

Shane Greenstein

*Kellogg Graduate
School of
Management*

*Northwestern
University*

*s-greenstein1@
nwu.edu*

Ties that bind: foreseeing foreclosure

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

A second disclosure: I have a peculiar perspective on the present case. I teach commercialization of technology to MBAs. All my students want to know how they can get a monopoly just like Microsoft's. I sometimes wonder if I do more damage to the US economy by teaching students how to obtain monopolies or by teaching them how to compete against them.

OK, enough on 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 ongoing 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 (see box). This slant tends to cast antitrust issues as a question of judicial edicts over forbidden business tactics. While that focus is fine for some purposes, it is a narrow base from which to begin a broad discussion about competition policy in innovative markets.

The one structural feature lying at the heart of vertical relationships in the IT industry is this: when firms innovate and commercialize technology, they act as both partners and competitors at the same time. Large and small firms alike do this. Although the current discussion focuses on the behavior of large firms, I can anticipate the layman's response—that the same rules

ought to apply to any size firm. In industries characterized by lots of experimentation, I personally don't think the same rules ought to apply to both large and small firms. However, this principle is easier said than done.

Putting the discussion in context

Understanding what foreclosure policy is, and how it counts in the long run, means appreciating what typically emerges during the commercialization of new information technology.

Product cycles under conditions of weak intellectual property protection determine most commercial behavior in IT. Firms prototype new functions, beta test them on big users, and 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 beating everyone else to market.

Remarkably, despite fierce competition and experimentation, the same types of technological arrangements tend to show up as key components of most user 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, and 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 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.

In addition, many suppliers, such as Intel and Microsoft, can supply parts for popular platforms. In some specialty 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 could raise antitrust issues, but its combination with other factors usually leads to vertical issues, which is what this discussion focuses on.

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. A few years ago, Baxter and TSI revised their EDI application for hospitals—programs called Value-Link and OnCall. It was a large undertaking, and a risky product launch, which 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.

It is not surprising that TSI used Windows and Access as part of its system. What would have been the point of writing a whole new operating system when these developers had enough to do on other parts of the 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 old arrangements with new functionality.

Foreclosure
Foreclosure policy
Tying
Vertical relationship

Illegal vertical relationship

Definitions

Actions that exclude rivals
Antitrust laws that prevent monopolies from excluding rivals
Actions of firm with a monopoly that forces other firms to buy related products
Firm-to-firm transaction as opposed to a horizontal relationship (collusion between directly competing firms)
Firm-to-firm transaction that coerces buyers, excludes rivals, abuses monopoly power, and reneges on deals

Tomorrow's revenue depends on retaining these rights.

The fight is subtle and tactics change frequently. Firms 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 unsuccessful 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.

A more concrete example centers on Microsoft's popular platform, Windows. The company is in an excellent position to bring out new functionality in applications closely related to that platform. If it doesn't, 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. It is not at all surprising that a firm such as Microsoft agrees to distribute its products through many channels and, if it can, limits the distribution of its rivals. And it is not an odd feature of this market that one product's sale, such as a Web browser, influences sales and customer experience in other areas such as networking and Internet applications.

As an aside, some commentators complain that Microsoft makes money by copying Netscape's ideas. This is neither the first instance in which Microsoft has done something similar, nor is Microsoft the only firm to practice this art—it is a fact of life in software. For that matter, Netscape did not hand-

somely reward the University of Illinois, where Mosaic, the first popular browser, emerged. More to the point, these issues are the domain of intellectual property policy, not antitrust policy, which puts them outside of the jurisdiction of the Department of Justice.

Cooperation and partners

Nobody should have a problem with an established firm that continually experiments and tries to enhance its own products if everyone benefits, right?

But confusion 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 products. If partners always cooperated (either explicitly or implicitly), there would be no issue. More to the point, sometimes the lack of cooperation seems justified and within 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 software, offering licensing deals, and arranging joint marketing agreements. Many large firms regularly do these things. For example, Microsoft had no particular problem with Baxter and TSI's development of EDI applications for health care, since their

continued on p. 83

