

of conduct matter, then it is through their influences on access that such effects would occur. We maintain that the first prong of our proposed test would provide appropriate discipline to the enforcement community to resist the temptation to second-guess and intervene in business decisions that are unlikely to disable competitive, efficient firms. We correspondingly maintain that adherence to our test would provide an appropriate spur for attention, instead, to mutually beneficial and serious business-like negotiations concerning genuine needs for access to OSS bottlenecks and what would constitute appropriate compensatory terms.

Chapter 7

Comment on Ordoover and Willig

Shane M. Greenstein
Northwestern University

1. INTRODUCTION

First, a disclosure: I have no financial interests in the present Microsoft antitrust suit with the Department of Justice. Indeed, I have never had any financial stakes in any of Microsoft's legal battles, except that I use some of their products.

A second disclosure: I have a peculiar perspective on the present case. That is because I teach commercialization of technology to MBAs. They all want to know how they can get monopolies just like Microsoft. I sometimes wonder if I do more damage to the US economy by teaching students how to obtain monopolies or if I undo that damage by teaching them how to compete against monopolies.

OK, enough disclosures. Though I am not a lawyer, I am a market analyst of information technology industries. Recently I have found myself taking part in conversations about antitrust law. Though I am familiar with some of the nuances of the field, I must confess that it feels a lot like walking in on an on-going conversation between friends, where the conversation is primarily about its own history.

Lawyers focus on whether behavior in the software industry fits into the present legal definitions for foreclosure and tying. This slant tends to cast antitrust issues as a question about judicial edicts over forbidden business tactics. While that focus is fine for some purposes, and I will pay some homage to it in this comment, it is a narrow base from which to begin a

broad discussion about competition policy in innovative markets, as in information technology markets.

The Ordo and Willig essay presents a very general and quite provocative analysis of vertical issues. It is my role to elaborate on some of the key questions behind their paper. At times it may appear as if I am only complicating their elegant essay by bringing up many messy details on how the computer industry works in practice. Mostly, however, I will simply emphasize one structural feature which lies at the heart of their questions about vertical relationships in information technology industries: when firms innovate and commercialize technology, firms act as both partners and competitors at the same time. Large firms do this and so too do small firms. The question which naturally arises is whether the same rules of conduct for vertical contracting ought to apply to both small and large firms.

Let me be plain about my views: In industries characterized by lots of experimentation, such as computing, I do not think the same rules ought to apply to both large and small firms. However, this principle is easier to express than implement. I am not sure it is feasible to enforce in practice in this industry.

2. PUT THE DISCUSSION IN CONTEXT

To understand where foreclosure policy matters in the long run, it helps to appreciate what emerges during the commercialization of new information technology, particularly software.

Product cycles under conditions of weak intellectual property protection determine most commercial behavior in IT. Firms prototype new functions, beta-test on big users, market and sell their products, with little hope of avoiding rapid imitation. Firms improve their own products frequently, imitate others when they can, and develop their own ideas if they must. No technical lead is lasting, and the only path to success involves frequent experimentation and repeatedly being fast to market.

It is remarkable that despite this fierce competition and all this experimentation, the same types of technological arrangements tend to show up as key components of most user's systems. These arrangements are often called platforms.

Platforms arise because both users and software vendors make platform specific investments in such things as training, customized software, programming skills, programming tools or software libraries. Operating systems change slowly for this reason. The same story applies to communication protocols in networks or formats for digital storage of different types of media.

I am not saying that a single firm must own all the parts of a platform. A single firm can act as the primary supplier for most parts of a dominant platform, as IBM did in mainframes for many years and DEC did in superminis until recently. Many suppliers, such as Intel and Microsoft, can supply parts for a popular platform. In some speciality markets, there will also be other firms who offer key components, as occurs today in most client/server networking arrangements. IBM, Intel, Sun, Cisco and Oracle all dominate particular component markets of common arrangements.

In every era of the computing market there have been technical leaders who control the development and sale of unique assets at the heart of popular platforms. This pattern alone can raise some antitrust issues, but its combination with other factors usually leads to vertical issues, which is what I will discuss today.

The key thing to note is that, except in rare circumstances, it is much easier and cheaper for all software developers and users to develop new products on an existing platform than to build an entirely new one. Thus, suppliers of established software can often add new functionality at a lower cost than competitors. If competitors bring new functionality to market in the form of new software, they do it by providing compatible software.

Let me illustrate with an example that comes from out of the limelight. A few years ago Baxter and TSI revised their EDI application for hospitals - programs called Value-Link and OnCall, respectively. It was a large undertaking, and a risky product launch that took a good two years to get off the ground. After a few years of experimentation and marketing, it diffused to thousands of hospitals nationwide, helping these organizations reduce costs of supplies.

The important observation is this: It is not surprising that TSI used Windows and Access as part of their system; they had to write more programs for the front and back end of the whole system. What would have been the point of writing a new operating system and database for PCs when these developers had enough to do on other parts of the whole application?

Many such examples exist. The situation is endemic to networked PCs or new developments in Internet-based applications.

Vertical issues arise as a result of these complex combinations of software. Manufacturers fight to control the pivotal parts of a platform because it influences their rights to modify the old arrangements with new functionality. Tomorrow's revenue depends on retaining these rights.

The fight is subtle and tactics change frequently. Firms look to emulate each other's experiments, steal another firm's vision if they must, and induce users to switch to their products. They do this by integrating many functions into one product or by offering new products to loyal customers of their product. If they are not successful with a crucial part of a complex bundle of

applications, they must find another partner, relying on either a joint venture or licensing deal for the missing pieces.

These fights have various repercussions for users. First, users are offered frequent opportunities to upgrade to new functionality. Second, users end up with programs which contain overlapping functionality or many more features than they need. Third, if a joint venture or licensing deal was not planned or well executed, users may end up with a confused after-sales support system. Fourth, users may find that the vendor of one product tries to sell them a product from another developer because the two firms have a joint marketing arrangement. Fifth, sophisticated users end up spending much time integrating programs from different vendors or versions that are potentially incompatible.

It is not surprising that legal issues arise as a by-product of these business and technical fights and these cooperative deals.

Let me be concrete. Microsoft has rights to alter one key component in a popular platform. They are in an excellent position to bring out new functionality in applications closely related to that platform. If they do not, then someone else who makes software compatible with Microsoft's programs will do so soon enough. We can expect occasional fights between Microsoft and other firms trying to diffuse a compatible application with new functionality. It is not at all surprising that a firm such as Microsoft makes deals to distribute its products through many channels and, if it can, limit the distribution of its rivals. It is not at all surprising that users sometime are confused about the supply of products and that Microsoft might try to simplify their supply arrangements. And it is not an odd feature of this market that one product's sale, such as a web browser, influences the sale and customer experience in other areas, such as networking and Internet applications.

This is an appropriate time to mention, as an aside, that some commentators in the industry's trade press complain that Microsoft makes money by copying Netscape's ideas. While this complaint raises many issues, it is usually mischaracterized in popular fora. First of all, the Netscape/Microsoft battle is neither the first instance in which Microsoft has copied a rival to some extent, nor is Microsoft the only firm to practice this art. Arguably, Microsoft is one of most profitable practitioners of this art (for all kinds of reasons), but that is another matter. Simply put, most firms learn from each other's experiments, though not all firms profit from their learning. It is a fact of life in software. Second, both parties are guilty of this sin. Netscape did not handsomely reward the University of Illinois, where Mosaic, the first popular browser, emerged from the university's laboratory. This is where Marc Andreessen, Netscape's chief technology officer, was first exposed to experimental browser development.

More to the point, in thinking about antitrust issues nobody benefits from a debate about who invented what and when they invented it. These issues are really the domain of intellectual property policy, not antitrust policy, which puts them outside of the jurisdiction of the Department of Justice. It is still interesting policy, but it is another issue for another day.

3. WHY THIS IS A REALLY HARD POLICY PROBLEM

Nobody should have any problem with an established firm that continually experiments and tries to enhance its own products. Everyone benefits from such behavior.

The confusing situation arises when entrants develop new products or variations on popular products. Entrants usually need to partner either explicitly or implicitly with established platform providers to sell their product. If established platform providers always cooperated, either explicitly or implicitly, there would be no issue. More to the point, when cooperation is refused, sometimes the lack of cooperation seems justified and within any reasonable bounds for business behavior. Sometimes it is not, and defining those boundaries is difficult, especially when products are redesigned yearly.

Cooperation is usually forthcoming when established firms are interested in enhancing their own platforms with a new entrant's capabilities. In that case, cooperation may take many forms — sharing technical specifications, encouraging use of "beta versions" of new releases of software, offering licensing deals, and joint marketing arrangements. Many large firms regularly do these things. For example, Microsoft had no particular problem with Baxter and TSI's development efforts of EDI applications for healthcare, since it led to greater use of Microsoft products in hospitals and Microsoft has no short-term strategic interest in this type of application. Indeed, Microsoft supports an elaborate network of application developers, of which TSI is only one.

However, most firms, and Microsoft is no different than any other in this respect, will not make deals when it potentially threatens the value of their assets or their control over the assets in the future. If a new entrant's capabilities threaten the established firm's future plans, the firm will naturally use licensing deals and joint marketing arrangements as tools to limit their rival's business opportunities.

An example of these ambiguities has been in the news recently: Microsoft's dealings with Sun's Java language. This is a product with lots of promise but whose basic efficacy has yet to be proven. Microsoft was not

too keen on making Windows compatible with Java, because Java could provide functionality that Microsoft would also like to provide with different technology sometime in the future. It was widely publicized that Microsoft balked for quite awhile at signing a licensing deal with Sun.

Eventually they did sign, and, as I recall it, there was quite a lot of speculation about why they were doing it. Some industry analysts speculated that Microsoft signed because they wanted the additional functionality on their system, because Netscape would also have it. Other analysts speculated that Microsoft feared being isolated from a product which every other major developer in this industry supports. By being compatible, they would more easily learn to imitate.

Here is my point: Almost by definition this cooperative arrangement has tension built into it. Microsoft is now trying to modify Java just enough to benefit users and also hinder Sun's platform development strategy. It is in Sun's interest to have Java work the same on windows-based systems and Unix based systems, even if achieving that compatibility slows the development of new applications on Java just a bit. Accordingly, Microsoft is trying to raise doubts about the compatibility of Java applications across platforms. As a result, Sun is upset with Microsoft and claiming breach of contract. Even if these firms patch things up, this soap opera will continue until Java's commercial success is either assured or impossible.

So the ambiguity for policy is this: New entry is good, but in practice it requires cooperation from the potential competitor. In this case, in order to enter this market with something that has new features, some of which threatens some part of Microsoft's assets, Sun has to get Microsoft's on-going cooperation in the development effort. I have no idea whether Java will ultimately be a good idea or not for users, but it is certainly in society's interest to let Sun try. However, Sun will probably never have an opportunity to try the experiment they would like to do because Microsoft is going to hinder it at every step along the way.

More concretely, to diffuse a new application, all software developers prefer to have the cooperation of Microsoft, the very same firm with whom some of them might soon be competing. When Microsoft decides not to cooperate, at what point is that lack of cooperation outside of acceptable bounds of business behavior?

So it is that Microsoft sometimes encourages entry, offering new bundles of services, even giving away something because it helps their sales elsewhere. Sometimes they want to borrow ideas from their partners, making alliances with ulterior motives so they can get into the business themselves. And sometimes they do not want any partners at all or they want to discourage former partners from becoming rivals. All of these actions are allowable competitive tactics when they lead to new products,

more innovation and lower prices. Conflicts between Microsoft's interests and society's interests only arise in the latter situations when it discourages entry.

To summarize: it does not matter whether new commercial experiments come from established firms or new firms. Society benefits from exposure to new combinations of products. If foreclosure policy can make a difference, it can do so by altering the behavior of a firm who controls a key asset for most experiments. In other words, foreclosure policy renders it less difficult for a wide variety of experiments to make it into user's hands.

A sound competition policy should allow low-cost commercialization. If established firms do this well, then fine. After all, established firms act in the interest of society when they market new enhancements to their old products. In this respect, Microsoft gets a bum rap from the popular press these days. As a large developer of new technology in server and networking markets and in the PC industry, they have brought great advances to many users over the years. While it is difficult to pity a company with Microsoft's capitalization and with Microsoft's reputation for ungracious business tactics, I must concede that I react sympathetically to their claim that the present legal climate and the prospects of frequent judicial review run the risk of hindering their ability to successfully innovate in the future as they have in the past. Any competitive benefits associated with an excessive witch-hunt on Microsoft would be swamped by the misallocation of resources within Microsoft away from innovative activity.

Yet, there are also two sides to this story. Technology markets are inherently uncertain and users benefit from experimentation, even when much of it fails. This is because the economic benefits from one major commercial innovation are so large and one failure teaches others, who then try something new and so on, which eventually leads to commercial success. Experimentation only thrives with multiple points of entry. Occasionally an experiment from an unexpected corner rockets to prominence and threatens the established order of business. A sound competition policy should, therefore, prevent large established firms with unique assets, such as Microsoft, from shielding themselves from the sharp spur of competitive threats, even those with a low probability of succeeding. It is not in society's interest to have a single firm delay or hinder diffusion of new products or any other commercial experiment. In this respect Microsoft, or any other large firm in this industry, deserves additional scrutiny that we do not apply to small firms. Interesting commercial experiments from unexpected corners can be suppressed by large firms who use joint-ventures, exclusive marketing arrangements, bundled pricing and bundled marketing in ways that small firms cannot.

The most troubling and also most difficult questions are not about whether — according to some legal norm — a particular action by itself, say integrating new functionality into an operating system, is or is not tying. This strikes me as a narrow legal reading of the Microsoft/Netscape case and not a fruitful starting point towards a durable policy on vertical foreclosure in information technology markets. The interesting and useful issues are whether antitrust law can be used to prevent a platform provider from using bundling, along with exclusive deals and other vertical contracts, to make it unnecessarily difficult for other competitors to develop, commercialize and distribute a product that threatens the assets of the established firm. It might be possible to develop such an argument in some particular cases, but it is not easy.

It is difficult because it involves a backward-looking evaluation of strategies that were formulated under conditions of technical and market uncertainty, in other words, before they succeeded or failed. This type of policy exercise has all the dangers of 20/20 hindsight. And what if the Department of Justice finds anti-competitive behavior? The next step involves remedies that compel cooperation in the form of a contract or license that was not otherwise forthcoming. For example, imagine forcing Microsoft to license out their APIs at reasonable rates, as some commentators have suggested. If Microsoft is uncooperative, as one would expect, then this is a virtually impossible edict to implement in practice.

As a final disclosure, I should add that I am not yet convinced one way or another about the merits of a general anti-trust case against Microsoft, nor about the appropriate action to take in today's world. Like any technical user I get upset when some online application only works with Microsoft's Explorer and not with Netscape Navigator, but I also recognize that this is not necessarily an example of an exclusionary action (and, of course, I have seen it the other way around). Also, I know that the proper policy here requires a complex and careful evaluation of many actions — selective access and denials of APIs to developers, strong-arming OEMs, and other details — which may or may not have made experimentation and commercial entry unnecessarily difficult for others.

While I have focused on Microsoft and Netscape to illustrate the argument, these issues are more general than just the particular circumstances associated with today's legal battle. I take it as axiomatic that past progress was only a prologue. There will be future entrants marketing the browser of their day — it will be voice recognition technology or reliable Internet telephony or something else I cannot imagine. These same issues will arise again during other periods of experimentation, possibly with these or other firms.

Chapter 8

Microsoft And Browsers *Are the Antitrust Problems Really New?*

Lawrence J. White
New York University

1. INTRODUCTION

The question of Microsoft's market power (if any) and what to do about it (if anything) has attracted a great deal of attention in the political world and the media, as well as from specialists in antitrust policy. It has also generated significant antitrust litigation. This conference by the Progress & Freedom Foundation is a testament to the convergence of all of these strands of attention.

A common claim in the media is that the software industry has special characteristics that create special problems for antitrust policy and enforcement. The antitrust laws were written a century ago for application to "smokestack industries" (the argument goes), whereas modern information-based industries, such as software, represent a new set of problems. Issues of "network effects" and "network externalities", interacting with economies of scale, are claimed to complicate the analysis.

This paper will address those claims, as well as discussing the specific recommendations set forth in the accompanying paper by Janusz Ordover and Robert Willig. An important point of my paper is that the antitrust issues that surround Microsoft are not new nor are they difficult to understand, although they may be difficult to adjudicate in a world of inadequate information.

To set the stage for what follows, let us establish some important terminology: One set of important antitrust issues focuses on a firm's