EXHIBIT 2

UNITED STATES DISTRICT COURT

DISTRICT OF MARYLAND

NOVELL, INCORPORATED,

Plaintiffs,

vs.

MICROSOFT CORPORATION

Defendant.

MDL Docket No. 1332

DECLARATION OF ROGER G. NOLL

My name is Roger G. Noll, and I reside in Palo Alto, California. I am a Professor of Economics *emeritus* at Stanford University. I also am a Senior Fellow and the Co-Director of the Program on Regulatory Policy of the Stanford Institute for Economic Policy Research, and a Senior Fellow in the Stanford Center for International Development. A list of my employment history and publications is contained in my curriculum vitae, which is included as Appendix A to this declaration.

My main field of research is public policies toward business, including the economics of antitrust. Before retiring from the Department of Economics at Stanford, I taught graduate and undergraduate courses in the economics of antitrust and regulation. I am the author, co-author or editor of thirteen books and the author or co-author of over 300 published articles, many of which deal with antitrust and/or the information technology sector of the economy.

I have testified on antitrust issues before Congressional committees and the Federal Trade Commission as an independent expert at the request of a committee or the Commission. I have served as a consultant for Congressional committees, the Antitrust Division of the U. S. Department of Justice, the

Federal Trade Commission, the Attorney General of the State of New York, and several private litigants on antitrust matters. I have provided live testimony in court in the following antitrust cases that are still active pending appeal or that have concluded in the last five years.

Metropolitan Intercollegiate Basketball Association vs. National Collegiate Athletic Association (U.S. District Court, New York, New York);

Gordon, et al., vs. Microsoft (Superior Court, Hennepin County, Minneapolis, Minnesota);

Seven Network v. News Limited (Federal Court, District of New South Wales, Sydney, Australia);

In Re Tableware Antitrust Litigation (U. S. District Court, San Francisco);

In the Matter of Adjustment of Rates and Terms for Pre-existing Subscription and Satellite Digital Audio Radio Service (Copyright Royalty Board, Washington, D. C.); and

Bernard Parish, et al., vs. National Football League Players Association (U. S. District Court, San Francisco).

I also testified at an arbitration hearing in a process created by the Federal Communications Commission to resolve disputes over retransmission agreements between Fox television network and multi-channel video programming distribution systems:

Echostar Communications vs. News Corporation.

In addition, I have submitted expert reports and/or been deposed in the following other cases that are still pending or have reached conclusion within the last five years:

National Association of Optometrists and Opticians, et al., vs. Lockyer, et al., (U.S. District Court, Sacramento);

Fran Am Partnership vs. Sports Car Clubs of America (U. S. District Court, Denver);

Intertainer vs. Time-Warner, et al. (U.S. District Court, Los Angeles);

Joe Comes, et al., v. Microsoft (District Court for Polk County, Des Moines, Iowa);

In Re Dynamic Random Access Memory (DRAM) Antitrust Litigation (U. S. District Court, San Francisco);

Brian Bock, et al., vs. Honeywell International (Superior Court, San Francisco);

Vincent Fagan and Anthony Gianasca v. Honeywell International (Superior Court for Middlesex County, Boston, Massachusetts);

John McKinnon v. Honeywell International (Superior Court for York County, Alfred, Maine);

Fleury vs. Cartier International (U. S. District Court, San Francisco);

Eric Seiken vs. Pearle Vision (Superior Court for San Diego County, San Diego);

Jason White, et al., vs. National Collegiate Athletic Association (U. S. District Court, Los Angeles);

In Re Static Random Access Memory (SRAM) Antitrust Litigation (U. S. District Court, San Francisco);

Fair Isaac, et al., vs. Equifax, et al. (U. S. District Court, Minneapolis);

Apple iPod iTunes Anti-Trust Litigation (U. S. District Court, San Jose); and

Minority Television Project vs. Federal Communications Commission (U. S. District Court, San Francisco).

I also was the co-author of an *amicus* submission to the Supreme Court in *PSEG, et al., vs. Riverkeeper*.

ASSIGNMENT

Attorneys for the plaintiff in this litigation have asked me to undertake an economic analysis of the liability issues in this case. Specifically, Plaintiff's attorneys have asked me to determine whether, from the perspective of antitrust economics, Microsoft obtained and/or maintained monopoly power in operating systems through anticompetitive actions in one or more relevant markets associated with applications and middleware products for personal computers (PCs) in which the plaintiff, Novell, participates. I understand that the Novell products at issue in this matter include office productivity software that was acquired from WordPerfect and Borland, and that the Microsoft conduct at issue dates from May 18, 1994. I also have considered software products in the same relevant markets as Novell applications and middleware that are produced by other vendors, operating systems produced by Novell and others, and conduct by Microsoft that took place before May 18, 1994, to the extent that Microsoft's conduct with respect to these products and at prior times is useful for understanding Microsoft's general business strategies in dealing with its competitors and the performance of the markets in which the Novell applications and middleware products were sold after May 1994. For my work on this case, I am compensated at the rate of $700 per hour.

To undertake this assignment, I have read or had summarized under my direction a large number of discovery documents, depositions, deposition exhibits, legal submissions and court opinions in connection with this proceeding as well as *Caldera vs. Microsoft*, *U. S. vs. Microsoft*, the California Microsoft litigation, the Minnesota Microsoft litigation, the Iowa Microsoft litigation, the European Union Microsoft litigation, and the investigation of Microsoft by the Federal Trade Commission. I also have read and considered the implications of the *Findings of Fact* and the decision of the U. S. Court of Appeals in *U.S. v. Microsoft*. I also have read or had summarized for me scholarly and trade press publications on the hardware and software components of the computer industry. In addition, under my direction others have collected and analyzed data about sales, market presence and product performance in computer hardware and software. In this effort I have been assisted by economists at ApplEcon and Bates White. I also have discussed this case with other expert consultants for the plaintiffs, Mr. Ronald Alepin and Dr. Frederick Warren-Boulton, and have read and discussed the expert report that was submitted on behalf of plaintiffs in the this case by Mr. Alepin. I have read and am relying upon Mr. Alepin's report regarding technical issues concerning the hardware and software of personal computers and his descriptions of the actions by Microsoft that undermined the success of Novell's software products. I also have read the expert reports that were submitted in the Iowa antitrust case, *Joe Comes, et al., vs. Microsoft*. Finally, I have relied on the knowledge I have gained in over forty years of experience in studying and teaching the economics of antitrust and the information technology sector.

Because the expert reports and supporting documentation on behalf of Microsoft have not yet been submitted, my analysis and conclusions are subject to amendment and expansion as I consider this additional information.

SUMMARY

The primary results of my analysis are as follows.

Software products usefully can be placed in three categories: operating systems, applications, and middleware. An operating system sits between a computer's microprocessor and other programs that perform various functional tasks. The operating system communicates directly with the microprocessor and exposes applications program interfaces (APIs). Applications and middleware programs communicate with a computer through the operating system's APIs. Applications programs perform the functions that are the source of user demand for personal computers, such as creating, manipulating, displaying and transporting various types of files. Middleware programs both expose APIs for use by applications and communicate with the PC through the APIs of an operating system. Some products perform only middleware functions, but some applications programs also expose APIs and serve as middleware for other applications.

Market definition begins with reference products, which in this litigation are the operating system and graphic user interface (GUI) that were included in Windows 95 and later versions of Windows, plus office productivity applications and middleware that were contained in Microsoft Office and Novell PerfectOffice. Versions of the Novell and Microsoft office productivity products that were on the market in 1994 ran on Microsoft operating systems long before 1994, and periodically these products have been redesigned to run on Windows 95 and latter versions of Windows. PerfectOffice included WordPerfect word processor, QuattroPro spreadsheet, Paradox database, GroupWise messaging, and Presentations presentation software, among other applications, and the PerfectOffice shell, which could be used as a desktop work environment. Microsoft Office included Word word processor, Excel spreadsheet, Access database, PowerPoint presentation, and Microsoft Mail groupware, among other programs. Another important competitor in 1994 was Lotus, which offered messaging and a desktop shell in Lotus Notes, and other office productivity applications in Lotus SmartSuite, including AmiPro (now WordPro) word processor, Lotus 1-2-3 spreadsheet, Approach database, and Freelance presentation software.

In this litigation, the principal market definition issues are, first, to identify the relevant markets that include Microsoft's operating system and GUI, and second, to determine whether other applications and middleware products are sold in different product markets than the operating system and the GUI. Although the court in *U.S. v. Microsoft* determined that operating systems for personal computers (PCs) that use an x86-compatible microprocessor are a relevant antitrust product market, I nonetheless explain why this relevant product market is correctly defined from the perspective of antitrust economics for the purpose of setting forth the underlying logic of market definition analysis for products in personal computer systems.

By following the same method of analysis that was used to define the relevant market for operating systems, I conclude that GUIs that run on an x86 operating system are sold in a separate relevant product market from x86 operating systems. I also conclude that the other products at issue in this litigation, including browsers, word processors, groupware, and other office productivity applications are not part of the relevant markets for either operating systems or GUIs. Defining the relevant markets for each of these products is unnecessary for purposes of determining whether Microsoft used its market power in operating systems and GUIs to disadvantage competing applications and middleware products and thereby to solidify its market power in operating systems. Nevertheless, I analyze the markets for some of these products, and conclude that each is functionally distinct, has independent demand, and is independently produced, and so is sold in a separate relevant product market. The geographic scope of each relevant market is at least the English-speaking world and probably the entire world.

Since the early 1990s, many software products have been sold in bundles. For example, word processor, spreadsheet, presentation, database and groupware originally were sold as separate products, but in the early 1990s began to be bundled into office productivity suites, such as Microsoft Office and PerfectOffice. The Microsoft operating system originally was a stand-alone product (MS DOS), but in Windows 95 was bundled with a graphic user interface, an e-mail application, and several other software products. Internet Explorer was bundled with Windows in an upgrade to Windows 95 that was released in August 1996. Although bundling is ubiquitous, I conclude that, for the products at issue in this litigation, each major component of a bundle is sold in a distinct relevant product market; however, none of my conclusions are affected if the relevant markets are defined as the operating system/GUI bundle and the bundle of productivity applications.

Since May 1994 Microsoft has enjoyed substantial market power in all of the relevant markets at issue in this litigation. In the markets for x86 operating systems and graphic user interfaces, Microsoft enjoys monopoly power. I also conclude that Microsoft enjoys substantial market power in office productivity applications.

A major issue in this litigation is how Microsoft's market power in operating systems could have been maintained and enhanced after May 1994 by a reduction in competition in markets for applications and middleware that run on Microsoft's operating system. To address this issue requires understanding some important economic characteristics of the computer industry.

When Microsoft was preparing to launch Windows 1995, its most important competitors were IBM and Novell. After IBM acquired OS/2 and Lotus, and after Novell acquired DR DOS, QuattroPro (including the Paradox database program) and the software of the WordPerfect Corporation (which included WordPerfect, a messaging application that became GroupWise, and the shell that was later part of PerfectOffice), both firms offered an array of software products that included an operating system, middleware and productivity applications that were comparable, if not superior, to Microsoft's products. In addition, Novell NetWare had the potential to become a "thin client" network operating system in which applications and files could reside on a central computer and be shared by a network of PCs that did not need as complex an operating system as Microsoft's products. Thus, both IBM and Novell posed serious potential threats to Microsoft's monopoly power in PC operating systems.

While the conceptual distinctions among the three types of programs are clear, the precise boundaries are not well defined. In particular, by exposing APIs that improve the functionality of applications, middleware is a potential competitor to an operating system. Likewise, applications programs can expose APIs, thereby competing with middleware and operating systems as platforms for other applications. The indistinct and moveable boundary among types of software is called vertical competition.

Microsoft's market power in operating systems is due in part to the applications barrier to entry. When users buy a new PC they prefer an operating system that supports the applications programs that they have used in the past. "Cross-platform" applications enable users to switch operating systems without switching applications, and thereby facilitate competition in operating systems. But many applications run only on the Microsoft operating system, thereby causing end users to prefer a PC with Microsoft's operating system even if its price/performance ratio is unfavorable compared to other operating systems that do not support these applications.

Because Microsoft knew that cross-platform applications could undermine the applications barrier to entry, some of its actions sought to reduce competition in relevant markets for productivity applications in which Microsoft competed with Novell. At issue in this litigation are anticompetitive acts undertaken by Microsoft as part of this broader strategy but that specifically reduced competition between Microsoft and Novell in office productivity software and messaging software. Microsoft sought to eliminate competition between Microsoft and Novell (and IBM as well) in these products in part to increase the profitability of its productivity applications, but also in part because Novell's and IBM's suite of applications and middleware posed two related threats to Microsoft's market power in operating systems and GUIs.

First, WordPerfect and GroupWise (and Lotus SmartSuite and Notes) were cross-platform applications (that is, they could operate on other PC operating systems that competed with Microsoft). Because office productivity products are extremely important applications, strong cross-platform competitors in these products reduce the applications barrier to entry that protects the dominant position of Microsoft's operating system and GUI.

Second, Borland Office (later PerfectOffice, released in December 1994), WordPerfect Office (later GroupWise), and Lotus Notes contained middleware as well as applications. Borland Office and Lotus Notes exposed APIs, provided software development tools, and included desktop shell. PerfectOffice contained a feature called DAD (Desktop Application Director) which allowed users to create icons for and launch programs that were not part of PerfectOffice. These features allowed applications developers to build applications that could be launched from the desktop environments in WordPerfect Office or Lotus Notes. Users could "live" in Lotus Notes or PerfectOffice, using either rather than Windows as their desktop environment. By 1995 PerfectOffice was installed with and made use of the Netscape Navigator browser (later Corel included Navigator into its WordPerfect Office suite). The combination of Netscape and PerfectOffice was a threat to become an alternative platform for applications programs that would reduce the need for end-users to upgrade their operating system in order to obtain new applications features.

In the face of these competitive threats, Microsoft maintained and enhanced its market power in the markets for operating systems and GUIs through anticompetitive actions aimed at middleware and applications programs that in the mid 1990s posed a threat to the operating system monopoly. An important background to this case is that around the time of the development and introduction of Windows 95, Microsoft engaged in numerous anticompetitive acts that were aimed at a variety of competing products, ranging from operating systems, middleware to applications, many of which were not aimed at Novell. In some cases, such as Internet browsers and media players, Novell did not even offer these types of products; however, because these products were complements to Novell's applications and middleware products, Microsoft's actions to undermine these products affected competition in the markets for Novell's products. The over-arching economic conclusion to be inferred from these actions was that Microsoft would destroy any product that it regarded as a threat to its core business assets, which in 1994 and later included the Microsoft's operating system and GUI.

The actions taken by Microsoft to undermine competing applications and middleware products fall into three broad strategic categories.

The first strategy is to manipulate, withhold and misrepresent the functions and APIs of Microsoft's operating system for the purpose of disadvantaging competitors of Microsoft applications. Microsoft manipulated the content and documentation of Windows 95 and subsequent versions of Windows to undermine the functionality and delay the release of products that it regarded as threatening its core business. Examples that affected Novell are Namespace Extensions, a feature of Windows 95 that was de-documented because Microsoft thought that Novell and Lotus would use these features more effectively than Microsoft's applications developers; printing APIs that Microsoft described and promised would be developed in Windows 95 to support the innovative approach to printing that WordPerfect had successfully implemented in Windows 3.1 but that Microsoft never made functional; and the Messaging API (MAPI), a set of APIs for implementing e-mail and delivering messaging functionality to other applications that Microsoft periodically altered without notice, thereby breaking competing messaging applications and server software.

The second strategy is to tie Microsoft's monopoly products to competitive products. Examples that affected Novell are the decisions to incorporate Internet Explorer, Microsoft's browser, in Windows, and the decision to tie the fully functional version of MAPI to Microsoft's messaging applications, thereby reducing the functionality of competing e-mail programs and server-based messaging systems.

The third strategy is foreclosing market opportunities to competitors. Examples that affected

Novell are disparagement of competitors' products, denial of the right to use of the Windows logo to indicate that a competitor's application product was compatible with Windows 95, and exclusionary contracts for sales of Microsoft products to computer manufacturers, retailers and large customers.

These acts harmed competition in the markets for applications by reducing the functionality of competing applications programs and the choice of products that were available to consumers. From May 1994 through August 1995, when Windows 95 was in the final stages of development, several competing applications, including Novell's and IBM's, and middleware, including Netscape Navigator, were equal or superior to Microsoft's corresponding products in performance and price. Microsoft's actions prevented these products from being upgraded in a timely manner to take full advantage of the new functionality of Windows 95. End-users were harmed by the reduction in competition in applications and middleware markets that was caused by these actions, but they also were harmed by the effect of these actions in the market for operating systems. By reducing competition in applications and middleware, Microsoft harmed competition in operating systems by increasing the applications barrier to entry. These actions provided no offsetting benefit to end users, and so reduced the intrinsic value of, and hence the demand for, Microsoft Windows. Microsoft sacrificed demand for Windows and harmed consumers by taking actions that advantaged its own applications and solidified its monopoly power in operating systems by raising the applications barrier to entry.

Microsoft has no plausible business justification for its actions. The technical acts that undermined the functionality of competing applications were unnecessary and did not improve the functionality or reduce the cost of Microsoft products. Thus, end users received no benefit to compensate them for the loss of functionality and competition that these acts caused.

The basis for these conclusions is contained in the remainder of this report.

THE ECONOMICS OF PC SYSTEMS

To understand the liability issues in this case requires an understanding of the economics of the personal computer industry, including how the technology of PC systems shapes and is shaped by the economic incentives of end-users, hardware manufacturers, and software vendors. Because the economics and technology of PC systems interact, my analysis contains numerous statements about PC system technology and the structure and content of Microsoft's products. Throughout this report I rely on the technical information about computer hardware and software and about Microsoft's software products that is contained in the expert report by Ronald Alepin and the technical expert reports that were submitted in the Iowa antitrust litigation against Microsoft. My summary statements about computer technology and software are drawn from their reports as well as from my background knowledge of the industry from having studied it for over forty years.

PC System Components and Functions

Computers are devices for performing information processing functions, including data analysis, document preparation, electronic communication, Internet browsing, entertainment and transactions. Computer manufacturers (OEMs, for original equipment manufacturers) offer several types of computers, varying in size, speed and internal electronics. The type of computer that is ubiquitous in offices, homes and, in portable versions, travel is the personal computer (PC). PCs are relatively inexpensive (from several hundred to a few thousand dollars) and are sufficiently powerful to perform the computer functions that most people are likely to find useful.

A PC contains electronic devices, called PC hardware, for storing, retrieving and manipulating data and for communicating with peripheral devices, such as a keyboard, cursor control (e.g., a mouse), and printers. The heart of a PC is the microprocessor, which performs the active functions of a PC. Nearly all PCs (over 85 percent during the period at issue in this litigation) contain a microprocessor that is a member of a family of single chip central processing units (CPUs) that was developed by the Intel Corporation. Although Intel is the dominant producer of microprocessors, other companies also have produced CPUs that can support an operating system that was designed to run on a corresponding Intel CPU. Regardless of the identity of the chip manufacturer, Intel-compatible microprocessors commonly are called the x86 class of microprocessors because Intel's early versions had model numbers ending in 86. Intel's brand names for microprocessors in this class include Pentium and Celeron. In the mid-1990s, the only significant alternative to PCs that use Intel-compatible microprocessors was a PC that used a microprocessor manufactured by Motorola, including the PowerPC microprocessor from a joint venture by Apple, IBM and Motorola. These microprocessors were the CPUs for Apple personal computers and, in small volumes, for a few personal computers from other OEMs. In 2006 and 2007, Apple abandoned Motorola microprocessors and migrated its computers to the x86 family. During the period at issue in this case, Apple computers captured at most fifteen percent of the market for new PCs, and usually less.[1]

PCs have enjoyed remarkable technical progress. Two especially important elements of the computer – memory and microprocessors – benefit from Moore's Law, which states that the cost per unit of capacity of a semiconductor product falls by 50 percent every 18 months. As a result of Moore's law, both the memory (the quantity of information that can be stored and accessed) and the speed (how fast a given amount of information can be retrieved, manipulated and stored) of a PC have increased dramatically, thereby greatly expanding the range of functions that a PC can perform: "...software can work its magic only up to the limits of the 'iron.' And it is advances in microprocessor chip technology that drive the personal computer industry."[2] During the past twenty years, the leading edge Intel CPU has evolved from the 386 DX-25 (1988) to the Core 2 Quad (2008).[3] Figure 1 shows estimates of the computing power of the leading microprocessor in each year from 1987 to 2008. During this period CPU

[1] See the discussion of market shares in the section of this report dealing with market power.

[2] Microsoft Systems Software, 1989, Bates No. X000000587636.

[3] Intel microprocessors usually are at or near the leading edge of technology, although other producers have offered microprocessors that held an edge over Intel's chips for a while.

computing power increased by approximately a factor of 1000.

To perform functions on a PC, users need both software and more hardware than just the computer. Together the PC, additional hardware, and software are called the PC system. Examples of additional hardware that most users are likely to use are a mouse, keyboard, printer, audio speakers, visual display terminal and, in recent years, external storage devices, such as flash memory cards and CD recorders. The software that is found today on nearly all PCs includes an operating system, word processor (for preparing documents), Internet browser (for displaying files, including files on other computers that are accessed over a telecommunications network), spreadsheet (for analyzing data), electronic messaging (for communicating with other PC users), presentation (for projecting files on a screen), and media player (for playing audio and video files). In addition, since the mid-1990s nearly all PCs have had a graphical user interface (GUI), which enables users to access programs and perform functions within programs by pointing a cursor to a word or picture on the display screen. GUIs have eliminated the need for users to have even the slightest understanding of how to use a string of alphanumeric characters to issue instructions to a computer.

A layer cake – a series of layers arranged vertically – is a useful perceptual analogy for the combination of hardware and software that constitutes a PC system. The bottom layer is the electronic hardware in the personal computer, including the semiconductor device that acts as the "brains" of the PC, the microprocessor. The next layer is the operating system, which connects to the PC by issuing commands to the microprocessor. The next layer is middleware, which communicates with the microprocessor through the operating system. The top layer is applications, which perform the useful information storage, manipulation, and transportation activities that are the functional purpose of a PC. Applications communicate with the microprocessor through either middleware or the operating system. Except for applications, each layer of the cake is called a platform for the layers above it. Thus, the x86 family of microprocessors is a platform for operating systems that can control a PC that uses an x86 chip as its central processing unit, and the operating system in turn is the platform for the middleware and applications programs that make use of that operating system.

An operating system, the bottom layer of the PC system, is an essential computer program. An operating system is a software product that allows the user to manage the microprocessor and other hardware in the PC system. The *Computer User's Dictionary* from Microsoft Press (Fifth Edition, 1998) defines an operating system as "[t]he software that controls the allocation and usage of hardware resources such as memory, CPU time, disk space, and peripheral devices. The operating system is the foundation on which applications are built." An operating system contains applications programming interfaces (APIs) that other software can use to access the control functions of the operating system.

The other categories of software are applications and middleware. Both applications programs (like a word processor or e-mail) and middleware programs (like a GUI or an Internet browser) use the APIs of the PC's operating system to issue commands to the PC hardware. Applications programs allow the PC to perform the functions that are the primary reason for acquiring a PC, such as to create documents, to manipulate data, and to access the Internet. Middleware refers to software that both exposes its own APIs and makes use of the APIs of an operating system. Some middleware products also are applications, performing end-user functions and exposing APIs for the purpose of being used by or in combination with other applications. Middleware also can be simply a translator – exposing APIs that are compatible with one operating system for the purpose of enabling software to work on one or more operating systems that otherwise would be incompatible with that software.

In the terminology of the industry, the microprocessor and the computer's other electronic devices, the operating system, and the APIs that are exposed by middleware each are called a "platform" because each supports some form of software. These platforms plus the array of peripheral devices and applications that are used with them are called the "PC system." The combination of a microprocessor and an operating system also is called a platform. For example, the "Wintel platform" is the combination of a Windows operating system and an Intel microprocessor.

Economic Characteristics of PC Systems

The elements of a PC system have four features that must be taken into account to understand the economics of the computer industry and to undertake an antitrust economic analysis of this matter. These features are complementarity, vertical competition, economies of scale, and network effects.

Complementarity

Complementarity refers to the circumstance in which the value of one product is enhanced by the use of others. PC hardware, operating systems and middleware are valuable to users only because applications programs make use of them. Likewise, applications programs would be useless without PCs and operating systems that enable them to function.[4] Complementarity causes the demand for one element of the system to increase if another element has a lower price or higher quality. For example, if word processors become less expensive or more useful, the platform components (operating system, microprocessor) become more valuable. Likewise, if new software enables PC users to place spreadsheet tables into documents and electronic messages, then word processors and e-mail programs become more valuable to at least some end users.

Complementarity among the components of a PC system has been enhanced by rapid technical progress in computer hardware. Increased speed and memory have improved the functionality of PC software, which in turn has increased the demand for applications that take advantage of these enhanced features and for operating systems that permit middleware and applications programs to use these capabilities. Likewise, the increased functionality of software has increased the demand for PC hardware that facilitates the increased functionality of software.

[5] For an elaboration of the presence and significance of strong complementarities among components of PC systems, see Timothy F. Bresnahan, "Network Effects and Microsoft," SIEPR Discussion Paper No. 00-51 (August 2001), Stanford Institute for Economic Policy Research.

One implication of complementarity is that the producer of one component of a PC system can benefit from actions that improve the functionality of another component. That is, a producer of, say, an operating system can increase the value and sales of its product by assisting applications vendors in making their products perform better on the operating system. In particular, to maximize sales and profits of a new operating system, its producer must make the features of its operating system transparent and accessible to producers of applications. The argument also works in reverse. If a vendor creates an exciting new application that it can not implement because of a flaw in the operating system, the vendor and the operating system producer can benefit if the former communicates the problem to the latter.

Another implication of complementarity is that a producer of one product prefers that the market for complementary products be as competitive as possible. More competition in a complementary product means lower prices and greater quality in that market, which in turn increases demand for the other complementary product. Thus, the producer of one complementary product has an incentive to facilitate entry into the markets for complementary products.

Vertical Competition

Competition that emerges among components that are in different layers of the PC System and that otherwise are complements rather than substitutes is called vertical competition.[5] As a technical matter, a platform function can be incorporated into microprocessors, operating systems and middleware. Microprocessors and middleware can expose APIs that can be used by applications as an alternative to APIs that are exposed by operating systems. The significance of this fact is that competition can arise

[5] See Timothy F. Bresnahan and Shane Greenstein, "Technological Competition and the Structure of the Computer Industry," *Journal of Industrial Economics* 47(1) (March 1999), pp. 1-40, and Timothy F. Bresnahan, "The Right Remedy," SIEPR Discussion Paper 00-49 (August 2001), Stanford Institute for Economic Policy Research.

among microprocessors, operating systems and middleware; however, this competition is limited because of the efficiency advantages of placing some features in each component of the system.

The advantages of operating systems over microprocessors as the place to expose APIs for applications are the ease of correcting errors and technical limitations on the size of microprocessors. Operating systems can be very large, containing millions of lines of computer code. The factor limiting the size of an operating system is the size of the permanent storage memory of the PC, which is far larger than the capacity of a microprocessor and supporting memory chips that it uses as temporary storage devices when undertaking operations. Moreover, because operating systems are large and complex, they are virtually certain to contain "bugs" (coding errors). If mistakes are manufactured into a chip, they are difficult and costly to correct or to work around, whereas a coding error in software can be corrected or worked around by inserting a "patch" of a few lines of new code.

With respect to the boundaries between operating systems and middleware, it usually is more efficient to include some platform functions in operating systems because doing so allows more applications programs to access them. Operating system design faces a trade-off between the advantages of comprehensiveness (performing more functions to support a wider array of applications) and the disadvantages of complexity (reducing speed and stability due to the greater number of bugs in a larger, more complex program). In general, placing platform functions in an operating system (rather than middleware or an application) is less likely to be efficient – to have more costs than benefits – if the feature is used by fewer applications and users. If a platform function is included in an application, each application that seeks to perform the same function must include functionally equivalent programming to accomplish this task. Duplication of functionality among applications programs is undesirable because it increases both the cost and the total size of all of the programs stored in memory compared to placing the function in the operating system that supports these applications.

Historically, middleware has offered an alternative to operating systems for particular types of applications. Frequently competition between operating systems and middleware arises when an advance in technology creates new platform capabilities, but the innovator in writing software to implement these capabilities does not produce an operating system. In this case, the innovator creates middleware that exposes APIs to support innovative applications, but nevertheless accesses the APIs of an operating system. Thus, a market environment in which middleware entry is unencumbered facilitates rapid innovation in applications software and encourages greater innovation in PC hardware.

Notwithstanding the theoretically optimal place to add new functionality, vertical competition arises when a producer of one product is dissatisfied with a vertically related complementary product. Operating systems, applications programs, and middleware are complex programs that perform many functions. A vendor that is dissatisfied with a feature of a complementary product sometimes can solve the problem by adding new functionality to its own product that theoretically is more appropriate for the vertically related product. Eventually a sequence of extensions to the functionality of one product can cause it to evolve into an entirely new middleware product that exposes APIs to others who seek the same functionality, perhaps becoming a full-fledged new competitor to a vertically related complementary product. Thus, the availability of the option for vertical competition is a potentially important avenue for technical progress in software as well as for increasing competition in related software markets.

Economies of Scale

All of the major production activities associated with PC systems except PC manufacturing exhibit economies of scale. As Bill Gates said, "the wonderful thing about information technology products is that they're scale economic [sic]. That is, once you've done all that R&D, your ability to get them out to additional users is very, very inexpensive. Software is slightly better in this regard because it's almost zero marginal cost."[6]

Microprocessors, operating systems, middleware and applications have similar cost structures: a

[6] www.microsoft.com/billgates/speeches/2000/10-18digitaldividends.asp, Q and A session.

large fraction of total cost is accounted for by initial design and creation, while the cost of duplicating each copy of the finished product in a new PC system is relatively small. Like recorded music, motion pictures, newspapers, data bases and books, software products are information goods[7] in that the "first copy cost" of creating the initial unit (here, designing, writing, testing and debugging the program) is far more costly than making additional copies. For software products that are distributed over the Internet, the incremental cost of supplying one more copy is extremely small (although not zero, as some loosely state, because the long-run incremental costs of using the Internet and supporting server capacity are positive). If software products are distributed through OEMs, the incremental cost is even smaller (and is zero for the software vendor), for in this case the OEM copies and loads the product onto PCs before they are shipped to retail outlets or end-users.

In 1993, Microsoft estimated that its cost of goods sold – all the distribution costs that are not part of the first-copy costs – accounted for 17 percent of revenue.[8] Software products also have support costs, i.e., software vendors need to provide information and debugging services for customers.[9] These costs account for around ten percent of total costs and do not exhibit economies of scale.[10] Thus, over 70 percent of Microsoft's revenues are either profits or recovery of fixed costs.

[7] See Roger G. Noll, "The Economics of Information: A User's Guide," in *The Knowledge Economy: Annual Review of the Institute for Information Studies*. Aspen Institute, 1993.

[8] Microsoft's Business Strategy, September 1993, Bates Nos. MSPCA01181760-8.

[9] Microsoft does not support software pre-installed by OEMs or other large-volume licenses, which removes a large fraction of incremental unit costs from much of their sales.

[10] For example, in 1990 Microsoft allocated six percent of DOS retail upgrade revenues to support costs (Bates No. MSPCA01109177) and the support costs for the academic version of the Office Pro suite was estimated as $19.51 in 1997, compared to Microsoft's revenues per unit of $140 for this product (Bates No. FLAG000086545).

Economies of scale have important implications for market competition. A software product is financially viable only if its price exceeds the incremental cost of distributing one more unit. If price does not substantially exceed incremental cost, a producer cannot recover its first-copy cost. In highly competitive markets, price is driven to incremental cost; hence, software firms must have some market power to recover their total costs.

Another feature of information products is that competition is more likely to arise through product innovation and differentiation – special features for subsets of users – than simple duplication of products that already are in the market. One reason is that pure duplication is likely to be prohibited by intellectual property law (computer programs can be protected by both copyrights and patents). Another reason is that if several firms sell identical products, market competition will drive price to incremental cost, preventing any from recovering its first-copy costs. Hence, the incentive to enter will disappear after very few firms are in the market.

If products are differentiated, each can have loyal customers that prefer its unique features, somewhat ameliorating price competition among products. In this case, a greater number and variety of products will survive in the market, although the price of each will remain above its incremental cost. This type of market is especially likely to promote "dynamic competition" in which producers race to develop new features and other quality improvements. Dynamic competition benefits consumers by creating both price competition and quality competition.

Economies of scale also enhance the incentive of a product vendor to assist the vendor of a complementary product in increasing its functionality. Economies of scale cause the incremental profitability of an increase in sales to be higher than the average profitability of all sales. Hence, if an improvement in one product increases the sales of a complement, the profitability of the complement will increase without its producer having to make expenditures to improve it. Thus, economies of scale enhance the incentives for cooperation between vendors of complementary products.

Network effects[11]

A network effect arises when the value of a product to one customer is affected by the number of others who use it. This phenomenon initially was observed in the telephone industry, in which the value of a telephone to one person depends on how many other people can be called. Software products also can have network effects because users frequently want to share files and data bases. File sharing is inhibited if file formats are not fully compatible among different products that perform the same basic function. Thus, a user may regard a particular software product as inferior to competing products, but nonetheless acquire it to facilitate sharing files with others who use the same software product. For example, a user may prefer WordPerfect for word processing, but either supplement or replace it with Microsoft Word because some business and personal correspondents work only in Word. In this circumstance, ease of file sharing by using the less preferred product generates more value than the increased functionality of the preferred product, causing the user to buy the product with greater compatibility instead of the product with greater functionality.

Another network effect arises from the complementarities between an operating system and the applications it supports. Due to complementarities, the value of an operating system is derived in part from the value of the applications software that it supports. Hence, an operating system becomes more valuable as more software applications and middleware programs are created to interface with it. This phenomenon is an example of a network effect, and creates the applications barrier to entry. PC owners are likely to prefer an operating system which already supports many applications to a competing operating system that supports few applications. Users also are likely to want to continue to use

[11] See Michael Katz and Carl Shapiro, "Systems Competition and Network Effects," *Journal of Economic Perspectives* Vol. 8, No. 2 (1994), pp. 93-115; Timothy F. Bresnahan, "The Economics of the Microsoft Case," SIEPR Discussion Paper No. 00-50 (August 2001), Stanford Institute for Economic Policy Research; and Bresnahan, "Network Effects and Microsoft," *supra.*

applications that they already have learned how to use, or at least to be able to access files that were created by these applications, when buying a new PC/operating system platform.

An application that was created to run on one operating system can be revised to run on an alternative operating system (called "porting"). Once an application has been ported, a second operating system can become a viable choice for users who want to continue using that application. Applications and middleware that have been ported to several operating systems are called a cross-platform software, and if sufficiently many programs are cross-platform, the applications barrier to entry is destroyed.

Porting is not common because it is costly. Porting requires rewriting the code that serves to connect the application to the operating system. As a practical matter, although Microsoft has ported Word and Excel to the Apple PC system, most applications software vendors, whether companies that also sell operating systems or independent software vendors (ISVs), do not find it useful to port most applications. For example, in the early 1990s, the dominant DOS operating system (in which MS DOS was the biggest seller and DR DOS had become a serious alternative that was compatible with MS DOS) supported 21 word processors and 11 spreadsheets, while Apple, OS/2, and Unix supported six or fewer word processors and four or fewer spreadsheets.[12] Because most software products are not ported, incompatible operating systems support many different middleware and applications program that typically also are incompatible.

Porting applications becomes increasingly unattractive as the market share of the dominant operating system grows. The cost of porting an application to a second operating system is an example of a fixed "first-copy" cost because it does not depend on the number of users of the operating system to which a program is ported. As the number of users of an operating system shrinks, the potential sales of an application to users of that operating system also falls, causing the expected profitability of porting to decline. For this reason, as the market share of an operating system grows, it can reach a "tipping point"

[12] Bates Nos. MSPCA01108576-659.

at which applications vendors stop writing software for any other operating system.

The alternative to writing to the APIs of an operating system is to write to the APIs of middleware. This alternative can be an attractive option if two conditions are met. First, the middleware program is present on all or nearly all of the PCs that use the operating system to which the application otherwise would be written. Thus, writing to middleware APIs does not reduce the potential market to users of the dominant operating system. Second, the middleware program is cross-platform, thereby giving the applications vendor access to users who otherwise could not use the product. If these conditions are satisfied, a software vendor can gain access to all or nearly all customers on the original platform plus new customers on alternative platforms by writing to the middleware APIs. If many applications vendors write to the APIs of cross-platform middleware, the applications barrier to entry of the dominant operating system is eliminated, facilitating more robust competition in operating systems.

Summary of PC Economics

The four characteristics of PC systems have important implications for the nature of competition in markets for components of a PC system and the incentives of the firms in the industry. One important implication is that incremental buyers are extremely valuable because the profit from an incremental sale is far greater than the average profit per sale. Both economies of scale and network effects (one more customer increases the demand for the product) have this effect. Another implication is that a firm operating in only one market has an incentive to cooperate with firms in other markets, in part because other products are complements and in part because a failure to cooperate can lead to vertical competition.

MARKET DEFINITION

The principles for defining relevant markets are set forth in the *Horizontal Merger Guidelines* of the Antitrust Division of the U. S. Department of Justice and the Federal Trade Commission. A "relevant market" for purposes of antitrust analysis is a product or group of products that profitably can be monopolized. In defining a relevant market, one begins with the specific product and geographic location that is the focus of the analysis, then adds products and expands the geographic area until one has the smallest possible group of products that, if all of the products were offered by a single seller, would allow a significant, long-term increase in price above the competitive level.

The "smallest market" principle sometimes leads to market definitions that are narrower than is implied by the ordinary use of words like market and competitors. If a product is sold by a single seller with substantial market power – that is, the product commands a price above the competitive level – then that product by itself constitutes a relevant market for purposes of antitrust economics. For a product to be in the relevant market it must compete with the reference product and, perhaps with other products, must constrain the seller of the reference product to charge approximately competitive prices.

The concept that guides market definition in antitrust economics is the principle of substitution. Demand substitution refers to the notion that a user of a product may switch to another if the supplier of the first attempts to increase profits by raising price or reducing costs by shading quality. Whereas in a weak sense all products are substitutes and therefore compete with each other as long as expenditures by buyers are limited by a budget constraint, the concept of substitution that is used in defining relevant markets is the more demanding one of functional substitution. If one product is a very close substitute for another, an attempt by the seller of one product to raise price will cause its customers to switch to the other, thereby causing the price increase to be unprofitable. A relevant market, then, consists of a reference product (the focus of the antitrust concern) and other products that are close enough substitutes to restrain the price behavior of the reference product.

Substitution also can take place on the supply side. One example pertains to location. A seller of a product in one location, upon observing higher prices for the same or a similar product elsewhere, may be able to relocate sales to the latter location. In similar fashion, if the production process of one product is sufficiently flexible that production can be switched to another product at very low cost, an increase in the price of the latter will induce suppliers to switch production away from the first and to the second. The possibility of supply substitution is used to identify "potential competition" that might arise if a market were monopolized; however, if the switch in production is virtually costless and can be accomplished quickly, both products should be regarded as in the same market even though they are functionally different and so not close substitutes on the demand side of the market.

Reference Products

This litigation deals with the alleged use by Microsoft of its market power in x86 operating systems and GUIs to undermine various applications software products that were produced by Novell. Since 1995, each new version of Microsoft's operating system has been one of several products that are bundled together in a product that is called Windows. The reference products are two components of this bundle, the Microsoft operating systems (originally called MS DOS) and Microsoft's GUI (originally called Windows). Internet Explorer was first released in 1995 as part of an optional package of additional software for Windows 95, but an upgrade to Windows 95 that was released in August 1996 contained the Internet Explorer browser, as have all subsequent versions of Windows. More complete descriptions of these products are contained in the technical expert report, so only a brief summary is provided here. After describing the basic functionality of Microsoft's products, this report identifies some other products that have competed with Microsoft in each product category and explains why each is a distinct relevant market for purposes of antitrust economics.

Operating Systems

Microsoft's MS DOS was an operating system for x86 PCs that was sold as a separate product through MS DOS 6.22, which was released in 1994. Between 1987 and 1995, many new features of MS DOS (called DOS+) were not included in MS DOS, but instead were attached to the Windows GUI.

Beginning in 1995, the entire Microsoft operating system, including an update of MS DOS that Microsoft incorrectly said would be released as a separate product, was bundled with the GUI as Windows 95.[13] After the introduction of Windows 95, new versions of Windows were released to allow applications to use the capabilities of new microprocessors, and new versions of applications were developed to take advantage of these enhanced platform features.

Operating systems are defined in Microsoft's *Computer User's Dictionary* as the software that controls PC hardware for other programs. In practice operating systems long have been bundled with some applications, such as a calculator, a screensaver, and a few simple games such as solitaire or battleship. Nevertheless, these products are not essential features of an operating system. According to Bill Gates, "if it includes low level code, which is the lowest level code that runs the system, ... then I would say ... that's an operating system product."[14] For reasons given subsequently in this section, bundling an operating system with other applications does not normally cause the latter to become part of the relevant market for operating systems. Microsoft sources support the view that a necessary feature of an operating system product is code that enables another program to mobilize the hardware of a PC to carry out its functions.

Graphical User Interfaces (GUI)

Microsoft Windows originally was a graphical user interface (GUI). All operating systems have a mechanism by which a user communicates with the operating system about which program to run, which files to access, and what operations the program should perform. This interface is called a shell. One method of communicating these commands is to type a string of alphanumeric characters that

[13] Beginning with Windows 2.0, new platform features were bundled with the GUI, but MS DOS was still sold as a separate product.

[14] Deposition of Bill Gates, p. 41.

correspond to definitions of commands, programs and file names that are recognized by the operating system or an application. GUI software creates a graphical format by which users can select programs to run and, within programs, issue commands and perform functions by using a cursor-control device to point and click on visual icons and names on a screen, rather than memorize alphanumeric commands. One advantage of a GUI is that it eliminates the need to learn the character commands to undertake these tasks, and so makes PC software substantially more accessible to users. Another advantage of a GUI is that it provides a common display format and command nomenclature for all applications, thereby reducing the difficulty of learning and remember how to operate each application. For businesses, a GUI increases employee productivity and lowers training costs: "many managers believe that a multitasking, graphical PC interface is the best way to get more and better work out of their employees."[15]

Microsoft Windows originally was a graphical user interface (GUI) for use with DOS-compatible operating systems. Windows GUI is an application in that it enables users to access programs by using a mouse or other pointing device to "point and click" a cursor on icons on the initial screen that a user sees after turning on a PC. Windows GUI also is middleware in that it exposes APIs that enable applications software to create displays in the Windows format. Originally, the Windows GUI was cross-platform middleware as it ran on MS DOS, DR DOS and PC DOS.

With the Windows 2 family of products, Microsoft began bundling platform functions that normally would have been placed in an operating system with the Windows GUI, making Windows an incomplete operating system as well as GUI middleware. The Windows 2.x family was quite heterogeneous, since it included distinct versions for several different x86-compatible microprocessors, including Windows 286 (which included a more advanced task switching feature) and Windows 386 (which included a multitasking feature that was enabled by the "virtual DOS machine" element of Intel's 386 microprocessor). Windows 3.0 contained these functions plus others, including methods to eliminate

[15] Robert L. Scheier, "PC Managers Balk at Windows Pitch," July 9, 1990, *PC Week*, v. 7 no. 27, p. 1.

the 640k memory limitation for applications. Windows 3.1 included a broader range of features that allowed it to take better advantage of 32 bit microprocessors. Finally, the Windows GUI and its associated operating system features were bundled with the MS DOS system in Windows 95. Since then, Microsoft has not released an unbundled version of either MS DOS or the Windows GUI.

Browsers

Microsoft's Internet Explorer and Netscape Navigator are browsers. Netscape was the dominant supplier of browsers in 1995. Internet Explorer initially was released in a separate package of software in 1995, but was bundled in a 1996 update for Windows 95 and all subsequent versions of Windows. It also has run on the Apple Macintosh since April 1996.[16] Browsers enable a user to download and display a file in standardized formats from a computer of the user's choice over the Internet, as well as files in the same formats on the user's hard drive or an intranet server. Microsoft offers the following explanation of the functions of a browser: "Software that lets a user view HTML documents and access files and software related to these documents. Originally developed to allow users to view or browse documents on the World Wide Web, Web browsers can blur the distinction between local and remote sources for the user by providing access to documents on a network, an intranet, or the local hard drive. Web browser software is built on the concept of hyperlinks, which allow users to point and click with a mouse in order to jump from document to document in whatever order they desire. Most web browsers are also capable of downloading and transferring files, providing access to news groups, displaying graphics embedded in the document, and executing small programs, such as Java applets or ActiveX controls included by programmers in the documents."[17]

[16] Timothy F. Bresnahan and Pai-Ling Yin, "Economic and Technical Drivers of Technology Choice: Browsers," *Annales d'Economie et de Statistique* (forthcoming).

[17] *Microsoft Computer Dictionary*, 5th Edition, Microsoft Press, p. 562.

Browsers must be supported not only by the operating system on a PC, but also by the operating systems of the local server that connects them to the Internet and external servers that support the sites that they access. Since Microsoft bundled Internet Explorer with Windows, browsers have been given away for free, and rely on revenues from web sites to recover their costs.

Word Processors

Word (Microsoft), WordPerfect (then Novell) and WordPro, formerly AmiPro (IBM) are word processors. Word processors enable users to create documents in many styles and formats. According to *Webopedia*, a word processor is a program for "[u]sing a computer to create, edit, and print documents. Of all computer applications, word processing is most common. A word processor enables you to create a document, store it electronically on a disk, display it on a screen, modify it by entering commands and characters from the keyboard, and print it on a printer. The great advantage of word processing over using a typewriter is that you can make changes without retyping the entire document."[18] Microsoft defines a word processor as: "An application program for creating and manipulating text-based documents... Depending on the program and the equipment in use, word processors can display documents either in text mode (using highlighting, underlining, or color to represent italics, boldfacing, and other such formatting) or in graphics mode (in which formatting and, sometimes, a variety of fonts appear on the screen as they will on the printed page). All word processors offer at least limited facilities for document formatting, such as font changes, page layout, paragraph indentation, and the like. Some word processors can also check spelling, find synonyms, incorporate graphics created with another program, align mathematical formulas, create and print form letters, perform calculations, display documents in multiple on-screen windows, and enable users to record macros that simplify difficult or

[18] www.webopedia.com/TERM/w/word_processing.html.

repetitive operations."[19]

Whereas operating systems sometimes contain text editors with limited word processing capabilities, these programs are extremely simple and can produce only very rudimentary documents. Among the functions in a word processor that are not part of the text editors that are included with operating systems are the following capabilities:

* Highlighting text with shadings and colors;

* Creating tables of contents and indexes;

* Tracking changes in a document from draft to draft;

* Varying page size and margins within a document;

* Inserting footnotes, endnotes and page numbers;

* Placing headings that can vary throughout the document;

* Creating "macros" that, for commonly used sequences of characters, replace a series of key strokes with a single key stroke; and

* Checking spelling, punctuation, grammar and syntax.

As computers have become more powerful, word processors have incorporated a growing number of special features for incorporating graphical displays, art work, and mathematical equations; however, a word processor is not generally used to perform mathematical calculations or to create complex computer art. Users who want to perform complex calculations or to create computer art buy separate software in these categories.

Spreadsheets

Excel (Microsoft), QuattroPro (then Novell) and Lotus 1-2-3 (IBM) are spreadsheet programs. A spreadsheet is defined in Microsoft's 1989 mission statement for Excel: "tools for analysis and

[19] *Microsoft Computer User's Dictionary,* Fifth Edition, Microsoft Press, pp. 573-4.

presentation of numerical data in graphical computing environments" that ... "have models using numbers and formulas, make graphs based on those numbers, make decisions based on those numbers and present those numbers and graphs to others..."[20] Among the features included in a spreadsheet are the capability to perform the following operations:

* Having multiple worksheets within a single workbook;

* Creating macros;

* Merging or joining cells;

* Highlighting text;

* Transferring data among worksheets; and

* Tracking changes from draft to draft.

According to Microsoft, a spreadsheet is an "application commonly used for budgets, forecasting, and other finance-related tasks that organizes data values using cells, where the relationships between the cells are defined by formulas[...] Spreadsheet programs usually provide graphic capabilities for output and a variety of formatting options for text, numeric values, and graph features."[21]

These capabilities go far beyond the simple data capabilities that in some cases are built into an operating system.[22] Initially spreadsheets were useful primarily for keeping accounting records, making tables from data sets, and performing simple arithmetical operations across columns of data. As

[20] Analysis Business Unit, Excel 3-Year Plan, December 12, 1989, Bates Nos. X000000583042-61 and Excel Strategy and Ideas, July 1, 1990, Bates No. X000000209905.

[21] *Microsoft Computer User's Dictionary*, *op. cit.*, p. 494.

[22] Bill Gates: "The free spreadsheet should also ship with Chicago [Windows 95] – both on the CD and to OEMs. The free spreadsheet ... would be missing things like multiple sheet support and tons of other features of excel." September 17, 1994, Bates Nos. FLAG000041559-60. In fact, Microsoft later decided not to include a free spreadsheet with Windows 95.

computers have become more powerful, spreadsheets have incorporated capabilities for statistical analysis, such as correlations and ordinary least-squares regressions. By 2000, Excel could perform over 300 data manipulation functions. Nevertheless, despite these improvements, users who need to undertake more complex mathematical manipulations or statistical analyses find that a spreadsheet program is inadequate. These users purchase other software products, although typically these programs can read and manipulate data files from spreadsheets.

Presentation

Presentation software, sometimes called presentation graphics, contains graphic design tools that enable a user to organize numerical and textual information as a visual aid in a presentation, using slides, overlays or direct display from a computer. These programs contain a text editor, and the capability of incorporating tables, charts, graphs, photos and clip art into the displays.[23] A presentation program contains the following:

* A set of backgrounds and colors for displays;

* Variable fonts to match the display to the size of the room;

* Bullets and boxes for highlighting;

* A group of pictures and designs that can be incorporated with the text;

* Capability to convert data to pie charts, graphs and bars and to use color for emphasis;

* Commands for projecting displays on a screen;

* Slide transitions effects, such as fade and dissolve; and

* Capability to print multiple slides per page.[24]

[23] Wendy Russell, "What Is Presentation Software?" *About.com*, accessed April 30, 2009, at presentationsoft.about.com/od/pq/qt/070911pressoftw.htm.

[24] George Socka, "Presentation Software," *CMA Magazine* Vol. 68, No. 5 (June 1994), pp. 8-9.

The key feature of presentation software is that a library of templates simplifies the task of creating visual displays (such as a slide show) to just filling in the blanks.[25]

Microsoft's presentation application is PowerPoint, and in PerfectOffice and later WordPerfect Office, the presentation application is called Presentations. The Lotus product is called Freelance.

Relational Database

A relational database program, sometimes called a database management system, stores data in the form of related tables that simultaneously be referenced and manipulated. A database program allows the user to do the following:[26]

* design the structure of the database;

* create data entry forms for placing information in the database;

* validate data entries;

* sort and manipulate data entries;

* query the database; and

* collect data entries into reports.

Microsoft's relational database is called Access. The relational database program in PerfectOffice and, subsequently, Corel's WordPerfect Office was Paradox, which originally was created by Borland. The Lotus product is called Approach.

Groupware

Groupware, or collaboration software, refers to applications that enable end users to communicate

[25] Alan J. Fridlund and Tricia Harney, "Presentation Graphics," *InfoWorld* Vol. 16, No. 36 (September 5, 1994), pp. 65-77.

[26] Rose Vines, "Databases from Scratch," at www.geekgirls.com/databases_from_scratch_1.htm.

privately. "Groupware refers to programs that help people work together collectively while located remotely from each other... Groupware services can include the sharing of calendars, collective writing, e-mail handling, shared database access, electronic meetings with each person able to see and display information to others, and other activities..."[27] Among these other activities are hierarchical routing of documents for approval, and circulation and retrieval of forms. As with browsers, groupware must be supported by a PC operating system, network server software that manages and stores communications, and middleware that interfaces between the server, the application and the operating system.

Microsoft has changed the names of its groupware applications in a way that is confusing. Prior to the release of Windows 95, the name of Microsoft's principal groupware application was Microsoft Mail, which ran on Windows 3.1 and Windows for Workgroups. Microsoft also developed middleware to support groupware applications that was called MAPI, after messaging applications program interfaces. MAPI supported some versions of MS Mail. According to Microsoft's public statements, MAPI was designed to be an open standard that would support all groupware applications and all server-based messaging systems.

Microsoft's groupware application for Windows 95 was called Microsoft Exchange (sometimes Exchange Client). Initially, the installation of Microsoft exchange was an option a user could select when installing Windows 95.[28] Microsoft combined MAPI and Exchange so that an end-user could not install MAPI without installing Exchange. Microsoft later used the name Exchange for the server software that supported its groupware application, and the application was renamed Microsoft Messaging beginning with the Windows 95 OEM Service Release 1.[29] A more powerful groupware application, Outlook,

[27] SearchDomino at searchdomino.techtarget.com/sDefinition/0,,sid4_gci212217,00.html, accessed April 20, 2009. Domino is the middleware in Lotus Notes groupware.

[28] See technet.microsoft.com/en-us/library/cc751105.aspx#XSLTsection124121120120.

[29] See www.indiana.edu/~uits/telecom/messaging/whatexchange.html.

which also was bundled with MAPI, was included in Microsoft Office beginning in 1997. In Windows 98, the name of the simpler application that was bundled with the operating system was changed to Outlook Express, which continued to be bundled with MAPI.

GroupWise, Novell's groupware application product, originally was called WordPerfect Office, not to be confused with the suite of productivity applications that Corel now sells under that name or with PerfectOffice, the name of the suite of productivity applications that was offered by Novell. Originally Novell sold communal document management and form management software separately, but these were placed in GroupWise in 1996.

IBM's groupware product is Lotus Notes.

Suites and Other Bundles

Different software products frequently are sold in bundles. One ubiquitous bundle is office suites, which combine productivity applications such as word processor, spreadsheet, presentation, database, and groupware programs. Microsoft Office bundles Word, Excel, PowerPoint (presentation), Outlook, Access (database) and other applications. Microsoft Works contains a word processor and spreadsheet that are simpler than Word and Excel. The corresponding bundle from Novell was PerfectOffice, introduced in December 1994. PerfectOffice was derived from Borland Office, which contained productivity applications from both Borland and WordPerfect Corporation. PerfectOffice was the first productivity suite that enabled network applications as it included GroupWise and Envoy (collaborative publishing) as well as WordPerfect, QuattroPro, Presentations, and, in the "Professional" version, Paradox (database). The corresponding Lotus bundle is SmartSuite, which does not contain the messaging product, Lotus Notes, which is sold separately.

Since 1995 Windows has been a bundle of the Windows GUI, MS DOS, a simple groupware product, several middleware products (including Internet Explorer and Windows Media Player), and some applications, including a few games and several screen-savers.

The primary software products at issue in the litigation – MS DOS, Windows GUI, Word, WordPerfect, Microsoft Mail/Exchange Client/Outlook, GroupWise and Internet Explorer – initially were sold separately. Office suites were introduced in 1989, and the bundle containing the Windows GUI and Microsoft's operating system, while partially implemented beginning in 1987, was not complete until 1995. Today bundles are dominant in all relevant markets. Netscape, including its open source successor, Mozilla Firefox, is an exception, as it always has been acquired as a separate product. Internet Explorer was released initially as part of a bundle of add-ons to Windows 95, and has been bundled with Windows since the version of Windows 95 that was released in 1996. Excel and Word still can be acquired separately, as can some other word processors and spreadsheets, although almost all sales of both types of products by both Microsoft and its most important applications competitors are in suites. Several Microsoft applications, including PowerPoint, can only be acquired as part of the Office bundle. Most end-users now acquire the software that they use most intensively with only two purchases: a bundle that includes an operating system and GUI, and another as an office suite.

Alternatives to Microsoft Products

Alternatives to Microsoft products have existed and continue to exist in each product category. Table 1 contains a list of products for which I have been able to document measurable sales in each product category from 1988 to 2008. (The list probably is incomplete, but it is not likely to exclude any product that accounts for a significant share of sales in any relevant market.) Although MS DOS has been the most important operating system for x86-compatible PCs since before the period at issue in this litigation, others have attempted to compete. The most important examples in the late 1980s and early 1990s were DR DOS (created by Digital Research, and subsequently acquired first by Novell and then by Caldera), Unix (developed at AT&T Bell Labs and acquired by Novell in 1993, which sold the administration and development rights to Santa Cruz Operation in 1995), PC DOS (IBM), and OS/2 (jointly developed by IBM and Microsoft, but eventually owned only by IBM). Only Unix remains in the market[30] in the form of Solaris, Sun Microsystem's Unix-based operating system, which can be purchased as a feature-rich proprietary system (Solaris 10) or acquired for free as a simpler open-source system (OpenSolaris).[31] Linux is another x86 open-source operating system that has numerous free and commercial distributors.[32] The first commercial versions of Linux were introduced in 1994 by SuSE (subsequently acquired by Novell) and Red Hat. Other Linux entrants are Debian, Mandrake (now Mandriva) and Xandros. In addition, Intel and Sun developed an operating system in which Intel microprocessors exposed some APIs while the rest were exposed in Sun software, but this product was never brought to the market.

Since the introduction of Windows 3.0, the only significant direct competitor to the Windows GUI on the x86/MS DOS platform has been Presentation Manager (PM), which was sold bundled with OS/2. Until the break between IBM and Microsoft in 1991, PM was a joint venture between these two companies; however, IBM attempted to make OS/2 and PM a competitive alternative to MS DOS and Windows after its partnership with Microsoft dissolved. Hence, PM and OS/2 are considered as products that were marketed independently of Microsoft. Other GUIs were available early in the life of the Windows GUI. The most important was GEM. In 1988, the revenue share of Windows in the GUI market was 68 percent, while GEM had 24 percent, Presentation Manager (bundled with OS/2) had six percent, and all others had two percent. Since the overwhelming success of Windows 3.0, no other GUI has competed successfully with Microsoft on the x86 platform. Several GUIs are available for Linux-based operating systems and Sun's Solaris, but each of these products has a tiny share of a x86-

[30] IBM's PC DOS apparently still is used in some cash registers, but otherwise the last reported sale was in 2005.

[31] See http://www.sun.com/software/solaris/.

[32] For a current list see http://www.linux.org/dist/list.html.

compatible PC systems because these operating systems have low OS market shares.

One component of the Windows GUI is the Windows Explorer shell – an environment from which a user launches programs, opens files, and performs other tasks. Other firms have created shells, sometimes in connection with other applications products, that offer an alternative to Windows Explorer. The PerfectOffice shell originally was part of WordPerfect Office (the predecessor to GroupWise), which was then incorporated into PerfectOffice after the Novell acquisition. For Lotus and then IBM, the shell is part of Lotus Notes, the Lotus groupware product. These shells are examples of vertical competition between Microsoft and the other two firms in which middleware was included with one or more applications to compete with one element of another middleware product, the Windows GUI.

Excel, Internet Explorer and Word have similar histories. As measured by new installations, none was dominant in its category until 1993. Before 1993, the leader in spreadsheets was Lotus 1-2-3 and the leader in word processors was WordPerfect. Internet Explorer was introduced in 1995, when the leading browser was Netscape Navigator. All of these products (Netscape Navigator as Mozilla Firefox) remain in the market as alternatives to Microsoft, although each has lost its leadership position. Excel and Word became the leading products in their category in 1993, and Internet Explorer in 1997. While no other browsers have been important, in spreadsheets and word processors other products enjoyed some success until the early 1990s, notably AmiPro (Lotus), Multimate (Ashton-Tate) and WordStar in word processing, and QuattroPro (Borland) in spreadsheets. Today only four vendors of word processors that run on MS DOS/Windows have significant sales: MS Word/Works, Corel WordPerfect, IBM WordPro (formerly AmiPro, in Lotus SmartSuite), and Sun Writer (in StarOffice). Likewise, only four spreadsheets that run on MS DOS/Windows have significant sales: Excel, Lotus 1-2-3 (SmartSuite), QuattroPro (WordPerfect Office) and Calc (StarOffice).

In groupware, the leading products in the mid 1990s were Microsoft Mail/Exchange Client/ Messaging/Outlook, Lotus Notes, and GroupWise. Of these, only Microsoft's product is sold only in bundles, with a simpler product part of Windows and a more feature-filled product part of Office.

GroupWise was bundled with PerfectOffice in 1994, and remained part of WordPerfect Office after Corel acquired Novell's office applications programs except for GroupWise. Corel no longer includes GroupWise in its office suite. GroupWise also has been sold as a stand-alone product, even when it also was bundled. Lotus Notes and Lotus SmartSuite are still sold as separate products.

The Relevant Product Markets

An important issue regarding market definition is whether for purposes of antitrust analysis each of these categories of products is sold in a separate relevant market, or whether some should be combined into broader categories. In prior litigation Microsoft has claimed that the operating system includes GUI and middleware (including browsers) and that the relevant product market is a "PC system" that includes the hardware and the operating system for all PCs (not just the x86 family) and some other types of computers.[33] Moreover, the applications products at issue in this case might be regarded as in the same product market because nearly all sales are in suites. More generally, Microsoft has argued that system integration creates efficiency advantages from combining programs with different functions, which tends to cause different software products to merge into a single product for purposes of market definition.

The Operating Systems Market

Some market definition issues were resolved in *U. S. vs. Microsoft*. Both the District Court and the Court of Appeals for the District of Columbia ruled that operating systems for x86-compatible PCs constitute a relevant market for antitrust purposes; however, the Court of Appeals ruled that insufficient evidence was presented to determine the relevant market in which browsers were sold. Neither side

[33] "Expert Report of Kenneth Elzinga," *Caldera vs. Microsoft*. Microsoft's economic expert in *U. S. vs. Microsoft*, Richard Schmalensee, did not define the relevant markets, but in cross-examination testified that the government should have defined the relevant product as PC platforms.

addressed whether the operating system was separate from the GUI or how to define markets for applications (these issues were not relevant to the government's complaint or Microsoft's defense).

Because the government case provides useful guidance for defining other markets in software, a review of the reasons for the finding that x86-compatible operating systems constitute a relevant market is useful here. In defining the relevant market for Microsoft's operating system, the government emphasized functionality, i.e., the test for whether products are close enough substitutes to constitute real competitors is determined by whether they do or can perform the same user functions. The government argued that operating systems are functionally distinct from other PC system components because nearly all of their functions differ from functions found in either PC hardware or other software.

Recall the preceding discussion about the definition of operating systems. Basically, an operating system is the interface between computer hardware and other software. Theoretically, microprocessors and other microelectronics devices can include operating system functions. Likewise, middleware includes operating system functions. For example, a translator – middleware that converts calls to its APIs to calls to the APIs of an operating system – can expose an array of APIs that is comparable in functionality to an operating system; however, even a translator is useful only because its can run on one or more operating systems. Thus, the practical business reality of the industry is that operating systems contain substantial functionality that is not duplicated by either microprocessors or other software.

Microsoft and its economic experts argued in other litigation that these functional distinctions are economically irrelevant, in part because customers do not have independent demands for hardware and operating systems. Starting with the observation that all PCs must have an operating system, and noting that each customer has a maximum willingness to pay for a PC plus an operating system of given functional capabilities, Microsoft argued that the price of each component is limited by the price of the other. According to this argument, enough competition occurs horizontally across PC systems and vertically between microprocessors and operating systems to make all part of the same relevant market.

The issue that Microsoft has raised by proposing this market definition goes well beyond the existence of some overlap in the functional capabilities of microprocessors, operating systems, middleware and applications that was raised in the section about the economics of PC systems. Microsoft's claim goes beyond simply asserting the existence of some vertical competition, i.e., that producers of each of these products may encroach on features that formerly were offered by a product in another category. Instead, the claim is that competition between types of platforms, between microprocessors and operating systems, and between operating systems and functional substitutes incorporated into middleware and applications, is strong enough to be equivalent to competition within the same product category (e.g., between two x86 operating systems).

Microsoft's argument is theoretically possible in that it follows from a specific economic theory that is worth spelling out in some detail to determine the extent to which it applies to PCs. Theoretically, three conditions can cause this argument to be true. First, the platform market (combinations of microprocessors and operating systems) is competitive. Second, microprocessors can incorporate virtually all functions of an operating system without a substantial cost penalty or loss of quality. Third, consumers do not suffer from "lock-in" effects – that is, they face low switching costs among platforms and the applications a platform supports in response to small changes in relative prices, thereby giving platform vendors an incentive to try to attract customers of other platforms by offering a better deal.

Competition in Platforms

The first condition for a PC system to be a relevant product market requires that the marketplace must contain several competing platforms, or combinations of a microprocessor and an operating system. While the market for PCs is generally regarded as highly competitive, the relevant products here are the internal electronics of the PC and the operating system. Recall the discussion about economies of scale in information products. For both microprocessors and operating systems, a large fraction of the cost of a product is its design – the first-copy cost. In addition, most competition in both components is dynamic and is related to product differentiation. In principle these products cannot be so competitive that firms have no market power or else firms could not recover their fixed costs. Thus, the cost structure of platform components is inconsistent with the theoretical requirements for robust platform competition.

Notwithstanding the theoretical argument against the possibility of extensive competition in both products, the practical reality is that there are few PC platforms. The reference operating system in this litigation is the sequence of Microsoft operating systems beginning with MS DOS and progressing through the versions of Windows that have bundled platform functions with the Windows GUI. Since the late 1980s, the operating system product for the x86-compatible PC has been dominated by Microsoft. The reference microprocessors in this case are members of Intel's x86 family and its successors. Several companies have produced microprocessors that are compatible with x86 microprocessors; however, Intel is the clear market leader, and among the rest only Advanced Micro Devices (AMD) has competed with Intel during the period at issue in this litigation in making major innovations in x86-compatible microprocessors.

The only alternative to x86-compatible PC microprocessors that enjoys significant sales has been the Motorola and IBM Power PC microprocessor that was sold with the Apple MacOS operating system but which Apple abandoned in 2007. In *U. S. v. Microsoft,* Microsoft argued that the Apple Macintosh-Motorola platform was in the same relevant market as the Wintel (Windows/x86) platform. Considering both operating systems and microprocessors and including Apple/Power PC as a competitor, the maximum number of PC platforms with more than a minuscule presence in the market has never been more than four, and sometimes has been three. One of these, the Intel-Microsoft pair, was dominant in the 1980s and remains so to the present. Thus, even if the basis for defining relevant market were all PC platforms, this product market has not been structurally competitive.

In *U. S. v. Microsoft*, Microsoft also unsuccessfully argued that several other platforms also competed with the Wintel platform, including network computers, workstations/servers, and hand-held computers like the Palm Pilot, along with their corresponding operating systems. Operating systems for other types of computers can not run on x86 PCs, so these products could be substitutes only if the combined computer and operating system platform were a substitute.[34]

These other computer platforms do not provide a similar combination of functionality and price, and so are not close substitutes. Small, portable computers ("handheld" and "pocket PC"), the most advanced versions of mobile telephones ("smart phones"), game computers, and other so-called "information appliances" are not substitutes because they lack operating systems that can support advanced versions of the most important PC applications, such as word processors, spread sheets, e-mail, and presentation programs.

Handheld computers and smart phones provide maximal portability, wireless computer-to-computer connections, and Internet connection to support e-mail and web browsing; however, these devices provide significantly less functionality and cost less than PCs. The fundamental limitation to such devices is the source of their convenience: very small size, which translates into a very small keyboard and display. The functional differences translated to a much different price range. CNET News, a web site that focuses on information technology, provides comparative reviews of a wide range of handheld computers, pocket PCs, and smart phones. The CNET prices for the most highly recommended products in these three categories were all below $700, compared with $479 to $3999 for the most highly recommended portable PCs.[35] According to CNET, the "core applications" of handheld computers are "address book, calendar, to-do list and memos."[36] The best products have versions of

[34] "Findings of Fact," paragraph 20.

[35] See reviews.cnet.com/4521-6522_7-5021304-1.html?tag=ft, reviews.cnet.com/4521-6522_7-5021304-2.html?tag=subnav, reviews.cnet.com/4521-6522_7-5021304-3.html?tag=subnav, reviews.cnet.com/4521-6522_7-5021304-4.html?tag=subnav, reviews.cnet.com/Desktops/4521-6526_7-5021300-1.html?tag=subnav and reviews.cnet.com/Desktops/4521-6526_7-5021300-4.html?tag=subnav, accessed May 29, 2006.

[36] See reviews.cnet.com/4520-9580_7-5139854-2.html?tag=tnav, accessed May 29, 2006.

standard productivity software, but with more limited features.[37] A 2002 CNET product review advised: "The primary reason for using a handheld computer is to keep track of your schedule, to-do list, and contacts... Don't buy a handheld device as your primary productivity-software device. Limited storage, less-than-full-size keyboards, and less-than-full-featured software will restrict your ability to do much work."[38] The most recent CNET review advised: "For those who get most of their work done on the go, a handheld can stand in for a full-fledged laptop in many situations."[39] The current review then discusses the specific applications on the leading handheld computers, noting that Windows-based handhelds support simpler versions of Word and Excel: "Windows Mobile 6 brings the real deal so you can not only see said files but also edit them, though the editing capabilities are pretty light. It still shocks us, however, that Microsoft offers no convenient way to synchronize Word and Excel documents with their desktop counterparts unless you relocate everything to a special folder."[40] And PowerPoint files can only be viewed, not edited.[41]

The other principal form of information appliance is a game computer. Like personal computers, game machines have evolved rapidly, and now support extremely sophisticated game programs with extensive audio and video capabilities. But game machine manufacturers have continued their strategies of producing specialized, proprietary products. These computers still do not have software that permits them to be used for applications other than games.

The District Court in *U. S. v. Microsoft* concluded that no form of information appliance was a

[37] *Ibid.*

[38] "Why Would I Want a Handheld?" www.zdnet.com/filters/printerfriendly/0,6061,2251592-3.html, July 9, 1999, accessed July 21, 2002.reviews.cnet.com/4520-9580_7-5139854-1.html?tag=text

[39] See reviews.cnet.com/4520-9580_7-5139854-1.html?tag=tnav, accessed April 28, 2009.

[40] See reviews.cnet.com/4520-11309_7-6624304-3.html?tag=nav, accessed April 28, 2009.

[41] See reviews.cnet.com/4520-9580_7-5139854-2.html?tag=tnav, accessed April 28, 2009.

plausible substitute for a personal computer. "No operating system designed for a hand-help computer, a 'smart' wireless television, a television set-top box, or a game console is capable of performing as an adequate operating system for an Intel-compatible PC... [W]ithin the next few years, those consumers who otherwise would use an Intel-compatible PC system solely for storing addresses and schedules, for sending and receiving e-mail, for browsing the web, and for playing video games might be able to choose a complementary set of information appliances over an Intel-compatible PC system without incurring substantial costs... [W]hile some consumers may decide to make do with one or more information appliances in place of an Intel-compatible PC system, the number of these consumers will, for the foreseeable future, remain small... One reason for this is the fact that no single type of information appliance, nor even all types in the aggregate, provides all of the features that most consumers have come to rely on in their PC systems and in the applications that run on them."[42] Indeed, the "next few years" have arrived, and handheld computers and smart phones have evolved as predicted. But PCs also have evolved, with ever-increasing computational power and memory, so that information appliances still are not substitutes for a full-fledged desktop or laptop PC.

Servers and workstations are another type of small computer that can have some functional overlap with a PC. The terms server and workstation apply to a variety of very different computers. For example, a "web server" is simply a computer that contains programming that enables it to function as a host site on the World Wide Web so that it can be accessed by other computers.[43] A PC is able to function as a web server with the appropriate software. Network servers are special purpose computers

[42] "Findings of Fact," paragraphs 22-3.

[43] "Web server (HTTP server) n. 1. Server software that uses HTTP to serve up HTML documents and any associated files and scripts when requested by a client, such as a Web browser. The connection between client and server is usually broken after the requested document or file has been served. HTTP servers are used on Web and Intranet sites." Microsoft Computer Dictionary, 5th Edition, Microsoft Press.

that are connected to many other PCs and that contain data and programs that other computers can use.[44] These types of servers can be much more powerful than a PC, but they are not intended to be used by one person as a desktop PC. Likewise, workstations come in many varieties, from relatively inexpensive computers that are used for specialized purposes, such as computer-assisted design, to desktop computers that are substantially more powerful – faster microprocessors, better visual displays – than a PC. Network servers and workstations are priced anywhere between several hundred dollars and $20,000,[45] depending on their purpose and capabilities. General purpose workstations frequently support standard productivity applications that are found on a PC, but their additional capabilities are primarily useful for scientists, engineers and financial analysts who do elaborate data analysis or complex mathematics. Most organizations find it worthwhile to give all or most office employees their own PC (but not any of these other computer products), and then to connect PCs with one or more network servers.

Network computers – sometimes called "dumb clients" or "thin clients" – are a simpler version of a desktop computer that contain PC hardware and operating systems, but have less memory and computing power as they rely upon accessing software and hardware from other computer systems through a network. The attraction of network computers is that they offer cost savings over PCs by enabling users to place relatively cheap PCs on the desktop and then use a much more powerful server for

[44] "Server n. 1. On a local area network (LAN), a computer running administrative software that controls access to the network and its resources, such as printers and disk drives, and provides resources to computers functioning as workstations on the network. 2. On the Internet or other network, a computer or program that responds to commands from a client. For example, a file server may contain an archive of data or program files; when a client submits a request for a file, the server transfers a copy of the file to the client." *Microsoft Computer Dictionary*, 5th Edition, Microsoft Press.

[45] See www.smarter.com/sm-servers--ca-449_ch-2.html and www.smarter.com/sm-workstation--ca-247_ch-2.html, accessed May 29, 2006.

storing data and programs and for performing heavy-duty processing. The disadvantages of network computers is that they have functional liabilities due to the limits in storing and processing data locally, reliability problems arising from reliance on another computer and a network to overcome these limitations, and potential loss of convenience due to the necessity that all networked users adopt the same suite of major applications programs.[46] For stand-alone users, such as households and small businesses, network computers are a realistic alternative only in combination with high-speed Internet access and the presence of an Internet services provider that is willing to provide a general purpose network server for an array of dumb clients, which has not yet occurred.

Since the mid-1990s, many industry participants, including Microsoft executives, have viewed network computers as a serious potential competitor to the Windows/x86 PC platform. In the mid 1990s, Novell was the acknowledge leader in network computing. Novell's software portfolio, containing office productivity applications, groupware, desktop middleware and network server, made it the most important networked threat to Windows 95. Today network computing as an alternative to a fat client environment has been implemented using Linux, including a Novell system using its SuSE product and similar Linux products from Red Hat and Ubuntu.[47] Recently Dell announced that it will offer a product pre-loaded with Novell's thin client.[48] Nevertheless, thin client solutions remain more of a promise than a reality because of the applications barrier to entry. Productivity applications for Linux-based systems are

[46] "Findings of Fact," paragraphs 25-6. The last disadvantage is mitigated by the fact that Microsoft dominates all important categories of office productivity software so that "fat client" systems also all tend to use the same key applications.

[47] Sean Michael Kerner, "Novell Goes Thin at Brainshare," *Internetnews.com*, March 19, 2007, at www.internetnews.com/dev-news/article.php/3666506.

[48] Daniel Pop-Silaghi, "Dell Partners with Novell in Thin Client Distribution," *Softpedia*, February 3, 2009, at news.softpedia.com/news/Dell-Partners-with-Novell-in-Thin-Client-Distribution-103595.shtml.

compatible with some but not all of Microsoft's office productivity software, and even where compatibility has been achieved, it is after a new version has been released.

Both the District Court and the Court of Appeals concluded that other platforms are not reasonably close substitutes for an x86 PC. According to the District Court in *U. S. v. Microsoft*: "Currently there are no products, nor are there likely to be any in the near future, that a significant percentage of consumers world-wide could substitute for Intel-compatible PC operating systems without incurring substantial costs... It follows that, if one firm controlled the licensing of all Intel-compatible PC operating systems world-wide, it could set the price of a license substantially above that which would be charged in a competitive market and leave the price there for a substantial period of time without losing so many customers as to make the action unprofitable. Therefore, ... the relevant market is the licensing of all Intel-compatible PC operating systems world-wide."[49]

Although information technology has progressed since these conclusions were reached, circumstances have not changed sufficiently to cause them to be revised. In particular, the PC remains the most efficient platform for major productivity applications, such as desktop publishing, financial analysis, statistical analysis, presentations and distribution of messages and information over the Internet. Consequently, the personal computer continues to account for nearly all home and office productivity uses of computers, Wintel continues to be the dominant PC platform, and other platforms offer very little competition against it.

The conclusion that network computers are not in the same relevant market as x86 operating systems also may hinge on the presence of a barrier to entry against integrating network computers and server operating systems into the x86 operating systems market. As discussed below, Microsoft took actions to prevent competing server operating systems and cross-platform applications that could run on a networked system as well as Windows from being fully functional with Windows.

[49] "Findings of Fact," paragraph 18.

Substitution among Platform Components

The second condition for the relevant market to be the PC system is that microprocessors and operating systems are close substitutes. As a purely technical matter operating system software can be placed on integrated circuits and included in the chip set of a PC. Nevertheless, software has important flexibility advantages over hardware, as discussed in the previous section. When Intel sought to reduce its dependence on Microsoft in PC platforms by creating sets of APIs to correspond to its advancements in microprocessors, it did not attempt to place these APIs on a chip. Instead, Intel either wrote platform software or cooperated with others (Netscape and Sun) to write platform software to be used in connection with advanced features of its microprocessors. Thus, the practical business reality is that integrated circuit technology has not yet progressed to the point at which chips and operating system software are reasonably complete substitutes.

Because hardware and software are not close substitutes, each separately could be profitably monopolized. That is, a hardware monopolist or a software monopolist could increase price above the competitive level without fearing that producers of the other component would increase the capabilities of their products so that consumers could avoid the price increase by buying more of it.

Lock-in Effects

The third condition is that PC systems do not exhibit lock-in effects. When users acquire a PC platform, they must learn how to use the middleware and applications programs that it supports for performing the functions that were the motivation for buying a PC. If PC systems and the software that they support are not identical, these requirements create "switching costs" for users who decide to adopt another PC system. In addition, over time users develop a library of files in the formats of their old software. Unless all of the software that is supported by two PC platforms is compatible, these files cannot be switched to another system later without incurring significant cost. Finally, user network effects create still another switching cost. If platforms support incompatible software, another PC platform will not be as valuable to a group of interconnected users unless all switch simultaneously. Whereas users within the same organization may be able to coordinate such a switch, in most cases this will be difficult, and, in any case, if the value of file sharing differs among users who make independent decisions, some may find the switching costs offset by the potential benefits while others do not, in which case not all will switch and network benefits to all will be lost.

Recall that extremely rapid technological progress in microprocessors and memory has led to advances in software to take advantage of these capabilities. Superficially, this feature of the industry appears to present an opportunity for entry. Each time a new microprocessor is introduced, users must upgrade their software to take advantage of the additional capabilities of the new hardware. However, users still will want to access old files, and they will face less difficulty learning new bells and whistles on upgrades to their old software than an entirely new set of software products. These switching costs apply to all types of software: operating systems, middleware and applications.

Because many applications are not ported across operating systems, most users who switch platforms must acquire new applications. This element of switching costs gives rise to the applications barrier to entry.[50] Thus, users will be willing to pay a price premium to stick with the old platform and its associated software rather than switch to another of similar or even better quality and functionality. Put another way, only a very large advance in microprocessors and associated software has a chance of offsetting switching costs and attracting a significant number of users.

Summary of Platform Competition

Given that key components of PC systems – microprocessors, operating systems, middleware and

[50] "In considering a switch in operating systems, retraining was of notably less concern than compatibility with installed software and existing files." "Environments in the Microprocessor Marketplace: A Subscriber Study," *PC Magazine*, January 1991, Bates No. MSPCA 01109493.

applications – have lock-in effects, are not close substitutes, and are not all competitively supplied, the argument for a platform market that includes both microprocessors and operating systems breaks down. In addition, even if the PC platform "market" contained several distinct products that were built from microprocessors and operating systems from different sources, PC systems still would not be a relevant market if microprocessors and operating systems are not close substitutes. If not all platform functions can be incorporated seamlessly as part of either component with no penalties in terms of cost and functionality, then either microprocessors or operating systems can be profitably monopolized.

For example, merger or collusion among all firms producing operating systems would enable these firms jointly to capture the monopoly profits available from PC systems and consumers would face a monopoly price for the combination of a PC and an operating system. For this outcome not to prevail, microprocessors would have to be substitutes for operating systems, microprocessors would have to be sold in a competitive market, and users would have to be able to switch from their now-monopolized operating system to a fully microprocessor-based PC without experiencing significant switching costs.

The profitability of anticompetitive acts in operating systems is greater if operating systems differ in qualitative features. If the Wintel platform differs qualitatively from an Apple platform or an x86/Linux platform in ways that matter to some consumers, then, theoretically, monopolizing one family of operating systems (e.g., Windows/DOS for x86 microprocessors) could be profitable, in which case each family of operating systems is a separate relevant market. One important dimension of qualitative difference among operating systems is connected to the applications barrier to entry. Because MS DOS/Windows has been the dominant operating system (running on many more PCs than all others operating systems, including Apple, combined) for a quarter of a century, more applications have been developed for MS DOS/Windows than for any other operating system.[51]

For example, among operating systems other than MS DOS/Windows, only Apple supports a

[51] "Findings of Fact," paragraph 19.

GUI that is comparable to Microsoft's. PM for OS/2 also was comparable before it disappeared from the market. Although Microsoft has ported MS Office to the Macintosh, the Mac does not support many other Microsoft applications or most applications produced by ISVs for MS DOS/Windows, including Corel (WordPerfect and QuattroPro) and Lotus (AmiPro and 1-2-3), Microsoft's leading applications competitors. More generally, in *U. S. v. Microsoft* Windows was found to support over 70,000 applications.[52] By contrast, in 2003 (several years later), Apple claimed that over 16,000 applications were supported by Macintosh, and Sun reported that Solaris supported over 12,000 applications.[53]

The applications advantage of Windows enhances the value of PCs with Windows or a Windows-compatible operating system. Actions by Microsoft to reduce competition among operating systems that are compatible with Windows are more profitable than would be the case if cross-platform applications existed that could run on operating systems that otherwise were incompatible with Windows. For this reason, Microsoft has a heightened incentive to eliminate cross-platform applications such as Novell's and IBM's products in the mid 1990s. In principle, this line of argument could mean that the relevant market for operating systems is simply Windows-compatible operating systems.

Of course, neither the government nor the courts went this far in defining operating systems markets in *U. S. vs. Microsoft*, but this line of reasoning explains why in *U. S. v. Microsoft* the Apple operating system was excluded from the relevant market. The distinction between Apple and other x86 operating systems is that a consumer can not switch to Mac OS without also buying a new PC, while a consumer could switch from MS DOS/Windows to Linux without changing machines. "[T]he cost of switching to a non-Intel compatible PC operating system includes the price of not only a new operating system, but also a new PC and new peripheral devices. It also includes the effort of learning to use the

[52] "Findings of Fact," paragraph 40.

[53] These numbers were taken from the on-line catalogs: guide.apple.com/ussoftware.lasso and solutions.sun.com/catalog.html, both accessed on April 9, 2003.

new system, the cost of acquiring a new set of applications, and the work of replacing files and documents that were associated with the old applications. Very few consumers would entail these costs in response to the trivial increase in the price of an Intel-compatible PC system that would result from even a substantial increase in the price of an Intel-compatible PC operating system."[54] Thus, the distinction that keeps Apple's operating system out of the relevant market but keeps, say, Linux in is the cost of replacing hardware and software if one switches to Apple, compared to switching only software if one switches to another x86-compatible operating system.

Middleware and Applications Markets

The government's explanation in *U. S. v. Microsoft* for why microprocessors and operating systems that are not compatible with the x86 family are not part of the relevant market for operating systems provides a logical framework for addressing other market definition issues in this litigation. For example, the same logic supports the conclusion that operating systems are not in the same market as middleware that exposes APIs.

Recall the discussion about the superior efficiency of supporting multiple users that arises from incorporating many platform functions in operating systems rather than applications. In some cases, a specific type of application may require specific, distinct platform functions that are not useful to other applications, in which case this efficiency argument does not apply. In this circumstance, placing platform functions in middleware is efficient. Nevertheless, two important additional conclusions flow from this argument.

First, middleware cannot efficiently incorporate other platform functions that other types of applications will use unless it seeks to transmogrify into a middleware product for the second type of application or to a full-blown operating system that competes with the operating system that originally

[54] "Findings of Fact," paragraph 20.

supported it. The distinctive feature of middleware that distinguishes it from applications is that it exposes APIs. These APIs may perform platform functions to support one or more applications or they simply may provide a means for translating the APIs of one operating system to those of another. Middleware is distinct from operating systems in that it does not perform all platform functions that support applications, but instead, for some purposes, connects with the APIs of an operating system or, perhaps, another middleware product or the PC hardware.[55] For example, the most efficient design for middleware is to use the GUI that is already on the computer for displaying commands and files and to use the PC's operating system to control the allocation of memory to files and programs.

Second, if a middleware product provides platform functionality for a particular application, that product is in a separate relevant market from the operating system. Suppose an innovative application requires platform functions that are not provided by existing operating systems. One can imagine several possibilities: several products that provide a type of application and several distinct middleware products that instruct operating systems in running those programs, one company in either category and several in the other, one company in each, or one company providing both. The issue for defining the relevant market pertains to the first case, and asks whether a movement from the first case to the second would be profitable for the firm that gains control of that segment. In general, the answer is that monopolization of either product would increase profits by eliminating price competition for that product. Those engaging in the anticompetitive act merging either all middleware that provides this platform function or all applications that use it would be able to capture excess profits from PC users who were willing to pay more for the function performed by the new application than the cost of providing it to them, including both the application and the platform functionality that most efficiently supports the application.

A useful example is digital media players, which refers to software for downloading and playing high quality music from the Internet. To accomplish this task requires software that creates efficient

[55] "Findings of Fact," paragraph 28.

digital files of recordings that can be transmitted over the telecommunications network in a reasonable amount of time, and other software that reconstructs and plays the music from these files. The application in this case is creating, transmitting and playing music files, and the middleware is the set of interfaces between this software and the operating system that enables music to be stored, catalogued and played. Media players have some similarities to messaging systems, which require groupware on a PC for receiving, editing and sending messages, server software for managing a network of messaging correspondents, and middleware that implements communication between the application and both the operating system and server software.

Digital media players are a product market that can be profitably monopolized. A company that controls the platform features that are necessary for using computers as audio players could capture monopoly profits from computerized music delivery. Of course, the operating system that supports this middleware is a potential competitor in this market in that it can expand to include the functions performed by the middleware (as in fact happened when Microsoft introduced Windows Media Player); however, this possibility does not vitiate the fact that this particular application requires unique platform functionality that was costly to produce, which leads to the conclusion that the middleware that provided that functionality is in a separate relevant market, regardless of subsequent decisions about bundling Windows Media Player with the operating system.

One also can imagine a world in which many media players compete, but a single vendor monopolizes digitally recorded music. In this case, media player vendors do not earn monopoly profits, but the music provider can charge monopoly prices for audio files unless and until another music vendor enters the market. Hence, the provision of digital media files is a separate product from the media player.

One type of middleware product – a universal translator – does not exist, but if it did it would have significant implications for the feasibility of competition across PC platforms. This product would expose APIs that support a wide range of applications, and then would convert calls to its APIs to calls to the related APIs of each of many operating systems. In principle, this product would enable consumers who use one operating system to run applications that were written for other operating systems. If such a product existed, ISVs would have a strong incentive to write their applications to the APIs of the translator in order to have as great a potential market as possible. If this event transpired, the applications barrier to entry in operating systems would disappear, and other operating systems might become competitive with MS DOS/Windows; however, in practice, no such product exists, and even if it did, ISVs would not develop a full complement of applications that were written to its APIs until the translator became at least as ubiquitous as Windows, which probably would take years.[56] Nevertheless, the possibility of becoming a universal translator with a large installed base that could destroy the applications barrier to entry is precisely the threat that was created by products such as Netscape Navigator, Java, PerfectOffice and Lotus Notes.

The preceding discussion can be recast as raising the issues to be examined in order to define relevant markets in software. Three factors need to be taken into account in deciding whether a type of software product constitutes a distinct relevant market: functional uniqueness, independence of demand, and production uniqueness.

Functional Uniqueness

From the perspective of the user, uniqueness in functionality is necessary for a product to be in a separate relevant market. This functionality distinction requires that a category of products differs qualitatively from any other type of product in the benefits it provides to users, thereby preventing products outside the class from being close substitutes. The preceding section describes the functions performed by the major categories of software products at issue in this litigation, explaining how each has a distinct functionality. This discussion explains why the operating system, GUI, browser, spreadsheet, word processor and groupware are functionally distinct, so that a user can not substitute a product in one

[56] "Findings of Fact," paragraph 29.

category for a product in another category.

To return to the example of media players, other applications (e.g. word processor, spreadsheet, e-mail, browser) are not functional substitutes for applications that provide high-quality audio and video entertainment. Operating systems or other middleware that can not support entertainment applications are not functional substitutes for middleware such as ITunes, RealPlayer or Windows Media Player. A browser, for example, can access an Internet site where media content is stored, and so is a complement to a media player, but because the browser can not play this media content, it is not a substitute. With respect to browsers, "there is a consensus in the software industry as to the functionalities that a web browser offers a user... There is also a consensus in the software industry that these functionalities are distinct from the set of functionalities provided by an operating system."[57] Hence, operating systems, applications, browsers and other types of middleware are not functional substitutes.

Independence of Demand

Independence of demand contributes to creating a separate market. Products are substitutes if a small increase (or decrease) in the price of one product causes an increase (or decrease) in the demand for the other. For products to be in the same relevant market, they must be close substitutes. By comparison, two products are complements if an increase in the price of one product causes a reduction in demand for the other. A third case arises when there is no strong relationship between the price of one product and the demand for another. Independence of demand refers to a circumstance in which products may be used together, and may even be complements, but nevertheless the price, quality or brand of one product that consumers desire is not determined by the price, quality or brand of another product.

For example, for nearly all consumers, demand for left shoes is identical to the demand for right shoes. Notwithstanding that some future fashion guru may convince people that they want Nike on the

[57] "Findings of Fact," paragraph 150.

right foot and Florsheim on the left, current practice is to wear a matching pair. Hence, demand for shoes is not independent: once one knows which shoe a customer wants for the left foot, one also knows the preferred shoe for the right foot. However, the demand for socks – the other component of the "footwear system" – is not fully revealed by the choice of shoes, even though for most types of shoes a consumer who is wearing two shoes also will wear exactly two socks all of the time. Although shoes and socks are complements, they have independent demand. The reason is that one cannot determine a consumer's demand for any particular pair of socks on the basis of the consumer's stock of (or willingness to pay for) shoes, or vice versa. Thus, a customer who buys Nike athletic shoes will not necessarily buy Nike athletic socks. The same argument applies to components of many other "systems," such as engines and gasoline for autos, or speakers and CD players for stereos.

In this litigation, the independence issue is whether the demand for each category of software is separate from the demand for the others. This requirement is not that products lack complementarities or even are never substitutes. Instead, the requirement is that knowing the demand for one specific product is insufficient to know the demand for another. To take an example, knowing that a user uses a particular browser intensively to surf the Internet does not provide enough information to know which spreadsheet or a word processor the user prefers (if any), and how intensively these other products are used.

In practice, every x86 PC has an operating system, so that an x86 PC and an x86-compatible operating system are complements; however, some users install two operating systems on the same PC, and not all customers who buy an x86-based PC pick Microsoft Windows as their operating system. Thus, the operating system has an independent demand from the x86-based PC. This logic explains why Microsoft's "PC system" is not an appropriate definition of a relevant market. Likewise, nearly all new PCs also are shipped with a GUI pre-installed by the OEM; however, not every PC has a GUI. Today, users who pick a Linux operating system for an x86 PC do not use the Windows GUI. Instead, Linux users can choose among several GUIs that are written for Linux, and some Linux users acquire no GUI at all. Moreover, when GUIs for the x86/DOS platform were introduced, they were sold separately. Until

Windows 95, not all users of the Wintel platform purchased a GUI, choosing instead to use the character-based command shells that were and still are part of all x86 operating systems. As late as 1996, OEM sales of DOS exceeded OEM sales of Windows by two million.[58] Hence, the demands for operating systems and GUI must be independent or else no significant number of users would buy an operating system without a GUI if given the choice.

Users also vary in the extent to which they use different applications. Clerical staff in offices often primarily use word processors, while accountants make greater use of spreadsheets. In some office environments, media players have no business use at all.

The demand for browsers reveals substantial independence from the demand for operating systems despite the fact that both Windows and Apple Macintosh bundled Internet Explorer with the operating system for nearly a decade (Apple no longer bundles IE with Macintosh). In some offices, employers prefer that employees not spend time surfing the Internet, and so prefer that PCs not include browsers. "[F]or businesses desiring to inhibit employees' access to the Internet while minimizing system support costs, the most efficient solution is often using PC systems without browsers."[59] Surveys of PC users indicate that about ten percent of households that own a PC are not connected to the Internet and so are not likely to use a browser.[60] Five percent of people who have access to computers anywhere (home, work, or other) report never going on the Internet.[61]

[58] Microsoft OEMQuery 2001 database. Sales to IBM are excluded because of incomplete data.

[59] "Findings of Fact," paragraph 152.

[60] See Roger G. Noll, Dina Older-Aguilar, Richard Ross and Gregory Rosston, "The Digital Divide: Concepts, Measurement and Policy Issues," in *The Digital Divide*, California Council on Science and Technology, 2001 (also available at siepr.stanford.edu), and "Computer and Internet Use in the Unit3ed States: 2003," Bureau of the Census, October 2005, at www.census.gov/prod/2005pubs/p23-208.pdf.

[61] "Poll Shows More U. S. Adults Are Going Online at Home," *Wall Street Journal,* May 24, 2006, at

Among end-users who want a browser, "[m]any consumers desire to separate their choice of a web browser from their choice of an operating system.[62] Many users have more than one browser on their computer. The testimony of Franklin M. Fisher in *U. S. vs. Microsoft* contains substantial evidence from executives at several companies, including Microsoft, that the demand for browsers is independent of the demand for operating system, partly because users want a choice and partly because some users do not want a browser at all. Even though Internet Explorer still is bundled with Windows, thereby making IE available on nearly all PCs, IE's share of browser use has been falling. After peaking at around 90 percent, IE's share fell to about 2/3 in 2009.[63]

The normal procedure in economics for testing whether two products are substitutes or have independent demand is to determine whether their prices are highly correlated. Economists sometimes uses econometric models of price formation to determine whether prices within a hypothesized market are more highly correlated than are prices in markets with technically similar but functionally different products.[64] I have not performed such an analysis here for two reasons.

First, to perform a statistical analysis of comparative prices requires finding other factors, called identifiers, that influence the price and sales of each brand and product category. In the case of software products, the number of measurable independent factors affecting costs and demand (and hence prices) is insufficient to support this kind of analysis.

online.wsj.com/public/article/SB114840389678260791-IREjYVgN_rGLeE3_6Djin1jeJZc_20070523.html?mod=rss_free.

[62] "Findings of Fact," paragraph 150.

[63] "Top Browser Market Share Trends," accessed April 29, 2009, at marketshare.hitslink.com/ browser-market-share.aspx?qprid=1&qpdt=1&qpct=4&qptimeframe=M&qpsp=58&qpnp=11.

[64] Jonathan Baker and Timothy F. Bresnahan, "The Gains from Merger or Collusion in Product Differentiated Markets," *Journal of Industrial Economics*, Vol. 333, No. 4 (June 1985), pp. 427-44.

Second, to undertake a meaningful statistical analysis of relative price changes also requires having reliable price data for Microsoft and its competitors; however, discovery has produced only incomplete price data for OEM purchases of software from Microsoft's competitors. In addition, for much of the period at issue in this litigation Microsoft has been a virtual monopolist, so that sales prices of its competitors to OEMs reflect a very small number of transactions that frequently are subject to special provisions. For example, even though WordPerfect Office had a list price of $299.99, in 2002 Corel adopted a strategy of selling a stripped-down version of WordPerfect Office to OEMs at prices ranging from $0.25 to $0.50, apparently with the goal of subsequently selling upgrades.[65] While this strategy induced some OEMs to sell WordPerfect Office, the actual price was a temporary marketing ploy, not a true change in relative prices.

These problems provide an explanation for why in *U. S. vs. Microsoft* the government and the courts relied primarily on qualitative evidence about end-user functionality to ascertain whether products are substitutes. Notwithstanding these difficulties, the limited price data for periods in which Microsoft was not overwhelmingly dominant show that Microsoft's leading competitors in each product category have been similarly priced.[66]

MS DOS for the IBM PC was released in 1982, and revisions and updates were released frequently until MS DOS 6.22 was released in 1994, which would be the last stand-alone version (not bundled with the Windows GUI). The primary alternatives to MS DOS during this period were DR DOS and OS/2. In 1992 and 1993, the prices for MS DOS were approximately $67 and $70, respectively, compared to $68 and $60 for DR DOS. Then, in 1994 and 1995, the prices of both systems fell – to $48

[65] Various discovery documents report aspects of this marketing strategy. See Bates Nos. MSCCSUN000001546623-6 at 23, MSCCRN000000874297-301 at 300, SUN2591000004341, and MSPCAIA000000140885-86.

[66] The price data to follow are from the Retail Software Report, NPD Intelect Market Tracking.

and then $43 for MS DOS, and to $54 and then $51 for DR DOS. PC DOS was more expensive, with prices of $125 and $111 in 1992 and 1993; however, the PC DOS price then fell to $73 and $78 in 1994 and 1995. Obviously, these price changes were broadly similar during the period when there was maximum competition among operating systems. By comparison, Linux operating systems have been priced much lower. Since 1995, Linux OS from various vendors has varied between $25 and $50. Nevertheless, these systems have not attracted significant business from Microsoft, with penetration varying between two and five percent of x86-family PCs.

Comparing Windows with other GUIs during this period is essentially impossible. The only possible comparison is between OS/2 (with Presentation Manager) and a bundle of MS DOS and Windows. In 1993 and 1994, IBM sold the then-current release of OS/2 for $45 and $38, respectively, but this product lacked the features of the MS DOS/Windows bundle. During the next two years, OS/2 was a more realistic substitute and was priced at about $123, compared to $136 and $137, respectively, for the combination of MS DOS and Windows 3.1. Windows 95 was priced at $179, $184 and $191 in 1995 through 1997, compared to $122, $163 and $188 for OS/2. By the end of 1997 OS/2 was essentially dead, making further price comparisons irrelevant. Since 1999, IBM's market share in x86-compatible operating systems (OS/2 plus PC DOS) has been effectively zero.

The first version of Microsoft Word for DOS was released in 1984. In 1993, the price of Word for Windows was $289, and it fell to $286 in 1994. WordPerfect was priced around $257 in both years. AmiPro (IBM later changed the name of its SmartSuite word processor to WordPro) was priced at $248 in 1993, but slashed its price to $108 in 1994. In 1995, when Windows 95 became available, Microsoft's price increased to about $297, and then in 1996 to $301, while WordPerfect increased its price to $263 and then $272. AmiPro continued to drop to $95 and then $72. Since 1996, Word has been priced between $300 and $310, while WordPerfect is no longer available as a separate product. AmiPro's price continued to drop and in 2001 stood at $63. Its successor, WordPro, is only available as part of SmartSuite. These data show that WordPerfect prices stayed somewhat below those of Word, but together these product prices moved together. AmiPro's prices were far lower, yet it did not manage to capture much of the market. Because almost all word processors have been sold in suites since 1996, price comparisons after this date have little meaning.

In the late 1980s and early 1990s, the leading spreadsheet programs for DOS and Windows environments were Excel, Lotus 1-2-3 and QuattroPro. Since 1993, Excel for Windows prices have shown little variability, hovering between $297 and $307 in all years except 1999, when the price was $350. In 1993, Lotus 1-2-3 was $12 more expensive than Excel, but the gap narrowed until in 1997 and 1998 1-2-3 was slightly cheaper (by less than $1). In 2001, 1-2-3 was about $13 cheaper than Excel. By contrast, QuattroPro was far less expensive than either of these. Whereas QuattroPro was priced at about $189 in 1995 (still far below the others), in all other years between 1993 and 1998 its price has been below $100. QuattroPro and Lotus 1-2-3 are now available only as part of suites.

Comparing word processors and spreadsheets, Microsoft seems to have had one significant competitor in each category – WordPerfect and Lotus 1-2-3, respectively. The third product in each case clearly is a weaker competitor, with a low price and a low market share in individual sales.

Office suites have come to replace separately sold word processors and spreadsheets. IDC reports that by 1998, 98 percent of spreadsheets and 95 percent of word processors were sold as part of suites. Suites have been priced in a manner that implies that they are not a separate relevant product market. When both suites and separate productivity products were available, Microsoft Office was priced at roughly 75 percent of the sum of the prices of Excel and Word, and other software vendors followed a similar strategy. PerfectOffice was priced at 80 percent of the combined prices of WordPerfect and Quattro Pro in the last year (1996) one could buy both products separately. Lotus SmartSuite was been priced at roughly the sum of the AmiPro and Lotus 1-2-3 prices after 1994. Because AmiPro was priced far below Word and WordPerfect and QuattroPro was priced substantially below 1-2-3 and Excel, the prices of the PerfectOffice (then Corel's WordPerfect Office) and SmartSuite bundles have been somewhat below but broadly similar to the price of Microsoft Office.

Since 2000, the standard edition of Microsoft Office has been priced at nearly $400, with higher prices for versions containing more applications. WordPerfect Office has been priced at about $300, and SmartSuite about $235; however, the latter two products do not contain groupware except for an e-mail application, whereas MS Office contains Outlook. The fact that the latter two suites do not contain a complete groupware application product is evidence that the groupware and other office productivity applications are in separate relevant markets.

Microsoft's strategy of pricing its suites substantially below the combined prices of its components has led to the dominance of suites. Once a user has decided to buy one of the two core products, the incremental price of all the additional software in the suite is very small. Regardless of why this pricing strategy emerged, it has two harmful effects on users. First, it discourages users from assembling their own "best of breed" bundle, such as combining, say, WordPerfect, Lotus 1-2-3 and Power Point, without buying all three suites. Second, moving the applications markets to bundles created a barrier to entry in each applications category, thereby reducing both innovation and competition.[67]

In the case of browsers and GUIs for MS DOS, Microsoft combined MS DOS and the Windows GUI in Windows 95, and in 1996 added Internet Explorer. This bundle has remained in place through all subsequent releases of Windows. Netscape and later Mozilla give away the most important alternative browser, and no competing GUI product for the x86 PC has significant penetration. Initially browsers were sold separately and commanded a positive price. Before Microsoft began giving away IE and before it was bundled with Windows, Internet Explorer was part of a package of "frosting" that users could buy at extra cost to accompany Windows 95.[68] Until 1999, both Netscape and Internet Explorer technically

[67] Barry Nalebuff, "Bundling as an Entry Barrier," *Quarterly Journal of Economics* Vol. 119, No. 1 (February 2004), pp. 159-187, and "Exclusionary Bundling," *Antitrust Bulletin* Vol. 50, No. 3 (Fall 2005), pp. 321-370.

[68] "Findings of Fact," paragraph 137.

were available for sale at around $40, but in 1999 the price of Navigator fell to zero (where Firefox remains) for all distribution channels and IE fell to under $11. Today the price of IE is zero as well.

Distinctive Production

The third factor that contributes to the existence of separate relevant markets is distinctiveness in the production process for two categories of products. In some rare cases (not computer software) production differences are present between competing products – one can imagine competition in local travel between buses and subways, even though each service is produced very differently. Here the focus is on the activities that are necessary to produce a product. The key activity in producing software is programming. Software that is nearly identical in terms of the actual line-by-line code clearly is the same product, but this is too strict a test for whether two products have distinct production. Software that performs a particular function for end users can be written in many ways, so that functionally equivalent software may have little overlap in the actual structure and code of the underlying programs. Consequently, physically examining a program on a line-by-line basis does not necessarily determine whether one program is a distinct product in comparison with another.

For the purpose of market definition, the issue is whether one type of program is so close in design and structure to another that, at a low cost, the former could be transformed into the latter. Supply substitution of this form is unlikely to be plausible if each particular type of software product must perform unique functions that cannot be performed by segments (or, in the jargon of programmers, subroutines) that were written for other types of program. While all of the reference products (operating system, GUI, browser, groupware, word processor and spreadsheet) have some overlap in design and structure (for example, they all have printing commands), these overlaps do not include most of the structure of any of the separate products. Moreover, the task of developing software products is organized according to functional groups that concentrate on a particular product.

The technical issues in software design have important economic implications. All of the software at issue in this case has two important features: programs in each type are extremely large and complex, and programs in each type make use of links to programs of other types. Because of the first feature, the optimal design of a program is to divide it into a large number of distinct mini-programs, or modules, each of which performs a narrow, specific function. This procedure facilitates collaboration among many programmers to create a product, and substantially eases the task of identifying and correcting bugs. For this reason, the idea that categories of programs are in a meaningful sense merged when they are bundled together or when they interact is simply incorrect. Typically when programs are "integrated" (bundled together with links among them), virtually none of the underlying code is changed, and the products remain fully distinct. Thus, "integration" is not a basis for concluding that a combination of functionally distinct software is a single product in a new relevant market of the combined functionality of the separate programs.

Another important economic implication of the process of program creation is that enabling interoperability among applications is a platform function, not an applications function. Integration typically means allowing different programs to access the same files and creating a mechanism whereby one program can call on the other to perform a function. The code that accomplishes this task is part of the platform. For cross-application interaction to work requires creating technical standards for file formats and interfaces between programs, but these are precisely platform functions. APIs are technical standards governing interfaces, and file formats are technical standards for storing data which then can be shared across applications to the degree that each application finds useful. Nothing about this form of "integration" causes the products to lose their distinct functionality or physical separation, and so to be properly regarded as merged into the platform software that performs the integration function.

As an economic matter one might regard two types of programs as a single product for purposes of market definition if they always were used in a way that combined their functionality. For example, if MS DOS/Windows rarely was used without invoking the functionality of Internet Explorer, and this technical reality reflected a huge overlap between the two in the code required to make each work efficiently, then one justifiably would conclude that the two products were really the same. But this degree of integration does not exist between any two of the products at issue in this case.

For Internet Explorer, which was commingled with Windows 98 more than in Windows 95 and more than any of the other products have been commingled, the overlap with Windows still was tiny. Moreover, Microsoft is the only operating system vendor that does not offer consumers and OEMs a choice of browsers. "A number of operating system vendors offer consumers the choice of licensing their operating systems without a browser. Others bundle a browser with their operating system products but allow OEMs, value-added resellers, and consumers either to not install it or, if the browser has been pre-installed, to uninstall it." Moreover, Microsoft "has always marketed and distributed Internet Explorer separately from Windows in several channels."[69]

The same argument applies to the distinction between the operating system and GUI features of Windows. GUIs and operating systems are functionally distinct products, but more importantly, the code for the GUI remains separated from operating systems functions, as when the products were sold separately in the 1980s. The fact that Microsoft decided to bundle improvements in MS DOS and its GUI did not make these the same products either from a technical standpoint or from the perspective of user functionality. Microsoft is not alone in bundling a GUI with its operating system. So do Apple and Sun. Linux does not have an associated GUI that is bundled with the operating system, but instead offers customers a choice among separate GUIs that have different appearances and functional features.[70]

Porting and Market Definition

As discussed above, an application that is written for one operating system generally will not run

[69] "Findings of Fact," paragraph 153.

[70] See, for example, A. Russell Jones, "Linux vs. Windows: Choice vs. Usability," DevX.com, August 13, 2003, at www.devx.com/opensource/Article/16969.

on another operating system without rewriting it to access a different set of APIs. Porting can be achieved by an ISV by rewriting its program, by an operating system vendor by creating a program that translates the APIs of another operating system into its APIs, or by a third party that creates a middleware translator, but in all cases porting is costly. For example, as discussed in the section about the development of OS/2 in the late 1980s and early 1990s, several applications vendors faced substantial delays and costs in porting products that they had written for OS/2 to Windows. Because porting is costly, a significant and permanent price difference between equivalent applications that run on separate operating systems would not necessarily induce competition to arise through porting. Thus, one legitimately could conclude that the relevant product markets for middleware and applications products are limited to those that are supported by Microsoft operating systems.

Nevertheless, in subsequent analysis I assume that the applications markets are broader, and include applications that are written for any of the operating systems that are compatible with x86 microprocessors. I make this assumption because I have found relatively little information about porting costs for most applications and because many significant applications have been ported across operating systems. In any case, the choice between these two market definitions has little practical significance in that Microsoft is dominant in these markets even with the broader market definition.

Geographic Market

The geographic component of software markets is the world. Software can be distributed directly to computer users over the Internet at very low cost regardless of its point of origin and the location of the end user. A single master copy of a software product can be delivered to a reseller or even to final customer that owns a large number of PCs at extremely low cost, which then can be loaded onto numerous PCs at extremely low cost. Software that is sold to computer manufacturers as a master copy can be loaded onto new computers as they are produced. Because geographic location has almost no effect on the total cost of a delivered software product, supply substitution among locations causes the relevant geographic market to be the world.

Software products possibly could differ with respect to the language seen by the user; however, because of the ubiquitous use of English, apparently no significant software product is not offered in English. The costs of adapting software to a new language apparently are quite low. Microsoft produces versions of all of the software at issue in this litigation in many languages – as of 2002, Microsoft Office XP was available in 35 languages, including Slovenian, a language spoken by approximately two million people.[71] Hence, apparently the relevant markets in this case have no important linguistic boundaries.

Summary of Market Definitions

An operating system is a distinct product. Its function is to enable programs to make use of computer hardware to run applications. Operating systems have independent demand, and have distinct technical characteristics that make other products at best imperfect substitutes. Product differentiation, switching costs, and the applications barrier to entry explain why the operating systems for PCs that incorporate different (incompatible) microprocessors are in different relevant markets.

These factors plus the costs of porting (which is a form of switching cost) explain why each type of middleware and applications product is in a separate market, and similarly are separated by types of PC architecture. The relevant markets in this case pertain to middleware and applications that are supported by operating systems and microprocessors that are compatible with the Wintel platform. Each of the categories of middleware and applications programs at issue in this litigation are separate relevant product markets because, from a user standpoint, they perform different functions and have independent demand, and, from a supplier standpoint, they require distinct design and structure (and hence independent development costs) to bring to the market. Moreover, even though many products are sold as part of

[71] www.microsoft.com/office/faq.asp.

bundles, bundling became ubiquitous primarily because the dominant firm, Microsoft, has marketed its products in that way. Moreover, many products that compete with components of Microsoft's bundles were available only in bundles in 1994, and many still are available as stand-alone products.

MARKET POWER

Market power is the ability to control prices or exclude competitors in a relevant market. As discussed in the section about the economics of the industry, the technology of software production implies that software markets always will be imperfectly competitive and that viable firms will have some control over prices. Thus, we should not be shocked to find that Microsoft possesses market power in all relevant product markets at issue in this litigation. Nevertheless, all of the common indicators of market power support a much stronger conclusion: Microsoft enjoys a degree of market power that is extremely high in comparison with even the most successful firms in either software or other industries with similar characteristics. Antitrust economics makes a qualitative distinction between *market power* and *monopoly power*. The latter refers to a degree of market power that makes the firm immune from competitive forces. Although this distinction is not quantitatively precise, and therefore is one that normally I do not make, in this instance if any firm has ever possessed monopoly power, it is Microsoft.

Measures of Market Power

Economists use several methods for measuring market power. With respect to market power in setting prices, the three most commonly used are market concentration in the presence of barriers to entry, the Lerner Index (the mark-up of price over marginal cost), and profitability. All three indicators not only show that Microsoft has market power, which is hardly a surprise for a viable software company, but that it ranks at or near the top in all measures of market power among U. S. companies. In addition, the ability to exclude competitors and to engage in price discrimination also are indicators of market power.

Concentration

Market concentration refers to the extent to which sales are accounted for by a small number of firms. Economists generally regard concentration as a useful but not determinative indicator of market power. While denigrating concentration measures in antitrust litigation is a popular sport among some economic consultants, the leading textbooks in antitrust economics continue to teach the most commonly used measure (the Herfindahl-Hirschman Index, or HHI, which is the sum of the squares of the market shares of the firms in the industry). They do so because, as a theoretical matter, the HHI has value.[72] Empirical research in the field of industrial organization continues to use the HHI to explain market performance. Federal antitrust agencies continue to use concentration as an indicator of the effect of a merger on market power.[73] Economics research shows that concentration does not necessarily imply market power, that the market power arising from a given level of concentration depends on technical conditions in the market, and that the extent of market power in markets with the same degree of concentration tends to differ according to attributes of the products, but that regardless of these caveats a firm with substantially more than half of total sales or the leading firm in a market containing only one, two or three firms virtually always enjoys considerable market power.

All of the relevant markets in this case are highly concentrated, well beyond the point at which one would expect to find competitive behavior. The HHI for each market is shown in Table 2. For every product in every year except for word processors in 1986-88, the HHI is well beyond the benchmark of

[72] "There are various indices of seller concentration. The most common is the aggregate share of the four largest or eight largest firms. A better index, however, now used by the enforcement agencies and beginning to creep into the case law as well, is the Herfindahl (officially the Herfindahl-Hirschman Index, or HHI)." Richard Posner, *Antitrust Law*, University of Chicago Press, 2001.

[73] Federal Trade Commission and U. S. Department of Justice Antitrust Division, *Horizontal Merger Guidelines*, April 1997.

2,000 at which one expects to find less than robust price competition and monopoly profits. By the end of the 1990s, the HHI for every market was over 7,500, an impressive statistic compared to the benchmark value of 10,000 under complete monopoly.

In every market, Microsoft is the leading firm. Table 3 shows Microsoft's share in each of these markets for most years from 1987 to 2000. At the beginning of the period Microsoft already was the leading firm in both x86 operating systems and GUI for use with MS DOS-compatible operating systems.

Microsoft's market share in x86-compatible operating systems was lowest in 1992, when the combined penetration of IBM (OS/2) and Digital Research (DR DOS) peaked; however, even at this low point, Microsoft's market share exceeded 75%, which is far above the level at which a firm can expect to have substantial market power. After 1992, Microsoft's market share in the relevant operating systems market climbed steadily to above 95% by the end of the decade.

In 1988, Microsoft was not the dominant supplier in office productivity software, but in the early 1990s, Microsoft became dominant. In 1997, Microsoft became dominant in browsers. In every case Microsoft's market share is high enough that one would expect Microsoft to possess substantial market power – that is, sufficient market power to earn profits substantially in excess of the return on investment that one would find in competitive markets with similar risk and production technology.

Because the x86 family of microprocessors dominates the PC business, these conclusions are not the result of restricting the market definition to x86 PCs. The only significant alternative to x86 PCs was Apple, before Apple switched to the x86 platform in 2006-7. If Apple were included in these markets, Microsoft's market share would remain well above the level at which the leading firm is likely to enjoy substantial market power. In software, Apple PCs use the Mac OS, GUI and media player, but the leading office productivity software on the Mac is a version of Microsoft Office that includes Word, Excel, PowerPoint and Entourage, a groupware application. By providing the most important office productivity applications for the Macintosh, Microsoft does not even face a threat that the core applications programs from the second-leading PC platform will be ported to Windows.

Barriers to Entry

High concentration and a high market share for the leading firm are reliable indicators of market power in the presence of barriers to entry. Several barriers to entry are important in markets for complex software products.

One barrier to entry is intellectual property rights. Software is protected by copyright, which means that no other firm can enter legally by simply copying the code from the incumbent's software. In addition, a "logic algorithm for processing data that implemented via stored instructions" also can be protected by patents, in which case a firm can enter legally only if it invents a non-infringing method to perform the same function that is performed by the patented procedure.[74] Although software patents initially were not particularly important to the profitability of software firms, changes in patent policy in 1995 substantially increased the market capitalization of software companies.[75] The increase in market value arising from stronger patent rights implies that the IP barrier to entry increased.

Another barrier to entry is the high fixed cost (first-copy cost) of large, complex software products. High fixed cost implies that a firm must make substantial sales in order to recover the cost of developing a new product.

Finally, network effects and switching costs create still another barrier to entry. Users must invest in learning to use a software program, and want to be able to continue to access files that were created using their prior set of software. Users also derive benefit from using the same software that collaborators use. All of these create an additional barrier to entry, and for operating systems these are summarized as the applications barrier to entry.

[74] James Bessen and Robert M. Hunt, "An Empirical Look at Software Patents," Working Paper No. 03-17/R, Federal Reserve Bank of Philadelphia, March 2004, p. 8.

[75] Bronwyn H. Hall and Megan MacGarvie, "The Private Value of Software Patents," Working Paper 12195, National Bureau of Economic Research, May 2006.

The history of the markets for operating systems, GUIs, office productivity applications, and other middleware bears out the difficulty of entry. For several years the only competitors to Windows have been Linux and some GUIs that can run on it, and the Solaris Unix system (which bundles an operating system and a GUI). Linux is completely open-source software, and Solaris also comes in an open-source version. Open-source systems are maintained and upgraded by a volunteer user community. The last fully proprietary commercial operating systems were DR DOS and OS/2, which disappeared from the market in the mid-1990s. In the absence of barriers to entry, Microsoft's high profits and high market concentration would induce competitive entry.

Lerner Index

The Lerner Index measures market power as the fraction of the sales price that is a mark-up over marginal cost.[76] All software products are likely to have extremely high mark-ups because their marginal cost, or the additional cost of duplicating, distributing and supporting one more copy of a program, typically is so low. For Microsoft products that are sold to consumers on diskettes or CDs, marginal cost is at most a few dollars, which is a few percent of Microsoft's sales price.[77] In addition, for all office

[76] Microsoft uses the term "gross margin" to express this concept. "You subtract the costs that are specific to that one copy, so, for example, if you're selling it in a box, the cost of the box and the disk, or if there is support that you have to give for that copy, you would subtract that out as well, or if there's a royalty you're paying to a third party. So those would be, I think -- I think everything you subtract to get to what's called a gross margin." Deposition of William Gates, Volume II, February 28, pp. 542-3.

[77] Microsoft documents provide indirect information about the plausible magnitude of marginal cost through statements about the cost of goods sold (or COGS). For example, in 1994 Microsoft estimated COGS for Office 4.2 as $6.02, and in 1990 estimated COGS for DOS 5.0 Upgrade as $9.02 on a 3.5" diskette and $2.03 for multiple licenses based on a single copy, which is an upper bound for costs to

applications sales and for Windows sales that are not through the OEM channel, the expected support costs are between fifteen and twenty percent of Microsoft's gross revenues.[78] Thus, for sales of physical copies either directly or through retail outlets, Microsoft's gross margin is on the order of 75 percent. For Windows sales to OEMs, the Lerner Index is between 90 and 95 percent.[79]

Other features of Microsoft's pricing provide additional evidence of substantial market power. One example is extensive price discrimination among categories of users who buy the same products. For example, the Microsoft Enterprise Agreement for businesses sets prices on the basis of a firm's "previous commitment to Microsoft technology," which includes the proportion of the firm's PCs that are shipped with Office and Windows.[80] Unlike modest volume discounts (based strictly on quantity sold), discounts based on the fraction of sales can not possibly be based on cost.

In 1998 Microsoft announced a new corporate software subscription program that required customers to subscribe to all Microsoft upgrades and new versions for three years. For companies that normally do not purchase every upgrade of Microsoft products, the plan imposed a price increase that the Gartner Group, a corporate research firm, has estimated to vary between 33 and 107 percent.[81]

In *U. S. vs. Microsoft*, the District Court observed that "Microsoft's actual pricing behavior is

OEMs. Bates Nos. FLAG 00000059155, MSPCA01109178.

[78] Deposition of William Gates, Volume II, February 28, 2002, p. 423.

[79] According to William Gates, in response to a question about the gross margin in OEM sales: "There isn't much marginal cost there because their -- most of the support work is done by the OEMs, so itwould be in the 95 percent range gross." Deposition of William Gates, Volume II, February 28, 2002, pp. 422-3.

[80] "EA Price Grid: Executive Overview of Microsoft Enterprise Agreements," May 5, 1998, p. 3, Bates No. MSPCA01227418.

[81] Joe Wilcox, "New Microsoft Licenses May Increase Costs," May 10, 2001, CNET News.com (news.com.com/2100-1001-257390.html), accessed August 3, 2002.

consistent with the proposition that the firm enjoys monopoly power in the market for Intel-compatible PC operating systems."[82] Examples of monopoly power in operating systems (actually, the Windows bundle of operating system and GUI) that the Court cited[83] were:

* Ignoring prices of competitors in setting prices for Windows 98;

* Raising the price of Windows 95 before Windows 98 was released when one would expect the price of an obsolete version to fall, not rise, when a better version becomes available;

* Engaging in price discrimination against Gateway and IBM when they resisted Microsoft's actions to preserve the applications barrier to entry;

* Imposing burdensome restrictions on its customers; and

* Microsoft's calculation that a $49 price for the Windows 98 upgrade to Windows 95 would be profitable, but deciding instead to charge $89 on the basis of another calculation that the latter price would maximize revenues.

The Court also noted that Microsoft's anticompetitive actions against potential threats to its operating system monopoly "could only have been advantageous if they operated to reinforce monopoly power."[84]

The testimony of Franklin M. Fisher provides additional evidence to support the conclusion that Microsoft has market power in the relevant market for operating systems. Professor Fisher cites deposition testimony by officials from Gateway, Hewlett-Packard, Micron, NEC, Packard Bell and Compaq to the effect that OEMs have no meaningful alternative to the Microsoft operating system.[85] Professor Fisher also cites the deposition testimony of Joachim Kempin that in setting the OEM royalty

[82] "Findings of Fact," paragraph 62.

[83] "Findings of Fact," paragraphs 62-6.

[84] "Findings of Fact," paragraph 67.

[85] Direct Testimony of Franklin M. Fisher, pp. 22-5.

rates for Windows 98, he did not consider the prices set by other vendors.[86] The failure to consider competitors in setting prices implies that the firm has monopoly power.

Profitability

Profits are another indicator of market power. High profits signal other firms that a market can support profitable expansion of output, and so indicate that expansion by smaller firms or entry of new ones is likely to be desirable. If a firm can sustain high profits over a long period of time, it has durable market power in that others have been unable to expand production and compete successfully against it.

Because Microsoft has sold the products at issue in this litigation in two bundles – Windows and Microsoft Office – one can not unpack Microsoft's profitability on a product line basis. Instead, one must rely on the overall profitability of the company and its own internal statements about its sense of the course of its profits.

Short-term profit as reported in an annual financial statement frequently is not a reliable indicator of market power. Annual data can be distorted by unusual events, such as the recession that began in 2008. In addition, annualized profits are affected by somewhat arbitrary accounting rules regarding the evaluation of long-term assets. Examples are depreciation rates for capital investments, the practice of counting research as a current expenditure rather than an investment, and provisions for uncollectible accounts. But Microsoft's profitability has been so high for so long that one can not reasonably quarrel with the conclusion that the company is among the most profitable in the world.

The *amicus* submission on remedies in *U. S. vs Microsoft* that I wrote in collaboration with Robert Litan, William Nordhaus and Frederic M. Scherer contains a calculation of Microsoft's profitability that was undertaken by Professor Nordhaus.[87] According to this calculation, Microsoft's

[86] *Ibid.*, p. 29.

[87] "Remedies Brief of Amici Curiae," at http://aei-brookings.org/admin/authorpdfs/redirect-