.

4.1 Applying the Merger Guidelines to Horizontal Software Mergers

Before looking at specific mergers, we first discuss how the Guidelines, which apply to all industries, are implemented in software markets, in the light of the fundamental economic characteristics of software markets that we have enumerated above. We have already discussed how markets are defined, shares are measured, and entry conditions are assessed in software markets. This leaves estimating the likely competitive effects of a proposed software merger. Following the Guidelines, we consider three types of competitive effect: unilateral competitive effects, coordinated competitive effects, and synergies.

4.1.1 Unilateral Competitive Effects

The focus of merger enforcement in the software industry has been on unilateral competitive effects: the danger that the merged firm, acting independently of any remaining rivals, would find it profitable to raise its

prices after the merger. This concern is founded on economic theory which demonstrates that there is quite generally an incentive to raise prices following the consolidation of rival brands[22]

Two primary factors determine the magnitude of these unilateral competitive effects (at least until we factor in entry and product repositioning below): the gross margins for the merging brands, and the diversion ratio between those two brands.[23] The diversion ratio from brand 1 to brand 2 measures the fraction of sales lost by brand 1 when its price is raised that are captured by brand 2. Unilateral competitive effects are greatest when gross margins are high and when the diversion ratio is high.

This line of reasoning indicates that software mergers can indeed lead to significant unilateral competitive effects, unless entry is relatively easy. As noted earlier, gross margins in software tend to be very high; while not themselves indicative of monopoly power, high gross margins do exacerbate concerns about post-merger price increases. Diversion ratios can be high as well if there are very few popular brands, or if the merging programs are especially "close" in product space.

To evaluate unilateral competitive effects in software markets properly, it is important to factor in consumer switching costs. Switching costs tend to be a two-edged sword in merger analysis. They make it more difficult for one firm (an incumbent or entrant) to win converts from the other firm's installed base; yet any such victories are more profitable because the new customers then become locked in.

4.1.2 Coordinated Competitive Effects

Coordinated competitive effects — including the danger that a cartel will successfully form in the industry — historically were the focus of merger enforcement policy. The current focus of antitrust analysis in high-tech industries such as computer software has been on unilateral competitive effects. The theory is that price fixing is difficult for differentiated products, especially when much of the competition takes place on non-price terms such as continual product innovation and improvement. To the best of our knowledge, the agencies did not focus on a collusion theory in any of the merger cases discussed below.

[22] This is true whether the firms engage in pricing competition or quantity competition. See Davidson and Deneckere (1985) and Farrell and Shapiro (1990) respectively.

[23] See Shapiro (1996b) for an accessible treatment of gross margins and diversion ratios in merger analysis. See Werden and Froeb (1994) for a more extensive analysis using the "logit" model of demand.

4.1.3 Merger Synergies

The strong production and demand-side economies of scale present in software markets suggest that there may be efficiencies associated with horizontal mergers. For example, efficiencies would result if development costs can be saved by having one rather than two teams develop new products. Whether efficiencies of this type would make up for any loss in variety and loss in direct competition requires a fact-specific inquiry in any given case.

Other efficiencies flowing from software mergers may be achievable without the necessity of a merger. For example, while the degree of compatibility between two programs can be increased through a merger, such changes may well be possible through cooperation on development efforts and the licensing of copyrights without the necessity of a full merger. Under the Guidelines, efficiencies of this sort, which are not "merger-specific," cannot be used to defend or justify an otherwise anticompetitive deal.[24]

4.2 Vertical and Complementary Software Mergers

We have thus far focused on horizontal software mergers. In fact, a number of the software deals that have been challenged or modified by the agencies have involved vertical or complementary mergers which involve products that work together rather than serve as substitutes for one another.

The primary concern in these cases has been based on the two-level entry theory. Under this theory, a complementary merger can make entry more difficult by requiring an entrant to develop products in two markets at once: two distinct types of software, or hardware and software. A variant of this theory involves the integrated firm's degrading the compatibility of products sold by rivals that compete with its own products in one of the markets. For the two-level entry theory to be applicable, market power and entry barriers must be significant at each of the two levels. The market power must be such that an entrant into a single level is significantly disadvantaged by not being able have its component work with the otherwise complementary component produced by the merging firms. These theories, and their application, are subtle.

[24] The DOJ and FTC revised the 1992 *Guidelines* in 1997 to articulate more fully how efficiencies would be handled in the merger review process. Although the stated intention of the agencies was to be more receptive to efficiency claims, it remains to be seen how this will work in practice, both generally and in the software industry.

Vertical mergers offer their own prospects for efficiencies, too. Some theories of vertical integration, for example, imply that there are efficiencies associated with integration across complementary products. Somewhat ironically — in theory, at least — the integration of a firm with a monopoly in one product into a complementary product market can make entry into the latter market more difficult. This effect arises because the multi-market producer internalizes the complementarities and is a more aggressive competitor as a consequence. The net effect in this scenario would be to deter entry and lead to monopoly in both markets.

4.3 Software Mergers Challenged or Modified

We turn now to see how these principles have played out in actual merger investigations conducted by the DOJ and the FTC. We focus here on software mergers that were either abandoned or modified in response to antitrust challenge.[25] We do this because there is far less public information concerning transactions that were cleared without modification. We note, however, that the largest software deal to date, the acquisition of Lotus by IBM, was not modified by the antitrust agencies. Furthermore, Microsoft has engaged in a series of acquisitions that have not been challenged, including its acquisition of Vermeer and its FrontPage product.

4.3.1 Borland and Ashton-Tate (1991)

In 1991 Borland International announced its intention to acquire Ashton-Tate. The key product overlap was in the area of database management programs. The two leading programs at the time were Ashton-Tate's dBase program and Borland's Paradox program.

This case was an important early test of how mergers in the personal computer software industry would be treated by the antitrust agencies. Would the deal be blocked as the merger of the two leading suppliers of personal computer based "relational database" programs, or would the deal be permitted in the light of the highly dynamic nature of software markets? In their presentations at the Justice Department, the merging parties asserted that entry barriers into database software were low, that dBase was in decline, that Paradox had gained market share rapidly and recently, and that

[25] One of us, Shapiro, has been involved in many of these merger reviews. In particular, Shapiro worked for the government in the Adobe/Aldus, Microsoft/Intuit, and Computer Associates/Legent deals, and for the merging parties in the Borland/Ashton-Tate, Silicon Graphics/Alias/Wavefront, and Cadence/CCT deals. The statements in this paper are not intended to represent the views of either the government agencies or the companies involved.

the merger would offer dBase users a "migration path" to new and better software. Going to the heart of the matter, Jim Rill, then Assistant Attorney General for Antitrust, asked point blank whether the merging parties were asserting that the Division should permit all software mergers to proceed based on these arguments.

To satisfy DOJ concerns, Borland agreed to issue FoxPro a license to the dBase code; FoxPro was a rival to dBase then in litigation with Ashton-Tate over infringement of dBase copyrights. The license was intended to insure that the installed base of dBase users had a viable alternative outside of Borland's control. And look what happened! Paradox is dead, dBase faded out, Microsoft purchased FoxPro to serve the high end of the market and promoted Access at the low end, and Microsoft now dominates the personal computer database market. Some would point to this case as evidence that software markets are so fluid that mergers are of little concern. We would draw a more limited conclusion: that licensing fixes to mergers can indeed enable new competitors.

4.3.2 Adobe and Aldus (1994)

In 1994 Adobe announced its intention to acquire Aldus. The two companies sold the leading brands of professional illustration software: Adobe Illustrator and Aldus Freehand. The parties argued with some effect that each was driven to upgrade its product in order to earn revenues from its own installed base. This was not the only dimension along which competition took place, however. There was evidence, for example, of substantial direct pricing competition between the two programs, both for new customers and for sales to their own installed bases. In the end, the FTC required the merged firm to divest the FreeHand professional illustration software owned by Aldus to a third firm, Altsys Corporation, which had originally developed the software. Again licensing was seen as a fix to a direct horizontal overlap in software products.

4.3.3 Silicon Graphics and Alias and Wavefront (1995)

In 1994 Silicon Graphics, Inc. (SGI), a maker of high-end graphics workstations, announced its plans to acquire two relatively small software houses specializing in "entertainment graphics software," Alias Research Inc. and Wavefront Technologies. This software is used in producing high-resolution two- and three-dimensional images, e.g., the dinosaurs in Jurassic Park and the characters in electronic games. SGI was responding in part to Microsoft's acquisition of the third leading firm in this segment, SoftImage, Inc.

This double deal had significant horizontal as well as vertical aspects. The parties argued in part that SGI had no incentive to raise the price of the software, since this would cut into the sales of the SGI hardware running that software. Ultimately, expressing more concern over the vertical aspects of the deal than its horizontal element, the FTC, in a 3-2 vote, required SGI to enter into a porting agreement with one of DEC, HP, IBM, Sun or another company as approved by the Commission, to make sure that Alias's software was available on these other platforms.

The FTC also required that SGI

> establish and maintain an open architecture, and publish the Application Program Interfaces (APIs), for [SGI's] computers and operating systems in such manner that software developers and producers may develop and sell Entertainment Software for use on [SGI's] computers in competition with Entertainment Software offered by [SGI].[26]

For those watching the Microsoft case, and for those contemplating mergers in the software or hardware industry, the SGI precedent of opening up APIs is worthy of note. Although the FTC action can be criticized on a number of grounds, including the fact that SGI's market position has deteriorated markedly over the past three years (calling into question whether they ever had any meaningful monopoly), it stands as an example of mandated "open interfaces." Although critics assert that such provisions are either burdensome or unenforceable, or both, we are not aware of any disputes that have arisen under this consent decree regarding the definition of "open."

4.3.4 Microsoft and Intuit (1995)

In 1994, Microsoft proposed a $2 billion acquisition of Intuit, Inc. Intuit was the owner of Quicken, the leading personal financial software package. Microsoft's Money product performed many of the same functions. The government viewed Quicken and Money as competing in a market for "Personal Finance/Checkbook" software. In that market, Quicken was the leading product, with a 69 percent unit share, followed by Microsoft's Money with a 22 percent unit share.

[26] Decision and Order in the Matter of Silicon Graphics, Inc., Docket No. C-3626, November 1995. The FTC also required that SGI offer independent entertainment graphics software companies participation in its software development programs on terms no less favorable than those offered to other types of software companies.

The Antitrust Division took note of certain comments made by Intuit's Chairman in a September 1994 memorandum to his board about the proposed acquisition of Intuit by "Godzilla" (Intuit's code name for Microsoft): "Our combination gives FIs [Financial Institutions] one clear option, eliminating a bloody share war and speeding adoption. That, in turn enriches the terms of trade we can negotiate with FIs." Based on this and other evidence, the DOJ described Microsoft as Intuit's most significant competitor, and stated that the proposed acquisition would eliminate competition between Microsoft and Intuit, which had benefited consumers by leading to high quality, innovative products at low prices

The Antitrust Division rejected Microsoft's proposed "fix" in which some of its Money assets would have been transferred to Novell Inc. The Division believed that Novell would not be as effective a competitor with Money as was Microsoft. The Division also did not accept Microsoft's arguments that entry was easy,[27] and that competition from banks (e.g., on-line banking) would discipline the pricing of Quicken. Moreover, in this situation a licensing fix was regarded as inadequate. In response to DOJ's challenge, the parties abandoned the transaction in July 1995.

4.3.5 Computer Associates and Legent (1996)

In 1996 Computer Associates proposed to acquire Legent for $1.7 billion. The focus of the antitrust inquiry was on certain mainframe computer software markets. In particular, Computer Associates and Legent were the largest and second-largest vendors of systems management software products for IBM mainframe computers.

Mainframe software markets are different from personal computer software markets in a number of respects: mainframe software is a much more stable market, which is experiencing little if any growth; technological change is not so rapid; there is very substantial lock-in by individual customers, although network effects are less pronounced; the software itself is extremely sophisticated; and vendor reputation is critical, due to the "mission critical" nature of much of this software.

Computer Associates agreed to grant licenses for Legent's products in each of five software markets of concern to the Antitrust Division. The five areas all involved computer systems management software products used with mainframe computers running the VSE operating system: security

[27] The experience of Computer Associate's "Simply Money" program in this market is instructive regarding entry barriers in software. Even though Computer Associates virtually gave its program away, and received some favorable reviews, it still could not gain wide acceptance.

software; tape and disk management software; job scheduling software; and automated operations software. The goal of the settlement was to establish a new viable competitor in each of these areas. Two aspects of this case are noteworthy. First, notice that the relevant product markets are quite "narrow," reflecting the fact that users need solutions in each of these categories, and the specialized nature of the software that meets these needs. Second, the government found that entry was quite difficult, a reminder that ease-of-entry is not a silver bullet for merging software companies.

4.3.6 Autodesk and Softdesk (1997)

Autodesk, Inc. negotiated a consent decree in 1997 with the FTC to settle Commission concerns about its proposed $90 million acquisition of Softdesk, Inc. Autodesk develops and markets computer-aided design (CAD) software for use in the architecture, engineering, and construction industries, including "AutoCAD," a design engine for use on Windows-based personal computers. Autodesk products account for some 70 percent of the installed base of Windows-based CAD engines, with approximately 1.4 million users. Softdesk, which primarily sells CAD application software, was developing and testing its own CAD engine, IntelliCADD, and was within months of introducing IntelliCADD into the market, when the Autodesk acquisition of Softdesk was announced.

Compatibility issues were central in this enforcement action. The FTC asserted that "IntelliCADD, if brought to market, would have provided substantial direct competition to AutoCAD because it offered compatibility and transferability with AutoCAD generated files and application software — features other CAD engines do not offer." The FTC further alleged that "the large installed base of AutoCAD users necessitates that any new CAD engine developed and offered in the market offer file compatibility and transferability with AutoCAD in order to be an effective competitor."[28]

The FTC asserted that Autodesk's acquisition of Softdesk, as originally proposed, would have substantially lessened competition in the development and sale of CAD software engines. Under the terms of the settlement, IntelliCADD was divested to Boomerang Technology, Inc., which in turn assigned and sold its rights and title to IntelliCADD to Visio Corporation. The settlement did not include the IntelliCADD development team, although it did prohibit Autodesk and Softdesk from interfering with the ability of Boomerang to recruit or hire employees of Softdesk who worked on development of IntelliCADD.

[28] FTC Press Release, March 31, 1997, at http://www.ftc.gov/opa/1997/9703/autodesk.htm.

4.3.7 Cadence Design Systems and Cooper & Chyan Technology (1997)

Cadence Design Systems, Inc., of San Jose, California, agreed in 1997 to settle FTC charges that its $400 million acquisition of Cooper & Chyan Technology, Inc. (CCT) would substantially reduce competition for key software used to automate the design of integrated circuits, or "microchips."

The FTC's was primarily concerned with the vertical aspects of this transaction. In particular, Cadence's "Virtuoso" layout environment was seen as a platform on which a variety of software could run, and the FTC acted to ensure that other brands of software — competitive with that offered by CCT — would not be blocked from running on the Cadence platform. This case illustrates that many software companies, not just Microsoft, can be characterized as controlling a key "platform" with which other programs must work.

4.4 Are Software Mergers Different?

These cases demonstrate convincingly that software companies are not immune from the antitrust laws. Furthermore, a close look at these cases and how they were analyzed reveals that the computer software industry, while surely distinct from other industries, does not require its own unique merger policy. To the contrary, the 1992 Horizontal Merger *Guidelines* work just fine in the software industry. The specific application of the *Guidelines*, and Section 7 of the Clayton Act for that matter, to software is no doubt different from other industries, but the same could be said of a great many industries, from defense to telephones to pharmaceutical drugs to airlines to supermarkets to banking.

The primary arguments put forward by software companies to convince the FTC, the DOJ, and ultimately the courts not to block their proposed transactions are familiar: current market shares overstate future competitive significance; entry is easy because of rapid technological change; collusion is difficult because products are differentiated and prices for large buyers are negotiated in secret; the merging products are not "close" in the attributes they offer to consumers; competition from the installed base limits the market power associated with new sales; and dramatic synergies associated with having a single company offer a full product line counteract any loss of direct competition. These arguments may or may not carry the day. What is needed is factual inquiry on a case-by-case basis, not a new framework for merger analysis.

In summary, merger policy in the software industry appears to be on a sound footing, applying the general principles laid out in the 1992

Horizontal Merger *Guidelines* to the specific fact patterns found in software. Clearly, the enforcement agencies consider monopoly power in software to be a genuine concern, and they are prepared to prevent consolidations that threaten competition. This policy is all the more sound given the limited ability of antitrust law to control unilateral conduct by dominant software suppliers, as we discuss below.

5. COOPERATIVE STANDARD SETTING

There are many forms of inter-firm cooperation that fall short of merger. In systems markets generally, and software markets specifically, one crucial area of cooperation is in the setting of compatibility standards. These standards allow the various components of a system to work together. When firms write software adhering to the same standards, their programs can either communicate directly with one another, or share complementary components within the overall system. In networks markets, the extent to which various products are compatible with one another is one of the most important dimensions of market structure, conduct, and performance.

Collective standard setting is common in high-tech industries, including software. Even the fiercest enemies often team up in the software industry to promote new standards. Last year, Microsoft and Netscape, two companies hardly known as cozy partners, agreed to include compatible versions of Virtual Reality Modeling Language (developed by Silicon Graphics) in their browsers. This agreement is expected to make it far easier for consumers to view 3-D images on the Web. Earlier, Microsoft agreed to support the Open Profiling Standard, which permits users of personal computers to control what personal information is disclosed to a particular Web site, and which had previously been advanced by Netscape, along with Firefly Network, Inc. and Verisign Inc.

Clearly, antitrust concerns have not prevented many cooperative standard-setting efforts from proceeding. But neither is such activity immune from antitrust scrutiny. In the consumer electronics area, for example, the Justice Department investigated Sony, Philips, and others regarding the establishment of the CD standard in the 1980s. Cooperative efforts to set optical disc standards have also been challenged in private antitrust cases, on the theory that agreements to adhere to a standard are an unreasonable restraint of trade:

> [d]efendants have agreed, combined, and conspired to eliminate competition... by agreeing not to compete in the design of formats for compact discs and compact disc players, and by instead agreeing to establish, and establishing, a common format and design...[29]

Does cooperation lead to efficient standardization, increased competition, and additional consumer benefits? Or is cooperative standard setting a means for firms collectively to stifle competition, to the detriment of consumers and firms not included in the standard-setting group? Answering these questions and evaluating the limits that should be placed on cooperative standard-setting efforts require an analysis of the competitive effects of such cooperation in comparison with some reasonable but-for world. Inevitably, an antitrust analysis of cooperative standard-setting involves an assessment of how the market would likely evolve *without* the cooperation. One possibility is that multiple, incompatible products would prevail in the market, if not for the cooperation. Another possibility is that the market would eventually tip to a single product, even without cooperation. Even in this latter case, an initial industrywide standard can have significant efficiency and welfare consequences, for three reasons: (1) cooperation may lock in a different product design than would emerge from competition; (2) cooperation may eliminate a standards war waged prior to tipping; and (3) cooperation is likely to enable multiple firms to supply the industry-standard software, whereas a standards war may lead to a single, proprietary software product.

5.1 The Costs and Benefits of Compatibility and Standards

We begin by laying out the costs and benefits of achieving compatibility. We then turn to the legal treatment of cooperation to set compatibility standards.

5.1.1 Greater Realization of Network Effects

When all users are on a single network, the size of the network is maximized and so is the realization of network benefits. For communication networks, users benefit from the fact that any given user can communicate with any other. For hardware-software networks, users benefit from the fact that firms supplying components have access to a large market for their software. This is likely to lead to increased entry and variety, and greater price and innovation competition in the supply of software components.

[29] Second Amended Complaint, Disctronics Texas, Inc., et al. v. Pioneer Electronic Corp. et al. Eastern District of Texas, Case No. 4:95 CV 229, filed August 2, 1996 at 12.

5.1.2 Buyers are Protected from Stranding

When products are compatible, a consumer does not fear being stranded when he or she chooses to make a purchase from a particular supplier. When a consumer buys a television set in the United States, for example, he or she knows that it is compatible with the signals sent out by local broadcasters — the Federal Communications Commission (FCC) sets standards that all television receivers must meet. In contrast, neither the FCC nor anyone else set AM stereo standards for years. The result was consumer confusion and a reluctance to buy.

5.1.3 Constraints on Variety and Innovation

While the first two effects are benefits of standardization, the third effect is a cost. The need to adhere to a standard imposes limits on firms' product design choices. These limits can lead to static losses from the reduction in variety. And they can lead to dynamic losses as firms are foreclosed from certain paths of R&D that could result in innovative new products that could not comply with the standards. Note that these limits impose costs both at the time a new product is created, and later when it is possible to introduce a new generation offering greatly enhanced performance. In the latter case, firms must confront the issue of whether to preserve intergenerational compatibility.

5.1.4 Impact on Competition

In the presence of network effects, compatibility can fundamentally affect the nature of competition. The importance of compatibility stems from the fact that compatible programs constitute a single network. Increased adoption of one vendor's program does not create a competitive advantage for that vendor relative to its rivals' because the rivals' programs also benefit from the larger network size. In contrast, when programs are incompatible, different programs constitute different networks. Consequently, the increased adoption of a program creates a larger network for that program but not for competing programs. Thus, increased adoption of a particular program creates a competitive advantage for that program by raising the value of that software relative to programs that are not part of that network.

This fact has several consequences. To illustrate, suppose that everyone expects the market to tip eventually. If these expectations are correct, then eventually there will be a single network, whether or not firms agree to a common standard. In this setting, there are two ways to achieve industrywide compatibility. One is for firms to agree up front to a common

standard. The other is for firms to battle for dominance. Under incompatibility, firms will *compete for the market*. Firms may make big investments and incur initial losses as they attempt to become the dominant network. In contrast, under compatibility, firms will *compete within the market*. Network effects do not provide a means for a firm to pull ahead of its rivals and perhaps even become a monopolist. Instead, firms will compete along other dimensions, such as price, product features, and post-sales service.

This suggests an overall pattern. Cooperative standard setting mutes the intense front-end competition characteristic of a standards war, while permitting greater competition later in the life of a product, since multiple firms can provide products that comply with the standard. In other words, cooperative standard setting tends to decrease competition along some dimensions and in the near term, while increasing competition on other dimensions and in the future. On net, compatibility can either increase or decrease competition, depending on market conditions. To see how standardization affects competition, we must compare the evolution of a market with and without the compatibility of competing programs.

One must be careful in applying this analysis of competitive effects. Generally, it does not give a clear answer, but rather suggests a trade off: *ex ante* vs. *ex post* competition, you can have one but not both. There is, however, an important set of situations in which compatibility gives rises to increased competition at all points in time. These situations arise when the entire product category would fail to take off in the absence of standardization. This can happen if consumers withhold making initial purchases (or if producers of complementary components refrain from making investments) because they are too worried about being locked in to the wrong choice.

5.1.5 Weighing the Benefits and Costs

This discussion should make it clear that there are no easy or general answers regarding the impact of cooperative standard setting on competition, efficiency, and consumer welfare. Still, we believe that our economic framework helps frame the key questions and gives insight into the proper scope for collective standard setting. As a general matter, antitrust analysis of inter-firm cooperation should assess the harm to third parties who are not part of the agreement. The leading candidates are consumers and those suppliers who do not control and/or participate in the standard-setting process.

The clearest case favoring standard setting arises when collective action is essential to get the bandwagon moving at all. This could happen if two or

more firms have crucial intellectual property that must be contributed to develop a successful product. This also could happen if consumers simply would not adopt any product without the unified support of a number of software suppliers. In these situations, collective standard setting benefits consumers as well as the software vendors.

Collective standard setting also is likely to be desirable, even if multiple suppliers could offer competing programs, so long as network effects are strong and the standard does not unnecessarily restrict product variety. Because of the network effects, total efficiency is greatest when there is a single network; the best one can hope for is to achieve this result while enabling several firms to offer compatible programs. If variety can still flourish within the standard, the outcome can be very efficient and preserve considerable competition even while exploiting network effects.

Cooperation becomes more problematic if the participants agree to standards that compel each to pay royalties to the others. This may simply be a form of induced collusion. One sign of this may be agreements where one piece of intellectual property from each member of the coalition is included in the standard. Of course, this pattern may also reflect the fact that the parties are getting together to resolve blocking IPRs, in which case cooperation is necessary to move forward at all. To distinguish the cartel situation from the patent unblocking situation, the key question is whether a successful product could be launched by one or a subset of the parties without infringing the IPRs of the others.

Another pattern worthy of antitrust attention arises when a subset of firms in an industry adopt a standard that encompasses their IPRs and makes it necessary for anyone producing to that standard to make payments to those firms. This can be a means for that set of firms to jointly monopolize the market. Such concerns can be alleviated if the firms agree to license their intellectual property openly on fair and reasonable terms, as required by numerous standard-setting organizations including the American National Standards Institute and the International Standards Organization.

5.2 Legal Treatment of Cooperative Standard Setting

The question of whether firms should be allowed, or even encouraged, to set standards cooperatively is part of the broader issue of collaboration among competitors, a storied area within antitrust law. The limits imposed by public policy in the area of compatibility standards remain unclear. The most specific statement by the antitrust enforcement agencies can be found

in a recent FTC Staff Report. [30] The Staff Report recognized a need for clarification in this area:

> the time has come for a significant effort to rationalize, simplify, and articulate in one document the antitrust standards that federal enforcers will apply in assessing collaborations among competitors. This effort should be directed at drafting and promulgating 'competitor collaboration guidelines' that would be applicable to a wide variety of industry settings and flexible enough to apply sensibly as industries continue rapidly to innovate and evolve.[31]

Since that call for action, the FTC has conducted Joint Venture Hearings, and Chairman Pitofsky has stated that the Commission and the Antitrust Division are exploring the possibility of issuing guidelines for joint ventures, including standard-setting activities.

Antitrust policy can focus on the outcomes of cooperative standard setting, or it can focus on the process itself. Antitrust liability has been found for participants in a standard-setting process who abuse that process to exclude competitors from the market. This does not appear to be a problem for an "open" standard, but could arise if the companies promoting the standard block others from adhering to the standard or seek royalties from outsiders.

Legal cases have tended to look at quality and performance standards rather than compatibility standards. As the Supreme Court has noted, "Agreement on a product standard is, after all, implicitly an agreement not to manufacture, distribute, or purchase certain types of products."[32] To date, this type of reasoning has not been used to impose *per se* liability on software standard-setting activities. Indeed, we know of no successful antitrust challenges to cooperation to set software standards. We believe that the antitrust risks faced by companies who are trying to set software compatibility standards are minor as long as the scope of the agreement is limited to standard setting. While the law has typically looked for integration and risk-sharing among collaborators in order to classify cooperation as a joint venture and escape *per se* condemnation, these are not very helpful

[30] Federal Trade Commission, "Anticipating the 21st Century: Competition Policy in the New High-Tech Global Marketplace," Chapter 9, "Networks and Standards," (June 1996).

[31] *Ibid,* Chapter 10, "Joint Ventures," (June 1996) at 17.

[32] Allied Tube & Conduct Corp. v. Indian Head, Inc., 486 U.S. 492, 500 (1988). See Anton Yao (1995) for a more complete discussion of the legal treatment of performance standards.

screens for standard-setting activities. The essence of cooperative standard setting is not the sharing of risks associated with specific investments, or the integration of operations, but rather the contribution of complementary IPRs and the expression of unified support to ignite positive feedback for a new technology.

An excellent illustration of how the enforcement agencies can successfully handle standard-setting activities comes from the Justice Department's June 1997 approval of the proposal by Columbia University and nine companies to create a clearinghouse to offer a package license of patents needed to meet the MPEG-2 video compression standard developed by the Motion Picture Expert Group. The portfolio will only contain patents found to be truly essential to the MPEG-2 standard. The MPEG-2 standard is used in many forms of digital transmissions, including digital television, direct broadcast satellite, digital cable systems, personal computer video, DVD, and interactive media. It was important to the Justice Department that the pool was restricted to blocking patents, which are complements, not substitutes, as determined by an independent expert. The scope of the cooperation endorsed by the Justice Department was to unblock patent positions, and to reduce transaction costs through the use of a clearinghouse.

5.3 Standards and Hidden Intellectual Property Rights

Firms are sometimes accused of hiding intellectual property rights until after the proprietary technology has been embedded in a formal standard. We view this issue primarily as one of contract law. Standard setting groups should — and often do — have provisions in their charters compelling members either to reveal all relevant IPRs or to commit to licensing any IPRs embedded in the standard on "reasonable" terms.

In some cases, however, the precise requirements imposed by a standard-setting group may be unclear. In these circumstances, if the standard affects non-participants, including consumers, there is a public interest in clarifying the duties imposed on participants in a fashion that promotes rather than stifles competition.

5.3.1 Dell Computer and the VESA VL-bus Standard

The leading example of this type of antitrust action is FTC's consent agreement with Dell Computer Corporation, announced in November 1995. Although the case involved computer hardware, it is important for the software community as well. The assertion was that Dell threatened to exercise undisclosed patent rights against computer companies adopting the VL-bus standard, a mechanism to transfer data instructions between the

computer's CPU and its peripherals such as the hard disk drive or the display screen. The VL-bus was used in 486 chips, but the PCI bus has now supplanted it. According to the FTC,

> During the standard-setting process, VESA [Video Electronics Standard Association] asked its members to certify whether they had any patents, trademarks, or copyrights that conflicted with the proposed VL-bus standard; Dell certified that it had no such intellectual property rights. After VESA adopted the standard — based in part, on Dell's certification — Dell sought to enforce its patent against firms planning to follow the standard.[33]

There were two controversial issues surrounding this consent decree: (a) the FTC did not assert that Dell acquired market power, and indeed the VL-bus never was successful; and (b) the FTC did not assert that Dell *intentionally* misled VESA. Our analysis suggests that anticompetitive harm is unlikely to arise in the absence of significant market power and that the competitive effects are not dependent on Dell's intentions.

5.3.2 Motorola and the ITU V.34 Modem Standard

Another good example of how competition can be affected when standard-setting organizations impose ambiguous duties on participants is the case of Motorola and the V.34 modem standard adopted by the International Telecommunications Union. Motorola agreed to license its patents essential to the standard case to all comers on "fair, reasonable, and non-discriminatory terms."[34] Once the standard was in place, Motorola then made offers that some industry participants did not regard as meeting this obligation. Litigation ensued between Rockwell and Motorola, in part over the question of whether "reasonable" terms should mean: (a) the terms that Motorola could have obtained *ex ante*, in competition with other technology that could have been placed in the standard; or (b) the terms that Motorola could extract *ex post*, given that the standard is set and Motorola's patents are essential to that standard.

We think these issues are best dealt with by the standard-setting bodies, or standard-setting participants, either by making more explicit the duties imposed on participants, or by encouraging *ex ante* competition among different holders of IPRs to get their property into the standard.

[33] *See* http://www.ftc.gov/opa/1996/9606/dell2.htm.

[34] While one the authors, Shapiro, served as an expert in this matter retained by Rockwell, the views stated here do not reflect those of any party to the case.

Unfortunately, antitrust concerns have led at least some of these bodies to steer clear of such *ex ante* competition, on the grounds that their job is merely to set technical standards, not to get involved in "prices," including the terms on which intellectual property will be made available to other participants. The ironic result has been to embolden some companies to seek substantial royalties after participating in formal standard setting activities.

5.4 Policy Implications

What does this analysis tell us antitrust enforcers should look for when deciding whether to allow cooperative standard setting?

Do the firms in the proposed standards coalition have market power? Answering this question is made difficult by the fact that the product may not yet have been brought to market. The analysis must thus focus on capabilities. In this sense, the inquiry is akin to conducting a market power analysis for a merger case based on potential entry effects. If the firms lack market power and there are firms that jointly or individually could put forth competing standards, then the cooperation is unlikely to harm competition.

Does the coalition have open or closed membership? Open membership defuses the danger that the firms involved will exclude others from the market, but increases the likelihood that the members do or will possess market power. "Small" open groups thus are the least worrisome.

Do members of the coalition possess blocking patents or other IPRs? If two or more companies each have patents that are essential to production of the good, then some form of cooperation is far more likely to be desirable. Cooperation is not essential; the firms could license each other and third parties separately. However, separate licensing is prone to higher royalty rates than collective licensing because an IPR owner acting individually fails to take into account the harm it does to holders of complementary IPRs when it raises its license fees.

Are royalties required to adhere to the standard? Such royalties will tend to raise the price of any software complying with the standard. Royalties that reward owners of blocking patents or copyrights are easily defensible, but royalties can have cartel-like effects.

Is coordination critical to launch of the product? Cooperation is desirable in those situations where the product would fail to take off in the absence of standardization. Of course, the difficulty in applying this test is to determine whether standardization really is needed. Indications that either buyers or the suppliers of complementary components strongly favored standards can be useful evidence.

What ancillary restraints are placed on members of the standards coalition? Is a member firm allowed to produce software that does not adhere to the standard? If there are no limitations, then cooperation is less likely to harm competition. It is important to recognize, however, that there may be good reasons to limit members' ability to produce non-standard products. Indeed, as we will discuss in the next section, firms may produce non-standard products to sabotage the competitive efforts of rivals.

6. UNILATERAL COMPATIBILITY DECISIONS

We now examine the issue of whether dominant firms should be forced to make their products compatible with those of other suppliers. We begin by noting that there is something of a paradox in antitrust policy toward standard setting. While firms may be subject to legal attack for collectively setting standards that facilitate compatibility, a firm may also be subject to legal attack for *refusing* to participate in an industry standard. We believe that this paradox can be resolved by distinguishing between two generic settings. In cases where firms are *symmetrically* situated and have roughly equal actual and anticipated market positions, firms may cooperate to avoid intense competition. In *asymmetric* cases, where one firm is dominant or is expected to be so in the absence of compatibility, that firm may refuse to cooperate as a means of maintaining or achieving dominance.

6.1 The Effects of Compatibility on Price Competition

In our discussion of cooperative standard setting, we focused on settings that were symmetric in the sense that no one firm would be favored if there were competing networks. In those markets, incompatibility can create a winner-take-all situation in which competition to be the winner is intense early on in the product's life and competition is later diminished once a dominant firm emerges.

Incompatibility can have very different effects in situations where one firm is expected to be dominant at the time that compatibility decisions are being made. With incompatibility, reputation, existing installed bases, and expectations are key sources of competitive advantage. Thus, a firm that enjoys a superior reputation may favor incompatibility, which can lead to tipping and monopolization. For this reason, firms with reputation or installed based advantages often oppose compatibility and the development of industry standards, preferring to establish *de facto* standards over which they can exert greater control. The lesson: a dominant firm will often spurn

collective standard setting, betting that it can establish its own proprietary standard.

6.2 The Effects of Compatibility on Innovation

Compatibility and innovation interact in a complex manner.[35] In sorting out these effects, it is important to distinguish three types of compatibility:

1. compatibility between two different generations of a software product;
2. compatibility between rival offerings of the same software product;and
3. compatibility between two distinct software products, *e.g.,* the operating system and a spreadsheet program.

We address these in turn.

The need to maintain intergenerational compatibility limits design freedom and can slow innovation. This is true whether one firm controls the older generation of technology or it is open, and reflects a fundamental tradeoff between switching costs and innovation. In the face of a proprietary standard, incompatibility is likely to make it impossible for an entrant to come in with an incrementally better product because it would have to overcome both consumer switching costs and the lack of network benefits.

Likewise, the spur to innovate can be greatest when firms are engaged in a battle over incompatible software products. As the firms compete to establish new products, consumers may use product quality as a basis for forming expectations about future network size. In this setting, incompatibility can strengthen innovation incentives — a firm gets an extra kick in terms of consumer expectations if it successfully innovates.

These two arguments suggest that innovation will be greatest when software vendors are free, or even forced, to offer incompatible products. This is not the end of the story, however. We already noted above that the lack of a standard, or blocking IPRs, may cause the entire product category to flop. In addition, there is a very real danger that a single firm, controlling important standards and interfaces, will stifle innovation.

This situation, in which a single firm controls a key interface, to which others need access in order to innovate, is central to the debate over many of Microsoft's actions. Both sides in this debate have overplayed their hands. Microsoft-bashers, who assert that Microsoft can and will crush upstarts in any software category that is attractive to Microsoft, are clearly wrong. Microsoft can benefit from such innovations, either by acquiring the innovating firm or by encouraging complementary improvements that raise the value of Microsoft's own products. Yet defenders of Microsoft, who

[35] For a more in-depth treatment of the link between compatibility and innovation, see Farrell and Katz (1998).

assert that Microsoft seeks merely to offer better software to consumers, from any source, also err. They ignore Microsoft's incentives to control or stifle innovations that threaten its dominant position in the provision of operating systems.[36] If these innovations must work with Windows in order initially to gain acceptance, Microsoft can use its control over interfaces strategically to raise entry barriers, both for operating systems and for applications software.

6.3 Interfaces as Essential Facilities?

The prospect that a single firm, controlling a key input (interface), can protect a dominant position, or extend its dominance into new areas, raises a number of classic antitrust questions. These issues have frequently been explored under the rubric of the "essential facilities" doctrine. When, if ever, should the Courts step in and mandate that the dominant firm open its interface (the "facility") to enable additional competition?

The basic tradeoff is fairly clear. A software manufacturer may expend considerable resources to build a network (*e.g.,* Windows, the applications programming, and the training of users). If the manufacturer is forced to open its network to others, then investment incentives may be diminished. On the other hand, the incentives of potential entrants will be increased (or at least their incentives for incremental innovations will be).

Having said this, we are wary of imposing a duty to deal on owners of intellectual property, including Microsoft. Such a duty is fundamentally at odds with the granting of the intellectual property rights themselves, which explicitly involve the power to exclude others from infringing on those rights. Furthermore, invoking the essential facilities doctrine raises a host of practical problems regarding the terms and conditions on which the dominant firm will be forced to deal.

A less drastic remedy is to put limits on the ability of dominant firms to *change* their policies by shutting down interfaces that had been open. We recognize that rules of this type do not always protect consumers in network markets from lock-in, because uncoordinated consumers collectively may pick a less favorable product that is proprietary from the outset. We also recognize that determining what constitutes a "change in policy" can be very difficult in a dynamic environment. Still, rules against *installed-base opportunism* would seem far less dangerous than broader duties in terms of stifling innovation by leading firms and undermining intellectual property rights.

[36] See our discussion of tying and foreclosure below.

6.4 Policy Implications

There are at least three broad questions to answer in assessing whether a firm should be forced to open up its interfaces:

Does the firm have monopoly power? If not, any argument for mandatory open interfaces is weak at best. As with collaborative standard setting, this assessment must be forward looking.

How does the firm maintain incompatibility? If the firm has consistently enforced its intellectual property rights and prevented others from copying its interfaces (either program-to-program, or user-to-program as in the case of graphical user interfaces), the case for mandating open interfaces is weak. A stronger case can be made for intervention if the firm tries to close down interfaces that had previously been made open. Even if the interfaces lack intellectual property protection, and the firm keeps them secret and/or constantly changes them without providing specifications to outsiders, forcing the firm to freeze its interfaces can be dangerous. If duties are to be imposed in this situation, a better way to open the interfaces would be to allow changes but to require the firm to publicly announce them in a timely fashion, such as the European Union did in its undertaking with IBM.

Are open interfaces a remedy for other antitrust violations? We are much more receptive to mandatory open interfaces as a remedy for other antitrust violations. For example, as discussed above, companies may agree to license certain intellectual property as a remedy in a merger case.

7. TYING AND BUNDLING

Many people claim that firms can and will harm competition by forcing consumers to take the firm's products as part of a package deal. Economists have tended to dismiss tying claims, while the courts have made a muddle of them. With the battle between Microsoft and the U.S. Department of Justice over the bundling of Explorer with Windows95, the issue has gained a new level of publicity, if not respectability.

Because of the imprecision with which many people use the term *tying* and the related term of *bundling*, we start by defining them carefully.

Tying: *Software program B is tied to program A if firm M refuses to sell program A (the "tying" good) unless the customer also purchases program B (the "tied" good) from firm M.* It is important to note that this definition is by itself incomplete. One must also specify whether there is a requirement to purchase *all* of good B from firm M in order to be able to buy any of good A. It is also worth noting that the issue of compatibility is related to tying. If a single firm produces components *A* and *B*, and it ensures that they are

incompatible with competing components produced by any other suppliers, then the firm has effectively tied its two components.

Bundling: *Programs A and B are bundled if the price of the two programs sold together as a package is less than the sum of their individual-purchase prices.* It is also useful to distinguish between *pure bundling*, where the software is offered *only* as a package, and *mixed bundling*, where the individual components are offered for sale separately as well as in a package.

A special case of bundling is of considerable interest in the software industry, and more generally in markets for the licensing of intellectual property: when the price of programs *A* and *B* together is the *same* as the price of program *A* alone. This is the situation in which program *B* is given away for free to customers buying *A*. This practice is not uncommon in the licensing of patents, where collections of patents are often offered as a package. Such *package licenses* are most likely to arise in circumstances where the pieces of the package have extremely low, or zero, marginal cost to the supplier. The licensing of copyrighted software meets this test, as does the licensing of patents.

The legal doctrine on tying and bundling is confused in general, and software pushes that doctrine to its limits. Because marginal costs are so low, and because software components are complementary, it may be efficient to bundle software modules. In some circumstances, however, such bundling can raise concerns about the ability of other firms to compete against the firm engaging in bundling. The current fight between Microsoft and the U.S. Department of Justice is destined to become a classic example.

7.1 Economic Rationales for Tying and Bundling

There are a number of different rationales or motivations for tying and bundling. Because they can have very different effects on competition and consumer welfare, it is important to distinguish among them. For each rationale in turn, we examine the economic logic and identify the welfare effects. Having done that, we will then discuss how one might determine which rationale is relevant in any given case.

7.1.1 Transactions Costs Savings

One possible motivation for bundling two distinct pieces of software is that doing so may lower costs. That is, there may be economies of scale and/or scope in production, distribution, marketing, and licensing. In thinking about this motivation for bundling, it is important to distinguish bundling solely in terms of how the software is offered for sale versus actual

integration of the code for two programs. While writing integrated code may provide increased functionality, it is hard to see how simply bundling two programs on the same disc at a single price would. The latter may, however, economize on licensing and distribution costs.

When the software is commercially — but not technologically — bundled to achieve transaction costs savings, the software publisher does not need to *require* customers to buy two distinct products. Instead, a lower package price reflecting the cost savings should be sufficient to induce customers to take the bundle.

To the extent that bundling is driven by transactions costs savings, it will improve economic efficiency and generally can be expected to benefit consumers. It may harm competitors, however, and thus one might well see competitors raising legal challenges to the practice with allegations of competitive harms.

7.1.2 Quality Assurance

It is sometimes argued that two components must be tied together because consumers can observe only the components' joint performance or quality level. The logic of the argument is as follows. Suppose that the two components are used together in a system but are sold separately by two different firms. And suppose that one of the firms decides to cut its costs by lowering the quality of its component. Consumers would observe that the system performed less well, but they would not know which component was too blame. As a result, both firms would suffer and the firm cutting its quality might not bear the full costs of its actions.

For this rationale to hold, one needs more than the fact that two programs are components of a common system used to produce some service. It must also be the case that the seller of one component suffers a loss when the system produces poor output as a result of the other component. Such losses will arise when either: (a) it is costly for buyers to determine which component is responsible for poor system quality — even with the seller's assistance — and the seller thus suffers a loss of reputation; or (b) the seller offers post-sales service and support and cannot prove when the performance problems are due to the other component.[37] Moreover, it must be expensive or difficult for the seller of the tying good to certify suppliers or set product specifications for the tied good.

[37] A related issue came up with DR-DOS. In that instance, there was no bundling, but there was a warning message to consumers who attempted to use a non-Microsoft operating system in conjunction with Windows 3.1. Microsoft was accused of using scare tactics.

Tying for the purpose of quality assurance improves economic efficiency and generally can be expected to benefit consumers. It may harm competitors who supply low-quality variants of the tied good.

7.1.3 Metering and Price Discrimination

When various consumers buy different goods in various proportions to one another, it may be possible for a firm to increase its profits by tying one good to another, because the varying purchase quantities can serve as a basis for sorting consumers or for extracting rents from them more effectively. For this strategy to be profitable, the seller must have market power with respect to at least some customer class in the tying market. Moreover, the ability to set an additional price must gain the seller something.

In the case of tying, the standard logic is relatively straightforward. Suppose that the firm has market power with respect to component A, and that each consumer purchases either one unit of A or none. Component B is competitively supplied, and different consumers buy varying numbers of units. To be concrete, think of a copier and paper. By requiring customers to purchase paper at an inflated price, the copier manufacturer is able to charge higher total prices to those consumers who make more intensive use of the copiers (*i.e.*, the manufacturer engages in metering).[38]

What is less obvious is how pure bundling can help a manufacturer extract greater surplus from consumers. The manufacturer compares the profits of selling components A and B at separate prices or in a package. To the extent that a consumer's willingness to pay for component A is negatively correlated with her willingness to pay for component B, there will be less variability in consumer's willingness to pay for the package than for the individual components and thus the firm will better be able to charge a price for the package that extracts revenues from consumers.[39] Mixed bundling can help the seller by creating a finer pricing structure with which to divide consumers and extract revenues from them.

Economic analysis demonstrates that consumers may on balance gain or lose from metering. It is difficult to see how metering can be said to harm competition.[40]

[38] Surprisingly, there are conditions under which it makes economic sense to tie two *completely unrelated* goods for reasons related to metering. We are not aware of any legal cases in which this issue has come up.

[39] While a negative correlation gives rise to this effect, it is not a necessary condition.

[40] For a survey of price discrimination and its effects, see Varian (1989).

7.1.4 Foreclosure of Competition in the Tied Market

Roughly speaking, the foreclosure theory of bundling examines whether a firm can sell programs *A* and *B* together in a bundle in a way that profitably harms competition in the market for program *B*. The idea is that potential rivals are foreclosed from competing to sell the tied program, *B*, which allows the firm to earn supranormal profits from sales of program *B*. Although many economists dismissed it — often alleging that there is only one monopoly rent to appropriate — there is a logically coherent argument for anticompetitive foreclosure. However, a carefully circumscribed set of conditions must be satisfied for this argument to apply.[41]

For a seller to be able to engage in profitable foreclosure in the tied product market, the seller first must have market power with respect to at least some customer class in the tying market. Second, there must be actual or potential profits in the tied market that are worth going after. Third, there must not be other ways to extract the profits from the sale of the tied good, a condition through which many economists have questioned the importance of this motivation. Finally, in many circumstances, the firm must make a *commitment* to tying for foreclosure to be effective.

In terms of welfare, consumers may be harmed by this loss of competition. Clearly potential suppliers who are foreclosed from competing are harmed. There may be relatively little effect on consumers if they are merely being switched from one high-margin supplier to another. If the effect is to reduce competition in the tied good significantly, however, then consumers may be harmed by this loss of competition. Clearly potential suppliers who are foreclosed from competing are harmed.

7.1.5 Foreclosure of Competition in the Tied Good Market: Blocking Two-Stage Entry

The idea behind this motivation is that in the presence of a tie, a new firm cannot enter in the production of just one product; it is forced by tying to either produce both components or neither. This is the same "two-level entry" theory that we addressed above in our discussion of vertical software mergers.

For the two-level entry theory to apply to tying and bundling practices, the seller must have market power with respect to at least some customer class in the tying market. Moreover, the tie must significantly reduce competition in the supply of the tied good for all buyers (not merely those

[41] For a discussion of one such theory, see Whinston (1990).

subject to the tie). Furthermore, two-level entry must be measurably harder than entry into the tying product market alone.

It is important to recognize that the products in the two tiers could be provided by separate firms. The firms selling program *B* need not be the same firms that would later enter into production of program *A*. The point is that it would be harder to coordinate if entry into both markets had to be done simultaneously, and without the benefits of economies of scale and scope from splitting production across incumbent as well as entrant systems.

Turning to welfare effects, consumers may be harmed by the lack of entry into the tying good market. Consumers also may be harmed by the reduction in competition in the tied good market.

7.1.6 Non-Economic Rationale for Tying: Make Customers Buy an Inferior Product

It is often claimed that firms with a dominant position in product *A* will force customers to buy its own, inferior, version of product *B*, as condition for obtaining product *A*. Before reaching any conclusions about this practice, one needs to be careful to understand what incentive the seller has to do this. For example, if the two products are used in fixed proportions, the firm controlling product *A* can earn more profits by simply raising the price of *A* and letting consumers pick a superior *B* offered by other firms. We are wary of attacking tying based on this theory unless and until the theory is made more complete by including a coherent explanation of how the dominant firm profits from such tying. In other words, we do not regard this as a stand-alone motivation for tying, although it may make economic sense in conjunction with one of the other theories we have described above (*e.g.*, a firm may tie an inferior piece of software to engage in metering).

7.2 Identifying the Likely Rationale

Some of the motivations for tying and bundling listed above promote efficiency, while others reduce it. It is thus important to distinguish among them. One basis for that distinction is to assess whether the conditions identified above for each rationale do in fact hold in a particular instance. As a practical matter, company documents and testimony will factor heavily into this determination. Another basis for determining which rationale is likely to be the operative one is to examine the nature of the pricing of the software products in question.

If transaction costs savings are the motivation for bundling, then there is no need to "coerce" the buyer. We should see low incremental prices, as opposed to mandatory purchase terms. If quality assurance is the rationale,

then the contract should state actions that will be taken in the event of non-compliance which are consistent with the harm done by substandard components (*e.g.*, void the warranty). For metering and price discrimination, one would expect that the price of the tied good would exceed marginal cost except in very unusual circumstances. Similarly, under both foreclosure motivations, the price of the tied good should exceed marginal cost unless it can be established that the seller is engaging in predatory pricing (but then there is no need to tie the good and in fact the seller would likely be better off not doing so).

7.3 Critique of Legal Standard of What Constitutes Tying

The legal treatment of tying, while generally moving in the direction indicated by economic analysis, remains confused. Historically, the courts treated tying harshly under the antitrust laws, often imposing *per se* liability. In 1949, the Supreme Court stated that tying arrangements "serve hardly any purpose beyond the suppression of competition." (Standard Oil, 337 U.S. 293, at 305). The Court stated as recently as 1969 that tying arrangements "generally serve no legitimate business purpose that cannot be achieved in some less restrictive way" (Fortner I, 394 U.S.495, at 503). In 1984, by a 5-4 margin, the Court rejected a shift in the treatment of tying from *per se* to rule-of-reason analysis (Jefferson Parish, 466 U.S. 2). However, the distinction between *per se* and rule-of-reason treatment has blurred, because the courts are now willing to consider the economic effects of tying arrangements, including their possible efficiencies, before applying the *per se* rule. More specifically, the courts generally will condemn a tying arrangement only if:

1. the tie involves two separate products;
2. the sale of one product is conditioned on the purchase of another;
3. the seller has sufficient economic power in the market for the tying product to restrain trade in the market for the tied product; and
4. the tie is likely to have anticompetitive effects in the market for the tied product.

7.3.1 One Product or Two?

The thorniest of these four conditions is the distinction between one and two products. In Jefferson Parish, the Supreme Court found that anesthesiology services and other hospital services were distinct products, despite defendant's claim that they constituted a "functionally integrated package of services." The Court held that "the answer to the question whether one or two products are involved turns not on the functional relation between them, but rather on the character of demand for the two items." (466 U.S. 2, at 19). The Court asked specifically whether the tie linked products that were "distinguishable in the eyes of buyers." Since anesthesiology services could be purchased separately from the other facilities and services provided by the hospital, the Court found that these were two separate products.[42] We must note that the Court's test does not appear to track any of the economic analysis provided above.

7.3.2 Is the Tied Product Only Available with the Tying Product?

The Courts have also had to grapple with the second element of a tying violation, namely the "conditioning" requirement. Here, as with other vertical restraints, the Courts have clouded the issue somewhat by focusing on whether the sale of the tied product was "coerced" or "voluntary." What does it mean to require or "force" the purchase of the *tied* good? To economists, this is not a fruitful line of inquiry unless "coerced" is given economic content. For example, if sale of the tying product is not expressly conditioned on purchase of the tied product, one could ask whether the incremental price of the tied product is at least as large as its incremental cost. For software, with a short-run incremental cost close to zero, the question devolves to whether the tying product is sold separately, albeit at the same price as the bundle.

[42] Since Jefferson Parish, many lower courts have interpreted and implemented the Supreme Court's test. According to *Antitrust Law Developments* (Fourth), the lower courts have found that "separate products exist (or may exist) where the alleged tie involved repair parts and finished goods, flush doors and six-panel doors, single-chip microprocessor patents and other patents, computer peripheral storage devices and patented interconnect products, realtor multiple listing services and other realtor support services, equipment service and repair parts, and cooperatives and building maintenance services. Separate-product claims have been rejected in cases where the alleged tie was between single family homes and the leased land on which they were built, popular and hard-to-sell car models, mortgage financing and attorneys' fees, pathology services and a hospital's facilities and services, truck engines and warranties on those engines, licenses for popular and less popular game shows, a diagnostic software program and computer maintenance and repair services, the combination of words and photographs in a computerized real estate multilist database, access to a financial database and the terminals used to gain such access, and employment and advertising through a yellow pages directory." (footnoted omitted) We have had difficulty discerning a meaningful economic pattern from these decisions.

7.3.3 Economic Power Over the Tying Product

How much economic power over the tying product is required to trigger *per se* condemnation of tying? In Jefferson Parish, the Supreme Court summarized its prior rulings on market power,[43] saying that "we have condemned tying arrangements when the seller has some special ability — usually called 'market power' — to force a purchaser to do something that he would not do in a competitive market." (429 U.S. 610, at 13) The Court stated that a large market share, ownership of a patent, or even a unique product could lead to market power. But the Court also stated that a market share of 30 percent did not constitute proof of such market power. (429 U.S. 610, at 26-29) Since Jefferson Parish, something close to monopoly power in the tying product seems to be required to invoke *per se* treatment of tying.

7.3.4 Competitive Effects in the Market for the Tied Product

Requiring a showing of likely anticompetitive effects in the market for the tied product is very close to insisting upon a full rule-of-reason inquiry. Some (but not all) Courts have indeed gone in this direction, as have the U.S. Department of Justice and the Federal Trade Commission in their *Antitrust Guidelines for the Licensing of Intellectual Property* (issued April 1995). According to Section 5.3 of these guidelines: "The Agencies would be likely to challenge a tying arrangement if:

1. the seller has market power in the tying product,
2. the arrangement has an adverse effect on competition in the relevant market for the tied product, and
3. efficiency justifications for the arrangement do not outweigh the anticompetitive effects."

An open legal question is whether plaintiffs must show that the defendant is likely to achieve market power in the tied product market, as would be needed for the foreclosure motivations to be profitable.

7.3.5 Summary

Many readers will note that the legal tests used to identify and condemn tying line up poorly at best with the economic theories of the effects of

[43] The Supreme Court ruled in 1958 that monopoly power is not required for a tying offense, but rather whether "a party has sufficient economic power with respect to the tying product to appreciably restrain free competition in the market for the tied product." (Northern Pacific Railway, 356 U.S. 1, at 6) This standard was explored in depth in 1977 in Fortner II (429 U.S. 610), where the Court asked whether the defendant had "appreciable economic power" in the market for the tying product.

tying. Most notably, while economists naturally look at the overall impact of tying on the total cost to consumers of the tying and tied products, the law tends to compartmentalize the effects of tying. This tension between the legal and economic analysis of tying is by no means confined to software markets.

7.4 Tying in the Software Industry

How do software ties and bundles compare with these practices in other industries? Consider a "standard" tie, such as a copier and paper. This tie has a number of features: there are variable proportions; it is unlikely to satisfy the tied-industry profitability condition for foreclosure; and two-step entry is unlikely to an important factor given the technological differences. All of this suggests that metering is by far the most likely rationale.

In contrast, consider a software license that bundles two distinct programs. The distinctive features of software include: near zero incremental costs for seller; free disposal by purchasers in many cases;[44] typically fixed proportions; and often very strong complementarities (*e.g.*, between the operating system and an applications program). Under these conditions, tying is unlikely to be motivated by metering, and more likely to be a way to reduce transactions costs in distribution and to offer improved functionality (by code integration), to assure quality, to capture oligopolistic rents in the tied market, or to deter two-level entry.[45]

In the software setting, it is especially important to distinguish practices that simply make the program *B* available at a low (or zero) incremental price, from practices that impose an incremental cost on customers who use *rival* versions of program *B*.[46] Unlike giveaways of program *B*, these latter practices — which include designing program *A* to work poorly with rival versions of program *B* and imposing contractual provisions that limit customers from using rival versions of program *B* — directly harm consumers.[47]

[44] This depends in part on whether the packaging is purely commercial or involves the integration of code.

[45] This analysis changes if software publishers adopt a per-use licensing structure, in which case tying can be used to support metering along various dimensions related to the use of the tied product.

[46] This distinction is explored in depth in Gilbert and Shapiro (1998).

[47] See our discussion of exclusive dealing below.

What does this analysis tell us about Microsoft's bundling of Windows and Internet Explorer? Microsoft is claiming a transactions cost basis for the bundling in asserting that an integrated product enhances consumer value. Microsoft's opponents are claiming foreclosure, motivated by the goal of dominating the provision of browsers and, more significantly, blocking two-stage entry into operating systems following entry into browsers.

The proper statement of the policy problem is this: Is there a rule that, when applied consistently to actual markets by real policy enforcers, can weigh these competing claims? Such a rule is extremely difficult to craft, as are workable remedies. Clearly, there are serious problems with any policy that freezes the definition of an operating system. Microsoft's chief operating officer put it this way: "The principle at stake in this case is whether Microsoft — and every other software company — has the right to continually [sic] improve its products and add new innovations for consumers."[48] While we, too, are wary of any rule that stifles innovation, we do not accept this formulation of the Microsoft case. First, the statement assumes that rules applicable to Microsoft will automatically apply to other software companies lacking monopoly power; antitrust quite generally imposes selective limits on the conduct of dominant firms. Second, we distinguish Microsoft's contracting practices, which limit customers' choices and are the subject of the current dispute over compliance with the consent decree, from Microsoft's product design decisions. Microsoft's recent decision to revise its dealings with Internet Service Providers to permit ISPs to promote rival browsing software[49] is an excellent example of how contracting practices can be modified — either voluntarily or by court order — without delving into issues of product design.

Looking forward, the debate will likely turn to the question of the actual integration of code. These issues will be much more troublesome. One approach will be to let Microsoft and other software producers engage in any packaging that they want as long as the different programs are sufficiently entangled. Such a policy would do little to limit packaging. Alternatively, antitrust authorities could pursue a policy of requiring a modular approach to the production and sale of code, with well-defined, open interfaces between the modules. While in some ways attractive, such an approach clearly raises a thicket of thorny questions including: Who will define the scope of the individual modules? How will "openness" be defined and monitoring be conducted to enforce openness on a timely basis? How will we know that

[48] Robert J. Herbold quoted in "Microsoft Appeal Says U.S. Case is Flawed," *The New York Times*, January 30, 1998 at C2.

[49] "Microsoft Offers to Change Deals," *The New York Times*, March 3, 1998, at C7.

important economies of scope are not being lost? And what will happen to the incentives to innovate?

8. EXCLUSIVE DEALING

In addition to bundling its own products, a firm may attempt to package its products with those of *other* firms. More specifically, a software vendor might enter into an agreement with a personal computer manufacturer that the latter will ship only machines containing the operating system provided by that software vendor. Like tying, such exclusive dealing has been the subject of numerous antitrust disputes over the years.

To obtain the effects of exclusivity, there does not have to be an explicit agreement requiring exclusivity. Instead, preferential pricing for "loyal" customers, as well as certain types of quantity discounts, can be at least partial substitutes. One key Microsoft tactic addressed in the consent decree, for example, levied a per-processor license, which essentially made the marginal cost of actually putting the Microsoft operating system on a machine zero to a personal computer OEM who had signed such a license.[50] A similar effect could have been achieved by selling licenses on a lump-sum basis. Presumably Microsoft chose not to do the latter because it offered less opportunity to meter the value of the license.

Exclusive dealing with computer OEMs is much like traditional exclusive dealing to tie up distribution channels in other industries. We note here that there appears to be less reason to have exclusive dealing to create incentives for retailer support and promotion activities than is typical in many other industries (*e.g.*, fast food). We also note that, in the presence of network effects, exclusive dealing may be particularly harmful to competition because it can promote tipping.[51]

9. PREDATORY PRICING

Economists tend to dismiss claims of predatory pricing. Their reasoning is summarized by the old joke about how one conjugates the verb "to compete." The answer is: "I compete, you predate, they predate..." Several

[50] See Gilbert (1998) for an analysis of the 1995 Microsoft Consent Decree with the Justice Department.

[51] For an extensive analysis of exclusivity in network markets, see Balto (1997). See also Shapiro (1996a).

of the structural characteristics of software markets that we identified at the start of this paper give reason to be somewhat more concerned about predation in software markets than in other markets. At the same time, these conditions make it difficult to identify predation. And, as in all markets, there is a danger that remedies aimed at predation actually will have the effect of blunting legitimate competition.

One of the key structural features is the existence of network effects. The presence of network effects means that it may actually be profitable to engage in predation because once the rival has been put at a sufficient disadvantage in terms of actual and anticipated installed base, it may be impossible for that firm to compete effectively in the future. In other words, the prospect for recoupment of losses sustained in below-cost pricing, strongly emphasized in recent Supreme Court rulings on predatory pricing, can well be present in network markets. This possibility remains even if the target firm's initial investments are largely sunk.

The ability of a software supplier to engage in price discrimination to go after a particular group of users can reduce the cost of predation and thus make it more likely. For example, a firm might offer special deals to those users who switch away from the software of a rival.

While network effects and price discrimination make predatory pricing more likely, the next feature makes predation hard to measure. The marginal cost of a copy of a program is near zero. Hence, even giving away software will come close to meeting the requirement that price be no lower than marginal cost (or even average variable cost) in order to be considered non-predatory.[52] In the light of network effects and legitimate incentives to engage in penetration pricing to promote a network, even negative prices might not be predatory.

10. SHOULD ANTITRUST AUTHORITIES STAY ON THE SIDELINES?

We have emphasized that the antitrust analysis of software markets can be extremely complex. Moreover, there typically will be significant uncertainty over the future evolution of software markets. Do these factors imply that the antitrust authorities should refrain from intervention until they know all the answers?

[52] Furthermore, if the marginal cost is very small, insisting on a positive price can raise the cost of distribution significantly because it makes it necessary to track sales and collect payments. The distribution of software over the Internet would be much more cumbersome, for example.

We do not believe so. Markets with large production and demand-side economies of scale are prone to tipping. Dominance, once achieved, may be very hard to unwind. Doing so would either require the coordinated movement of lots of consumers — with the possibility that they would have to incur significant switching costs — or forcing open a network, which we have seen poses its own substantial set of problems. As the AT&T divestiture reminds us, breaking up a large, integrated company is a massive undertaking with its own substantial risks. Technological bundling also is extremely difficult to reverse. It is much like unscrambling the omelet. Witness the controversy surrounding whether Microsoft can easily separate the Windows operating system from its Internet Explorer.

These two considerations support a proactive policy. In many ways, this is like the theory underlying merger policy. Rather than let all mergers go through and then examine whether competition is harmed, some mergers are blocked before they can be consummated, even though their effects are difficult to predict in advance. The rationale for this approach is that it would be too difficult to undo those mergers that later turned out to harm competition.

In the end, we believe that targeted government intervention, based on established legal principles and accounting for the fundamental economic characteristics of the software industry, is fully supportable. This does not mean that the mere threat of tipping is justification for bringing an antitrust action against Microsoft. But nor does it mean that the Justice Department and the Federal Trade Commission should stay on the sidelines merely because software markets are complex and dynamic.

11. CONCLUSION

The software industry is widely seen as a wellspring of innovation, not to mention a source of U.S. export revenues. The fact that software markets are dynamic and complex does not imply, however, that these markets are immune from antitrust. Rather, it is critical that antitrust policy properly reflect the economic features that characterize the software industry: strong systems and network effects, very low marginal costs, durability, and rapid technological change.

We have systematically developed the implications of these economic features for antitrust policy in the software industry. Generally, we find that the application of antitrust economics to software mirrors its application in other markets. Merger policy is on a sound footing, the treatment of collaboration among competitors is generally sensible but may chill some

pro-competitive cooperation, and the treatment of tying and bundling is close to incoherent.

What are the implications for the treatment of Microsoft? We believe that Microsoft has economic power and antitrust authorities have valid reasons to limit Microsoft's conduct, whether it be exclusionary practices or anticompetitive acquisitions. But there are very real limits on the ability of antitrust law to reign in Microsoft. The government and the courts are on far stronger ground challenging Microsoft's contractual practices, to the extent they create entry barriers or constrain consumer choice, than attacking Microsoft's product design choices. With the introduction of Windows98 we may find out just where the solid ground ends and the swamp begins.

Whatever the outcome of the current dispute between Microsoft and the Justice Department, there is no need for a new antitrust policy. Our antitrust laws have proven flexible enough to handle new industries, with new economic features, before. We believe that antitrust enforcers also can handle the software industry, aided, we hope, by recent advances in economists' understanding of the strategic implications of compatibility, interfaces, and network effects. We simply need to continue along the path of increasing sophistication taken by the U.S. Department of Justice, the Federal Trade Commission, and (one hopes) the courts.

References

Anton, James and Dennis Yao, "Standard-Setting Consortia, Antitrust, and High-Technology Industries," *Antitrust Law Journal*, vol. 64, pp. 247-265, 1995.

Arthur, W. Brian, "Competing Technologies, Increasing Returns, And Lock-in by Historical Events," *The Economic Journal*, vol. 99, pp. 116-131, 1989.

Balto, David, "Networks and Exclusivity: Antitrust Analysis to Promote Network Competition," Federal Trade Commission, April 1997.

Bulow, Jeremy I., "Durable-Goods Monopolists," *Journal of Political Economy*, vol. 90, pp. 314-332, 1982.

Coase, Ronald, "Durability and Monopoly," *Journal of Law and Economics*, vol. 15, pp.143-149, 1972.

Davidson, C. and R. Deneckere, "Incentives to Form Coalitions with Bertrand Competition," *Rand Journal of Economics*, vol. 16, pp. 473-486, 1985.

Dybvig, Philip H., and Chester S. Spatt, "Adoption Externalities as Public Goods," *Journal of Public Economics*, vol. 20, pp. 231-247, 1983.

Economides, Nicholas and Lawrence J. White, "Networks and Compatibility: Implications for Antitrust," *European Economic Review*, vol. 38, pp. 651-662, 1994.

Farrell, Joseph and Michael Katz, "Antitrust, Intellectual Property Law, and Standard Setting," *Antitrust Bulletin*, forthcoming.

Farrell, Joseph and Garth Saloner, "Standardization, Compatibility, and Innovation," *Rand Journal of Economics*, vol. 16, pp. 70-83, 1985.

Farrell, Joseph and Garth Saloner, "Installed Base and Compatibility: Innovation, Product Preannouncement, and Predation," *American Economic Review*, vol. 76, pp. 940-955, 1986.

Farrell, Joseph and Carl Shapiro, "Horizontal Mergers: An Equilibrium Analysis," *American Economic Review*, vol. 80, (1990), pp. 107-.

Gilbert, Richard, "Networks, Standards, and the Use of Market Dominance: Microsoft," in J. Kwoka and L. White, eds., *The Antitrust Revolution: The Role of Economics*, Oxford University Press, forthcoming.

Gilbert, Richard, and Carl Shapiro, "Antitrust Issues in the Licensing of Intellectual Property: The Nine No-No's Meet the Nineties," *Brookings Papers on Economics: Microeconomics*, forthcoming.

Katz, Michael and Carl Shapiro, "Network Externalities, Competition and Compatibility," *American Economic Review*, vol. 75 (3), pp. 424-440, 1985.

Katz, Michael and Carl Shapiro, "Technology Adoption in the Presence of Network Externalities," *Journal of Political Economy*, vol. 94, pp. 822-841, 1986a.

Katz, Michael and Carl Shapiro, "Product Compatibility Choice in a Market with Technological Progress," *Oxford Economic Papers*, vol. 38, pp. 146-165, 1986b.

Katz, Michael and Carl Shapiro, "Product Introduction with Network Externalities," *Journal of Industrial Economics*, vol. 40, no. 1, pp. 55-84, 1992.

Katz, Michael and Carl Shapiro, "Systems Competition and Network Effects," *Journal of Economic Perspectives*, vol. 8, no. 2, pp. 93-115, 1994.

Klein, Joel I., "The Importance of Antitrust Enforcement in the New Economy," available at http://www.usdoj.gov/atr/public/speeches/1338.htm, 1998.

Lemley, Mark and David McGowan, "Legal Implications of Network Economic Effects," 86 California Law Review 479, 1998.

Liebowitz, S. J. and Stephen E. Margolis, (1990), "The Fable of the Keys," *Journal of Law and Economics*, vol. 33, no. 1, pp. 1-26, 1990.

Rohlfs, Jeffrey, "A Theory of Interdependent Demand for a Communications Service," *Bell Journal of Economics*, vol. 5, no. 1, pp. 16-37, 1974.

Shapiro, Carl, "Antitrust in Network Industries," available at http://www.usdoj.gov/atr/public/speeches/shapir.mar, 1996a.

Shapiro, Carl, "Mergers with Differentiated Products," *Antitrust*, Spring, pp. 23-30, 1996b.

Varian, Hal, "Price Discrimination," in *The Handbook of Industrial Organization*, R. Schmalensee and R.D. Willig (eds.), Amsterdam: North Holland Publishing, 1989.

Veblen, Thorstein, *Theory of the Leisure Class*, 1899.

Werden, Gregory, and Luke Froeb, "The Effects of Mergers in Differentiated Products Industries: Logit Demand and Merger Policy," *Journal of Law, Economics and Organization*, vol. 10, pp. 407-426, 1994.

Whinston, Michael, "Tying, Foreclosure, and Exclusion," *American Economic Review*, vol. 80, pp. 837-859, 1990.

COMPETITION, INNOVATION AND THE MICROSOFT MONOPOLY: ANTITRUST IN THE DIGITAL MARKETPLACE

Proceedings of a conference held by
The Progress & Freedom Foundation in Washington, DC
February 5, 1998

edited by

Jeffrey A. Eisenach and Thomas M. Lenard
The Progress & Freedom Foundation

KLUWER ACADEMIC PUBLISHERS

Contents