

The Era of Microsoft? Technological Innovation, Network Externalities, and the Seattle Factor in the US Software Industry

Edmund A. Egan

**Working Paper 87
January 1996**

Edmund A. Egan is a Doctoral Candidate in the Department of City and Regional Planning at the University of California at Berkeley.

Abstract

Microsoft Corporation, the largest company in the US software industry, has been under anti-trust scrutiny from the Department of Justice for most of the 1990s. In 1995, its planned acquisition of Intuit, Inc. prompted a Silicon Valley law firm, on behalf of unnamed complainants, to submit a White Paper to the DOJ, on the subject of Microsoft's long-term strategy. The White Paper, relying on the theoretical concepts of network externalities and lock-in effects, argues that Microsoft will use Intuit's products to attain monopolistic positions in network operating systems, on-line services, and electronic commerce, and will eventually be in a position to affect the *content* transmitted over electronic networks.

This paper disputes that claim. First, an analysis of Microsoft's growth vs. the US packaged software industry as a whole is presented, indicating that Microsoft actually has a fairly small share of total employment and sales. Secondly, a detailed review of the White Paper's argument is followed by a discussion of Microsoft's competitors, whose products also benefit from network externalities and lock-in effects. Ultimately, innovation will be more important than leverage for Microsoft. However, the paper argues that Microsoft's location in Seattle may prove to be a liability when it comes to rapid innovation; the corporation has grown much more rapidly than the Seattle software industry as a whole. Consequently, it cannot rely upon ready access to experienced workers in fields outside its traditional areas of specialization.

Key Words: Microsoft, Software industry, Strategy, Seattle, Lock-in effects

Introduction:

A consensus appears to be emerging concerning the US software industry: slowly but surely, its center is gravitating to Seattle. Microsoft's strategic control of PC operating systems, with Windows in the early 90s and now with Windows NT and Windows 95, has positioned it to capture not only key PC application markets like word processors and spreadsheets, where it has long had a presence, but new and emerging markets like enterprise servers, database management systems, multimedia tools and titles, interactive television, and electronic commerce. Why is Microsoft so dominant? According to Microsoft, it's because they make quality software; according to other observers, it's because of anti-competitive practices.

The most powerful adherent to the latter view has been the US Justice Department, which has been investigating Microsoft since 1991, and began seriously impeding its business and acquisition strategy in 1995. The DOJ effectively blocked a merger with Intuit, but backed away from prohibiting the bundling of Microsoft Network, an on-line service, with Windows 95.

Interestingly, the political pressure to put Microsoft under the anti-trust microscope has apparently not come from small companies, the usual supporters of such action, but from larger technology companies in Silicon Valley, from IBM, and also from the U. S. banking industry (Pare, 1995).

The question is even more interesting in that many insiders are concerned that these companies are openly inviting the federal government to regulate this very new industry. Among a whole host of regulatory issues: the scope of intellectual property protection, piracy enforcement, trade policy, consumer privacy, public interest issues, and anti-trust, no one in the industry appears to know what the right answers are. There is a very real concern that improper anti-trust regulation could damage the US's biggest exporting industry. Yet leading companies are inviting the government in - all because of the threat of Microsoft.

Moreover, while IBM's troubles are well enough known to explain why it would seek help from its old nemesis, the Justice Department, why the California companies? Is Microsoft really threatening Silicon Valley? Is the software industry really concentrating so rapidly that one company could attain total dominance - even allowing uncompetitive practices? Could California's decades of technological superiority really be overturned in this critical industry, by the targeting, linking, and leveraging of a single company in Seattle? Or is Microsoft, as its advocates claim, just another company that happens to produce quality software?

The case of - and against - Microsoft raises interesting organizational, and geographical issues, as well as regulatory ones. From an organizational perspective, most research on high technology industries emphasizes their dynamic industrial structure, and the relative superiority of vertically-disintegrated networks over integrated corporations (Saxenian, 1994; Scott, 1989, 1993). If Microsoft, a vertically integrated corporation, were really able to attain a monopolistic position in the software industry, it would clearly call this theory into question. The organizational question is related to a geographical one; Microsoft is located in Seattle, which, Boeing notwithstanding, is not a traditional center of high technology on the order of Silicon Valley, Los Angeles, or Boston. Were Microsoft to truly dominate the software industry, it would presumably do more than re-orient the spatial hierarchy of the software industry, it would call into question the very explanatory power of a spatial hierarchy in this industry, and lend support instead to a analysis of the software industry as footloose, or having a locational pattern that is subjectively determined by the preferences of successful corporate founders. However, such a conclusion would, in turn, tend to repudiate a significant amount of evidence that innovative high technology industries - including the software industry - do in fact exhibit strong tendencies towards spatial concentration in a few select areas (Hall, et. al., 1983; Park and Lewis, 1991; Lundmark, 1994; Egan, 1994).

The Technological Dynamics of Software

From the early 1960's until today, a truism, known as Moore's Law, has held in the semiconductor industry: the processing speed of chips of a given cost will double every 18 months (Cringley, 1992). Although the law may fail to hold in the future, because of quantum complications with extremely small chips, the years of its validity have left a significant mark on the computer and software industries. Technological advances in semiconductor design have resulted in the ability to inscribe ever more circuits on an ever smaller chip, producing greater processing speed memory. Since CPU chips essentially provide a service called "processing power", this is a tantamount to a remarkable increase in productivity. Such changes permit larger, and more complex, software systems, and induce a wider demand for software, because faster chips bring more and more of society's information-processing tasks within the feasible range of the computer.

In contrast, process innovation in software development has been chronically slow. As early as 1968, cost over-runs and reliability problems with many high-profile software projects led many to conclude that we were facing a "software crisis", caused by the haphazard and unsystematic nature of the programming process; as computer memory and programs, became larger, the chance for error greatly increased (Brooks, 1975; Naur and Randell, 1969). In large defense-related projects, the crisis took the form of unreliability in budgeting and unpredictability in the final system; in business applications the crisis took the form of an increasing backlog of work in MIS departments, which were both overworked and notoriously unaccountable to line departments (Friedman and Cornford, 1989).

The relationship between productivity improvements in these complementary sectors is complex, but important to understand. A CPU chip, like an Intel 486 chip, is a semiconductor with some software already installed; why is additional, "external" software necessary? Flexibility and cost-reduction in the information-processing system as a whole is the aim of the division between hardware and external software. There is no technical reason why a chip could not be specially designed and custom-built for every purpose, and indeed this is done in many cases. The reasons this is not universally done are economic, not technological. A re-programmable chip lowers its final cost to each user by spreading its development costs across as many users as possible. The hardware system is temporarily "customized", i.e. specified for a particular task, by the specific external software it works with. Such a relationship permits the market for mass-produced hardware to exist, but forces the work of customization to the software side.

A similar division occurs within the external software system, between applications and operating systems. All software transforms input information, according to fixed rules, to produce more useful output information. Applications are the software that users interact with to improve the usefulness of information; they tend to be tailored to a specific domain, such as engineering, statistics, graphic design, etc. An operating system (OS), on the other hand, is a more general-purpose, piece of software which is typically pre-installed in a computer; its function is to automate certain complicated, but frequently used, interactions with hardware. In this way, operating systems serve to lower the cost of applications development, by permitting different applications developers to all use the standard OS software. The division which occurs within the chip, between internal (standardized) and external (customized) software is replicated

within the external software system, and for the same reason: it permits the maximal re-use of a scarce resource - software - across many users. However, in the same way that the chip design constrains what the operating system can do, the operating system constrains what applications can do.

Finally, this division also occurs within applications software, between standardized packaged applications, and on-site customization of these applications. This customization can take many forms: an end-user using a spreadsheet to build an inventory or payroll system, a skilled programmer using a packaged programming language to write custom applications for clients, etc.

Because innovation in semiconductors has been so relentless for so long, there is a constant shift in the boundaries of these categories, but certain trends can be determined from the history of the computer industry. The earliest computers lacked operating systems at all; "applications" were written by highly specialized final users, and were designed to answer specific, calculation-intensive questions (Friedman and Cornford, 1989). As mainframe computers diffused to large corporations, operating systems and programming languages were invented to permit quicker applications development by a lower-skilled set of programmers (Fischer et. al., 1983). Computer companies like IBM provided the operating system software and languages, and worked with the MIS departments of the corporations to write specific applications to be used internally. Later on, in the 1970s, independent software companies helped to further reduce the time and cost of software development, by providing application platforms, such as database management systems or fourth generation programming languages, which provided some of flexibility of operating systems, yet were tailored, like applications, to more specific domains of use.

With the invention of the microprocessor and the personal computer, computer use diffused to individuals and small businesses, which were not linked to internal MIS departments. To meet this demand for software, the packaged software industry truly began to grow, producing high volumes of innovative applications, such as spreadsheets, for a much wider market (Keizer, 1992).

In other words, there has been a clear trend in the industry: as the cost of processing power declines and the installed base of computers increases world-wide, the pressure to create *appropriate* software, in *widely diverse* contexts, increases. Since there are nowhere near enough

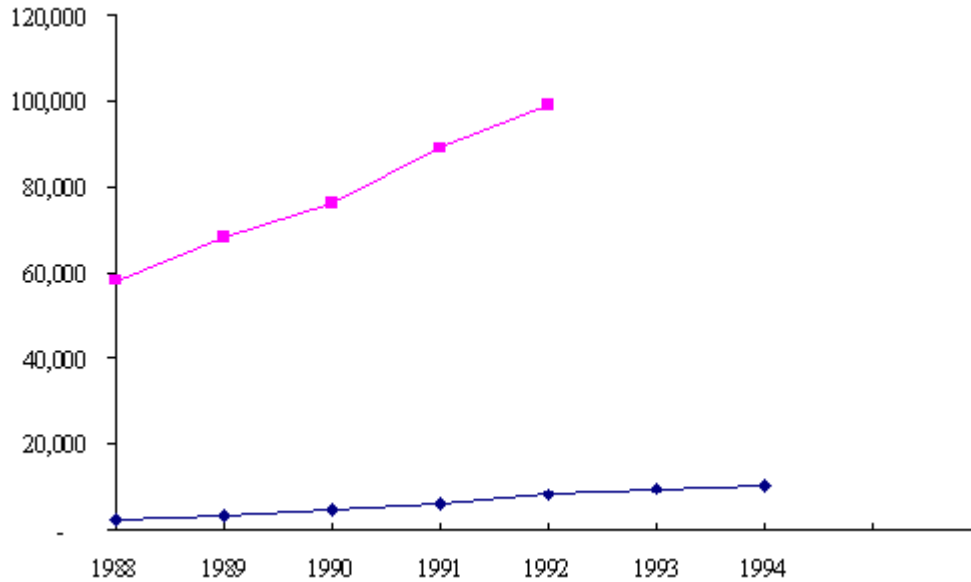
skilled programmers to write each application from scratch, and since programming is still, inherently, a craft, this calls for two types of solutions. On one hand, there is strong pressure for standardization, so different pieces of the final software system can be variously developed by system software companies, application companies, MIS departments, or end users, and yet be compatible with one another. The movement towards so-called "open systems" is symptomatic of this trend¹. On the other hand, there is a greater demand for packaged software which is targeted to a specific domain, and yet amenable to further customization, by MIS departments, external programmers, or the end user. The second trend reduces the demand for MIS programming labor and lowers the cost of software; the first trend complements the second by tending to create an applications market of maximal size. With these essential dynamics of the industry in place, we can now begin to analyze Microsoft's role in the contemporary software industry.

Microsoft: How Dominant?

Is Microsoft really dominating the US software industry? If by dominating, one means that Microsoft is making more and more of our software, and soon will be making all of our software, the answer is clearly no. Figure 1 shows packaged software employment in the United States and at Microsoft, from 1988-1994.

¹ In contrast to earlier practices in the computer industry, when computer vendors strategically bundled proprietary operating systems with their computers to prevent unwanted compatibility with other vendors' offerings, "open systems" feature compatibility between the hardware and software of different vendors (Senn, 1995).

**Figure 1:
Employment Trends in Packaged Software:
Microsoft's Domestic Employment, and the US Total, 1988-94
(number of employees)**

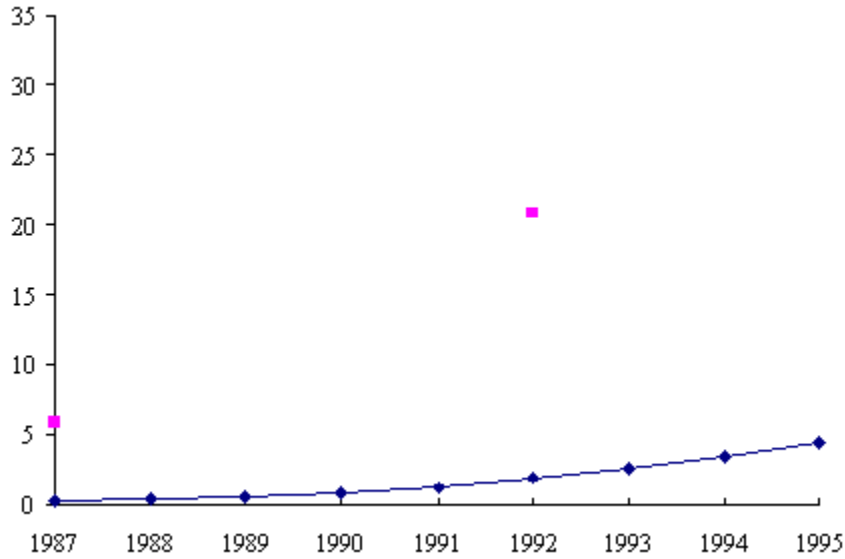


Source: County Business Patterns, Microsoft SEC 10-K filings.

While Microsoft's domestic employment has grown considerably faster than the industry as a whole, with a remarkable 28% average annual growth rate over the six year period, Microsoft has never accounted for more than 8.2% of US employment in packaged software. Its slower domestic employment growth during 1993 and 1994 means that share has probably declined since 1992.

What about market share? The essence of the fear surrounding Microsoft is tied to the monopoly profits it receives by controlling the OS standard, with Windows, in the Intel-PC desktop market. Since, by 1995, Windows was installed in 85 million PCs (Caldwell and Stahl, 1995), perhaps Microsoft's revenue share far outstrips its share of employment. However, the following chart shows that Microsoft only earns a small share of the total US packaged market, about 9%, according to the 1992 figures.

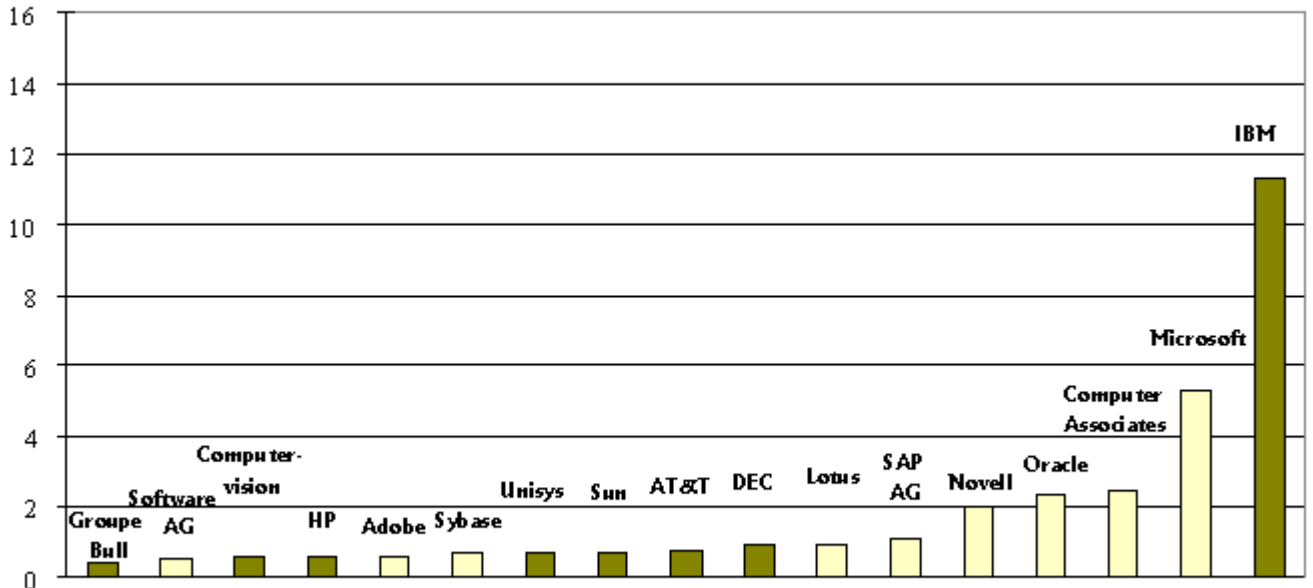
**Figure 2:
Sales Trends in Packaged Software:
Microsoft's Total Domestic Revenue, and the US Total, 1987-95
(\$US Billion)**



Source: US Census of Service Industries, 1992, 1987, Microsoft Annual Reports

Finally, amidst the concern about "software giant Microsoft", there is still the fact Microsoft is merely the largest *independent* software vendor in the world. However, not only do all large corporations produce vast, if unmeasurable, quantities of software, other computer companies sell software as well. Foremost among these is, still, IBM, as figure 3 indicates.

**Figure 3:
Leading Software Vendors:
1994 Worldwide Revenues from Software
(\$US Billion)**



Note: Dark Shading indicates a company which also sells hardware.
Source: Leibs et. al. (1995).

IBM actually sells twice as much software as Microsoft does; AT&T, a \$75 billion company, and Hewlett-Packard, a \$25 billion company, are also significant software producers. The US software industry as a whole: software services, packaged software, and computer integrated systems design, is a \$77 billion business², and the United States has 60% of the world market for software and related services (Siwek and Furchgott-Roth, 1993). Moreover, this is still excluding the work of the internal MIS departments. As of 1990, only 22% of US programmers and systems analysts worked for the software industry: the rest wrote software in other, user, industries³. In other words, Microsoft's \$5 billion in annual sales make it no serious threat to monopolize the industry in the near future.

² An International Data Corp. estimate, cited by Ann Winblad in Upside (1995).

³ This figure was determined by examining the distribution of the programming occupations: computer scientists, programmers, and systems analysts, across industries in the 1990 Occupation and Industry subject report of the US

Then why the anti-trust concern about Microsoft? The answer, and also the more interesting questions, begin with *where* Microsoft has its sales and market power. The market for *personal computer* software, as opposed to *all* software, is a considerably smaller - in the neighborhood of \$10-\$15 billion a year⁴. Microsoft is a major factor in this segment, which includes operating system and application software for the Macintosh and Intel PCs; its share is in the neighborhood of 35-40% (Cortese, 1995; Pascal, 1994a). Nevertheless, even this number is far below IBM's market share in the 1960s, which hovered between 65% and 70% for computers, software, *and* services (Fischer et. al., 1983; Sharpe, 1969).

However, the reason I have not been examining the PC segment all along is because the complaint to the Justice department went far beyond simply the PC software market. A White Paper, written by a Palo Alto law firm with the help of two Stanford economists, suggests that Microsoft is well positioned to use anti-competitive practices to control the *entire* software industry, the on-line services market, electronic commerce, and eventually, the very *content* of the information on the network. Now, such an Orwellian scenario is more than slightly intended to provide the government with ideological ammunition for anti-trust action. However, its arguments are both novel and persuasive, and it represents one of the few published overviews of trends in the US software industry. For this reason, despite its evident bias, we can learn a great deal about Microsoft's role in the software industry by reviewing and evaluating the White Paper.

The White Paper Case Against Microsoft

In 1995, the trade publication *Upside* published a condensed version of a "White Paper" on the subject of Microsoft's strategy and plans (Reback, 1995). This paper, written by the law firm of Wilson, Sonsini, Goodrich, and Rosati, was prepared at the behest of the unnamed complainants against Microsoft, and was submitted to the Justice Department.

Two events precipitated both the White Paper and the current round of government investigation: Microsoft's attempted acquisition of Intuit, a Silicon Valley developer of the personal finance packages Quicken and TurboTax, and the planned bundling of its on-line

Census. 22% of this occupational group worked for the industry group "Computer and Data Processing Services", or SIC 737; 78% worked for other industries.

⁴ Some estimates put it a good deal lower, on the order of \$7.3 billion for 1994 (*MacWeek*, 1995) Software written for a Microsoft OS (DOS, Windows), accounts for 85% of this total; the rest is Macintosh software. Microsoft, incidentally, is also the leading producer of application software for the Macintosh. If the MacWeek figure is correct, then Microsoft's share of the PC market is closer to 60%.

service, Microsoft Network, with Windows 95. As a result of the pressure, the intended merger with Intuit was called off.

With respect to Intuit, the paper's argument begins very simply. Personal finance software is a major applications segment in the PC software business, and Microsoft attained only 10% of the market on its own with its product, Microsoft Money. Quicken, on the other hand, is one of the most popular PC applications in history, and has 70% of the market. Therefore, Microsoft wants to buy Intuit, and this will cause excessive concentration in this segment of the industry. What about the much-vaunted low barriers to entry in the software industry, in which the only necessary investment is a computer? The paper argues:

"... high barriers to entry appear to reflect the substantial lock-in effects that exist in both the personal finance and tax preparation markets. In both cases, once customers have loaded what can be extensive financial data onto the program and become familiar with how the program works, switching costs can be very high ... Because of the substantial lock-in effects in these markets, it takes time to build up market share ... Even if a new product were announced tomorrow, it is very unlikely that the product could achieve substantial market penetration for several years. And in those intervening years, there appears to be little or no chance that Microsoft will not have consolidated its complete domination of these markets."

This is the central theoretical argument in the paper: "lock-in effects"--from whatever source - inhibit a switch to a better product, even if one exists. The users are, to some extent, at the mercy of the software provider. If the provider decides to bundle the locked-in product with an inferior complement, the complement will obtain a higher market share than it would unbundled, because some users will buy the bundle just to get the new version of a product they are locked into.

According to the paper, the fact that the acquirer in question is Microsoft only makes matters worse. Any company could buy Quicken and possibly lock-in a monopoly in personal finance software. Microsoft not only obtains lock-in through user acceptance of applications, but, more importantly, it has a lock-in monopoly in the operating systems market because of network externalities.

Network externalities refer to an old reality that has recently resurfaced in industrial economics: liquidity increases with the number of buyers and sellers in the market. Because of economic uncertainty, asset-holders value the liquidity of their assets, and simply being in a larger market conveys a costless benefit to them--an externality. Another way of defining network externalities is to say that the economic value of joining the network increases with its size (Economides, 1994).

In an industry, the network refers to the web of institutions and material links through which intermediate goods pass during the production process: hubs are material and information-processing sites, and links represent transportation, communication, and exchange. A dense network, with many links, indicates high liquidity, i.e. a higher chance for a buyer or seller to exchange an asset for money at its market price, at the place and time of his or her choosing. The classic examples are transportation and communication networks: the liquidity they convey, i.e. the ability to transmit material and information to others, increases with their size. As a consequence, their value to an agent increases with their size. This is why there is a lock-in associated with products that exhibit network externalities, even without considering switching costs; despite the presence of a new and superior substitute, consumers would have to forego the larger network surrounding the older product.

In the software industry, the network is comprised of the distinct hardware and software elements that comprise the information-processing chain, and their markets: according to the White Paper it moves from the user, to applications, to the desktop ("client") operating system, to the network ("server") operating system, and then to the hardware. Network externalities can appear at any point in this chain.

Clearly, aggregate network externalities are maximized by the concentration of all exchange into one standard market--one stock exchange, one rail network, one telephone network - because this structure most closely approaches the ideals of perfect competition in downstream markets, and maximal liquidity.

However, the question of control arises when that one market is private property: the provider of the networking service is a natural monopolist and could extract the full value of the network externalities as rent⁵. Furthermore, if the network-monopolist is also a competitor in the downstream industry, there is the possibility of anti-competitive vertical integration, with monopoly occurring in the downstream industry as well.

This is precisely the fear which the White Paper expresses about Microsoft. Operating systems are like network hubs: applications and hardware have to be compatible with them to pass data through them, therefore to maximize compatibility (liquidity), it is to everyone's advantage to have one standard operating system. The paper asserts that Microsoft has used, and

⁵ Extracting the full value, however, would eliminate the lock-in effect.

will continue to use, its monopoly control of operating systems to leverage control of up- and downstream markets:

"...control of certain layers in the various markets of the network create greater potential for leverage than control of other layers. In particular, there are a few 'gateway' layers into the network. Control of these layers represent the most effective platforms for leverage. Generally speaking, the OS layers represent the most powerful platforms for both horizontal and vertical leverage."

In particular, the paper repeats the oft-made claim that Microsoft used its DOS monopoly in PC operating systems, given to it by IBM, to "migrate" users to a new platform, Windows, over which it had total control. It then leaked Window specifications to its own application developers, and they were able to release Windows-compatible spreadsheets and word processors before their main competitors, Lotus and WordPerfect. As a result, Microsoft's products had an early lead in the new Windows applications market, which was cemented by later versions⁶. The idea is not that Microsoft made other company's applications totally closed to Windows, just delayed them; it needed Windows applications to build the Windows market. However, once enough application developers had moved from DOS-based to Windows-based applications, Windows became the standard. At that point, Microsoft could make minor revisions in the Windows source code which could make competing products incompatible: this strategic leveraging of network externalities would cut into the Windows market, but it could give the applications division a major boost. This is one of the major reasons, according to the paper, that Microsoft now has such a commanding lead in PC application software⁷.

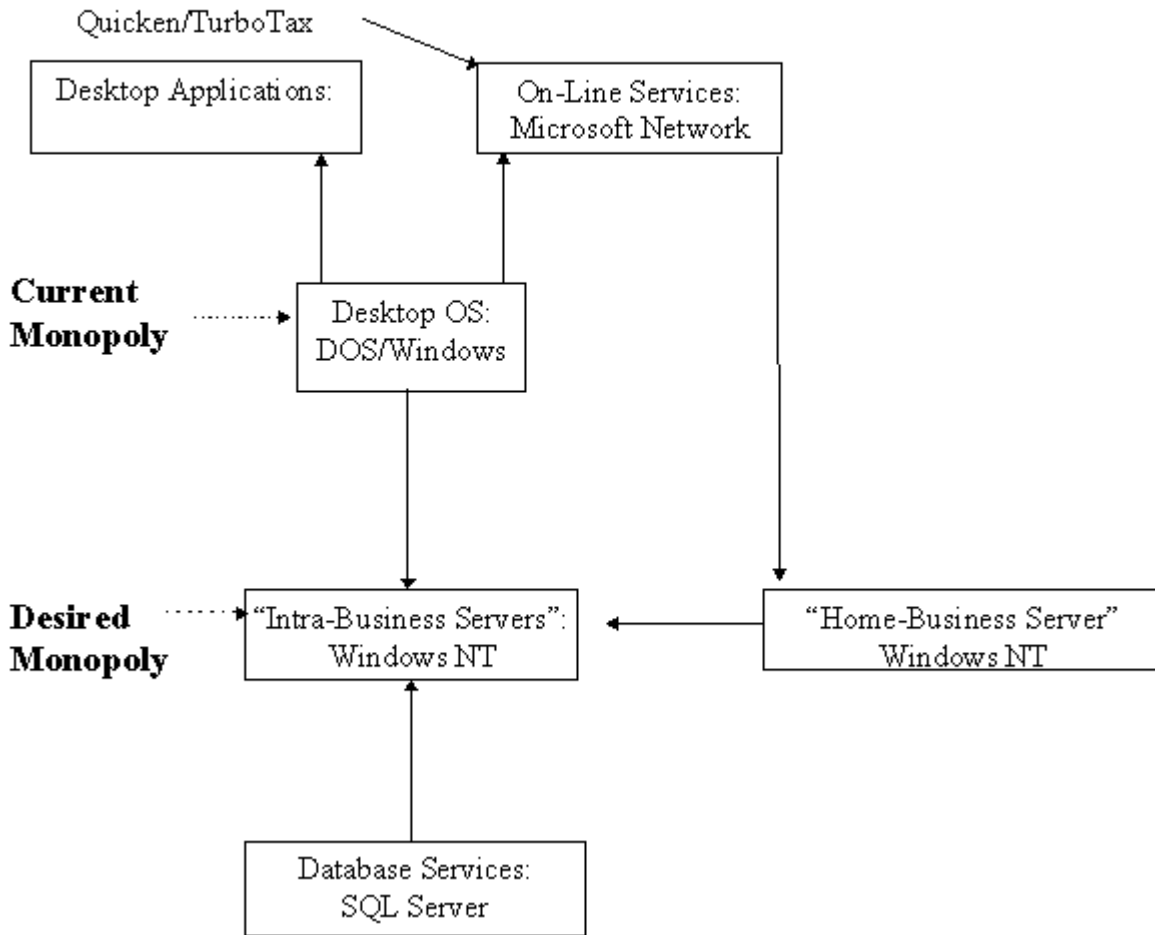
Furthermore, the paper asserts that the PC, or desktop, market is only the beginning. Microsoft is planning to use the same tactics to control the market for network operating systems, or server software. According to the White Paper, Microsoft expects that 300 million intra-business (or "intra-net") servers will eventually be used, connecting computers, cash registers, photocopiers, etc., as the advantages of data communication and distributed computing among computers become clearer. The server software that manages these client-server networks will in turn need to be networked to other servers, creating server-server compatibility issues. Potentially, massive network externalities could accrue around a network operating system

⁶ However, WordPerfect has since come back to tie Microsoft's Word in the word processing market (Computer Retail Week, 1995).

⁷ Another frequently mentioned method is for Microsoft to use its OS power to force application developers to agree to one-sided "joint development" agreements. Intuit's relationship with Microsoft began in this situation; they had preliminary talks, Intuit explained how Quicken worked, and Microsoft pulled out and introduced Microsoft Money less than a year later.

standard, which doesn't exist at the present time. Microsoft's product in the market, Windows NT, now has a weak position, but the paper argues that it has several points of leverage, as figure 4 shows:

Figure 4: Microsoft's Leverage in the Software Industry, according to the White Paper



Upward leverage from Windows 95, the desktop OS. As of 1995, the leading local area network (LAN) server software is Novell's Netware, with about 60% of the market. An IBM product is second; Windows NT is third (Marion, 1995). Microsoft would have to create compatibility problems between Windows and Netware, and hope that users continue to buy Windows upgrades despite the problems, preferably solving them by switching to Windows NT.

Using Microsoft Network to leverage the "Home-Business Server" market. At present, as the paper admits, this market does not exist. The paper presumes that Microsoft Network, together with Quicken, will become the market leader in on-line banking, and points out that

Microsoft and Visa are co-developing a security standard for electronic commerce. As a result, Microsoft could leverage its way to dominance in the home-business server market, when on-line shopping from the home becomes more of a reality. With this market under control, Microsoft could create server-server compatibility problems between home-business and intra-business servers, and this would encourage enterprises to choose Windows NT as the intra-business network operating system.

Downward leverage from the database services market. Perhaps the most important function of the network operating system is to distribute requests to, and output from, enterprise databases. Oracle is by far the leading company in this market, and Microsoft has never been able to produce a competing product in-house. In 1993 Microsoft acquired the rights to Sybase's SQL Server database; the two companies now develop their versions of the product independently. Microsoft makes sure that its SQL Server is absolutely compatible with Windows NT, and, according to the paper, has told customers that it intends to make desktop Windows (Windows 95) less compatible with competing database products.

As a result, Microsoft has three major leverage points to get Windows NT into a dominant position in the server market. And what would be the consequence of this maneuvering? The paper claims:

"The economic characteristics of the technologies and markets at issue here differ markedly from other, more conventional industries, in that these products (software products) and markets (networks) exhibit 'increasing returns,' also sometimes called "network effects.' ... Application of 'increasing returns' economic analysis would reasonably predict that, given the present situation, Microsoft will succeed in monopolizing the entire network (just as it has monopolized the desktop) and that the monopoly will remain in place for a very long period of time ... Application of the literature and logic of increasing returns economics to Microsoft's strategy ... strongly indicates that ... the forces of the economy will not be sufficient to impede Microsoft's move toward dominance."

In essence, the paper is arguing that the move to client-server networks is forcing a convergence in the software industry, eliminating the traditional distinction between PC, minicomputer, and mainframe computer software markets. In the new market, there are only servers and desktop clients. Microsoft controls the desktop, no one company controls the server, and therefore Microsoft is uniquely positioned to manipulate the lock-in effects attaching to its many successful products. By interfering with a competing product's compatibility with its standard, Microsoft can assist its own technically weaker entries in linked markets to become

standards themselves. Because, according to the paper, only Microsoft develops both operating systems and applications, only Microsoft has this capacity for leverage. Therefore, the paper argues that the Intuit merger be stopped, and that Microsoft should be split into two companies: one making applications, and the other making system software.

A Critical Examination of the White Paper Claims: Timing is Everything...

The White Paper makes a persuasive case against Microsoft; subsequent events have confirmed that Microsoft's strategy is very similar to that described in the paper⁸. The question is: can it succeed? The White Paper ignores the market analysis discussed earlier; that Microsoft is a fairly small part of the entire US software industry, which is at stake here, and that IBM actually sells twice as much software as Microsoft.

In fact, the paper gives short shrift to a whole series of obstacles in Microsoft's path. In the first place, the very idea that Microsoft has a monopoly in desktop operating systems is itself suspect; the Apple Macintosh has 15-20% of the personal computer market, and this number has not gone down significantly despite the rapid growth of the personal computer industry. Now one can always claim that Microsoft has a monopoly position in *Intel-based* personal computers, but this is simply semantics: personal computers are one market, and when consumers choose not to buy an Intel machine, they are also avoiding Microsoft Windows. In order to be assimilated into the Microsoft universe, the Macintosh OS would have to be made compatible with Windows, and Apple can use some of the same kind of leverage to prevent this, by adding its own incompatibilities in later versions. However, such a move will likely cost both companies.

For that matter, the same thing applies to the workstation market, where Sun, Silicon Graphics, Hewlett-Packard, and IBM, are the leading companies (Alexander, 1995). Workstations are traditionally a different market, but as PC's become more powerful and get networked, and workstations come down in price, they increasingly become substitutes. These machines typically run a version of the UNIX operating system, a clear competitor to Windows NT. Since the move towards client-server computing means that workstations and PCs will

⁸ In the summer of 1995, Microsoft has announced that its standard for exchanging data among applications, OLE, will be extended to work on other operating systems, such as UNIX and the Macintosh. The technology behind OLE was inspired by a Hewlett-Packard product called OpenDoc, which is more open and much more popular with the industry. A cross-platform OLE would extend Windows's leverage across competing operating systems.

increasingly need to be networked, the optimal choice for the user is to be able to integrate the two seamlessly; they will then be able to avoid switching costs. The more Microsoft tries to fight with the UNIX companies over a network operating system standard, the more of a market there is for a third party to create software that makes them run smoothly.

Thirdly, there is the database services market. Oracle has plans to become a standard-holder in its own right, by creating the database server that will drive electronic commerce over the Internet. While Oracle's products do run on top of operating systems, it has characteristics of operating systems too: developers write applications to specifically query Oracle databases. Naturally, Oracle supports this, since it solidifies its position in the database market, but if push comes to shove, its control over the specifications can be used to threaten developers who write for other database platforms, such as Microsoft's SQL Server.

In fact, many common applications have this feature; applications do not only have to run off of operating system platforms, but may run off of application platforms as well. As I said earlier, there is no cut-and-dried definition of what an operating system must be; it is simply software that serves to facilitate the creation of applications by automating frequently written code. As the computer market grows, it is not surprising that hybrid "application/platforms" should emerge that facilitate the programming process for one type of applications, such as database query, a specialty of SAP AG, one of the largest and fastest growing software companies in the US (Lieber, 1995). Groupware, a type of software that permits selective users on different networks access the same documents, is an excellent example of this. IBM's recent acquisition of Lotus was not intended to gain control of 1-2-3, its best-selling product, but rather Lotus Notes, the leading groupware application. Many industry insiders see Notes as a potential Windows-killer, despite the fact that it is not an operating system. Developers write Notes applications to customize the groupware for the specific needs of the organization. The organizations are thereby locked-in to Notes, in that switching to a new groupware product means re-writing those applications. Groupware, like WorldWide Web Internet browsers, and multi-media authoring tools, can exhibit network externalities, are not Microsoft's strong suit, and are not discussed at all in the White Paper.

Finally, there is the matter of IBM, and its \$11 billion annual software sales. While it is true that client-server computing doesn't bode well for future sales of mainframes, IBM has been selling mainframes for 40 years, and those machines can work perfectly well as server machines.

Because IBM has a hardware standard in these markets, it can control their software as well. In the mid-1980s, IBM, initially with Microsoft's assistance, undertook a highly ambitious development project to link its leading mainframe, minicomputer, and PC operating systems into a single architecture, called SAA (Webb, 1988). Although Microsoft's defection from the project, and subsequent success with Windows, set SAA back considerably, IBM still has considerable leverage at the upper end of the market.

Clearly, it is not the case that, as Microsoft publicly claims, there is no leverage, and that software products succeed purely on their merits. On the contrary, from Microsoft's perspective, there is too much leverage. Every innovation in software creates new leverage; users get locked into new skills, MIS departments and third-party developers get locked into new platforms, software companies get locked into hardware, and vice-versa. The point is, they get locked into *different* companies' software, and every time this happens, the purely vertically-integrated system gets harder and harder to achieve. There is an endless trade-off between technological innovation and network externalities; everybody wants compatibility, but everybody also wants the best product (Bacon, 1985). The paper's static view of software, in which the seven segments of the industry will "hold still" while Microsoft maneuvers its way to dominance, is the biggest flaw in its logic, if not its conclusion. Network externalities are presumed to be the only force in software. In fact, a new product does have to worry about the installed base (lock-in) of its competitors, but sufficiently superior technology can displace it. What other possible force could cause the fantastic pace of change that we have witnessed in computers and software? If network externalities and the power of the standard simply increased monotonically with the size of the installed base, IBM would still control everything, just as it did in the 1960s.

Microsoft does have the advantage of being an integrated company, being able to adjust the design and marketing strategies of products in different markets according to a central strategy, such as making Windows NT the standard network operating system. Its competitors are busy leveraging their own products and attempting to establish their own standards, and no individual company is so well positioned as Microsoft. On the other hand, Microsoft has to innovate against the entire industry; a single Notes or Netscape, can set back, or even force a fundamental alteration, of its strategy.

Nor can Microsoft afford to simply sit on its PC market, despite the fact that Windows 95 does appear destined to succeed. IBM in particular has had its competing product, OS/2 Warp,

on the market for some time, but, partly because Microsoft has not released OS/2 versions of its best-selling applications, IBM's product has been leveraged into a tiny market share. However, in a networked world, IBM may be able to exert downward pressure from its mainframe networks; what happens to Windows if IBM is eventually able to capture the network operating systems market? Microsoft will suddenly be much more willing to write OS/2 applications, and support greater OS/2 - Windows compatibility.

Paradoxically, we will only be able to evaluate Microsoft's real weakness by disclosing one of its real strengths: despite what many people say, Microsoft has had to innovate. It has not had to be the most innovative company in the industry, but it has had to innovate. The White Paper understandably proffers the idea that Microsoft was given its DOS monopoly by IBM, and has done nothing since then except collect rent and consolidate control. But an operating system is not valuable real estate, it's a piece of technology that offers use-value, and has to compete with other products that offer similar services. Had Microsoft been content to stay with DOS in the mid-1980s, other companies could have beaten Microsoft with a competing operating system, or perhaps the whole Intel-PC market would have lost ground to the Macintosh. The internal specifications of DOS were very widely known: why didn't anyone beat Microsoft to market with a Windows-like product?

The real question then is not: can Microsoft innovate, but rather, can Microsoft innovate well enough, quickly enough, for the task at hand? *The packaged software industry, with its multiple points of leverage, is a network of monopolies.* Its companies are engaged in monopolistic competition in the market for information-processing. For Microsoft, or anyone else, to sustain its impressive growth rate, it has to use leverage *and* produce quality products quickly to surpass the various monopolies in different points in the network. The longer it takes to develop a product of sufficient quality in a key target market, the more network externalities will accrue to its competitors, and the harder it will be to eventually leverage the leader. What determines how quickly innovation can occur? This is where Microsoft's relationship to the spatial hierarchy of software technology becomes of primary importance.

...And Location is Timing

Part of the reason Microsoft has a reputation for not being a technological leader is because it is far away from the established centers of the industry, Silicon Valley and Boston. In

the late 1970s, when Microsoft had decided to move away from its original location in Albuquerque, Northern California was considered as a location. However, Microsoft's founders rejected California because they felt the presence of other high-tech companies would be too difficult to keep key employees; other companies would be tempted to hire them away (Manes and Andrews, 1993). Seattle was, and to some extent still is, a perfect alternative, and it was the hometown of Microsoft's founders and most of its early employees. Of course, a peripheral location is a double-edged sword, as hiring experienced employees from outside of the company can be difficult. They can, of course, be tempted away from the industry centers, but doing so means not only switching jobs but switching locations. In an industry center, skilled workers have the chance to acquire the hottest skills, command the highest salaries, develop new products, and start new companies.

At Microsoft, they get stock, which can be very tempting; the company's stock options have produced thousands of millionaires since the company went public in 1986. Every employee has an opportunity to buy stock at 85% of the market price, and particularly valued employees are awarded additional stock. Microsoft has a strong internal labor market - its 7-10% annual turnover rate is very low for the industry (Cusumano and Selby, 1995). Typically, Microsoft hires college graduates right out of school, and works them through the internal labor market; until very recently, the average age of the company was below thirty (Microsoft, 1995). This hiring and compensation strategy, like its locational strategy, has two aspects: on one hand, labor costs can be low and retention is easier; on the other hand, Microsoft's new employees lack experience, and the corporation, in effect, subsidizes their on-the-job learning.

When new products in a relatively protected market are doing very well, as Windows was in the desktop OS market in the early 1990s, this need not be a problem. However, a recent article suggests that Microsoft's move towards networked, enterprise, computing means that it that it needs an influx of skills (Johnston, 1995); it is now competing in new areas, against companies with their own installed bases. In the late 1980s, for example, the entire development team for Windows NT, which is very well respected technically, was hired away from a DEC office in Seattle, as Zachary's excellent book *Showstopper!* describes (Zachary, 1994)⁹. Had this office been located in Boston, Windows NT would have been a vastly different product.

⁹ According to Zachary, the location of the DEC group in Seattle was no case of wanting to be "where the action is." The development team was headed by David Cutler, a DEC developer who had fallen out with management. He was

Ideally, Microsoft would like to be located in a labor market that permits them to both to quickly and easily hire the most qualified people, and yet still be able to preserve continuity in development teams and retain key people. If Seattle had a diverse and sizable software industry, apart from Microsoft, it could share the sunk costs of training a pool of young workers; if the industry was also very diverse, this would help Microsoft develop internal expertise in the new technologies. Angel (1991), and Saxenian (1994), attribute much of Silicon Valley's innovative character to the loose connections between firms and workers, producing a vibrant external labor market and numerous start-ups. Microsoft could take advantage of that kind of environment in Seattle; hiring and retaining workers should certainly easier now that it ever had been before; Microsoft's stock has been rocketing upward for years. Despite the fact that its annual sales place it 250th in the Fortune 500; it has the 10th highest market value: two notches below IBM, which outsells Microsoft by approximately 13:1 (Fortune, 1995). There should certainly be no problem in arranging sufficiently attracting stock deals to attract the right people. When Microsoft's primary competitive challenge is innovating in areas outside of its traditional specialization, doing so quickly, and integrating these products into its existing line in a strategically valuable way, helping to build the external labor market is the only alternative. It seems unlikely that Microsoft could leave Seattle for another city, and simply acquiring the technology - and the experienced developers - through merger raises the question of how to integrate it, quickly, with Microsoft's other offerings.

On this latter point, Microsoft has been quite active in recent years, and it has a strong tendency to relocate the developers from the acquired companies to its Redmond, WA headquarters. Table 1 lists Microsoft's acquisitions since 1989, the location of the acquired company, and how many employees were relocated to the Seattle area.

granted a development team to work on a next-generation product, and decided to locate it in Seattle, because it was so *far away* from DEC headquarters in Boston. When DEC pulled out of the project, Cutler's whole team was hired by Microsoft to develop Windows NT. Zachary's book is, among other things, a very interesting account of the difficulties of incorporating older, more experienced, programmers into the youthful campus atmosphere of Microsoft.

Table 1. Microsoft's Software Acquisitions since 1989

Company	Location	Year	Relocation?	Segment
Netwise	Boulder, CO	1995	49 employees	Networking
Network Managers	London, UK	1995	20 developers	Networking
RenderMorphics	London, UK	1995	no	Animation
Altamira	Mill Valley, CA	1994	4 employees	Animation
Softimage	Montreal, QB	1994	no	Animation
WebCorp	Sausalito, CA	1993	< 100 employees	Networking
Fox Software	Perrysburg, OH	1992	50 developers	Database
Consumer Software Inc.	Vancouver, BC	1991	< 85 developers	E-mail
Bauer	San Jose, CA	1989	20 employees	Printer Software

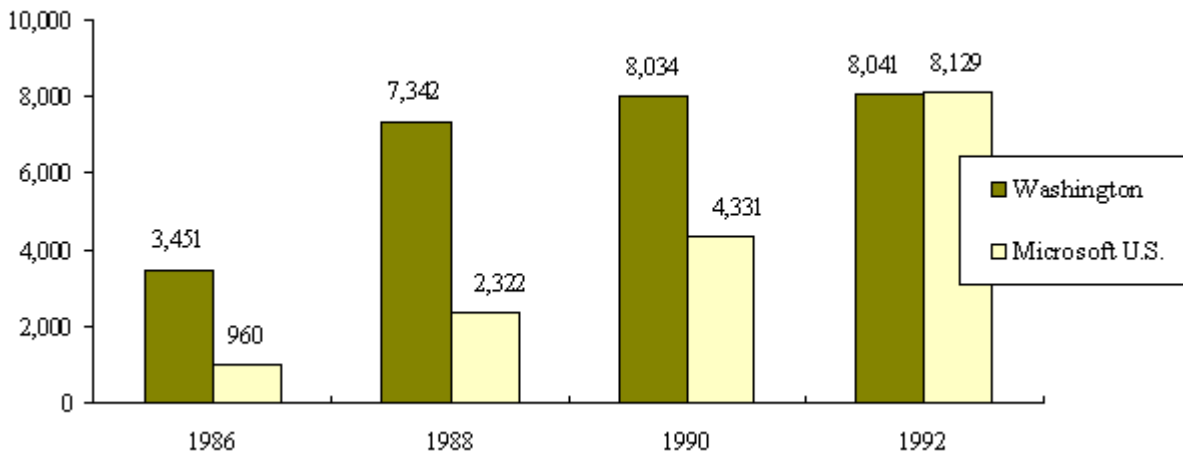
Sources: Netwise - Cox (1995).; Network Managers - Foley (1995); RenderMorphics - Clancy (1995); Rupley (1995); Altamira - Morrissey (1994); SoftImage - Computer Graphics World (1994); WebCorp - Morrissey (1993); Fox Software - Daly (1992), Hawkins (1992), Flynn (1992).; Consumer Software Inc. - Tanaka (1994); Bauer - Ryan (1989).

In total, between 300 and 400 software developers were brought to Redmond as a result of Microsoft's recent acquisitions. This is no trivial number, and it should be remembered that these workers are highly skilled in very strategic fields. In the case of the Fox Software acquisition, for example, the buy-out was apparently contingent upon the employees' consent to relocation (Hawkins, 1992). Nevertheless, from 1989-1994, according to Microsoft's 10-K filings, its employment in Product R&D increased from 1,288 to 4,417 (SEC, various years). This implies that approximately 90% of Microsoft's new software developers came to Redmond by other means, such as directly from college. In other words, Microsoft's acquisition strategy, while bringing some key people, has probably not fundamentally changed its recruiting strategy.

As for the broader Seattle software industry, if the most up-to-date government data can be relied upon, it seems that Microsoft itself accounts for the lion's share. Although breakdowns of Microsoft's local employment over time are not available, in 1995 over 50% of its total employment was located in the Puget Sound area (Microsoft, 1995) However, Microsoft's fortunes thus far appear to have little to do with the fate of the Seattle software industry as a whole. Figure 5 shows Microsoft's growth has been spectacular, while the Seattle industry stagnated from 1988-92; the recession of the early 1990s appears to have hit the local industry particularly hard. The employment total for Seattle in Figure 5 includes not merely packaged

software development, but customized programming services and computer integrated systems design as well¹⁰.

**Figure 5:
The Washington State Software Industry,
and Microsoft's US Employment, 1986-92.**



Source: County Business Patterns, Microsoft Annual Reports

Perhaps Microsoft has no real dependence on the external high-tech labor market; perhaps it is able to pay whatever it takes to bring in the best people from anywhere in the world. Perhaps all of the products it needs to leverage its way to dominance are already in place, and need only minor improvements. If, on the other hand, innovation in the software industry as a whole is liable to continue, then Microsoft will have to be everywhere, in every new market, to attempt to strategically manage a proliferating array of products. This will require many new, but experienced, employees, and an internal labor market may not be adequate to the task. In the long run, how long can Microsoft afford to be the only game in town?

Conclusion: Microsoft vs. Silicon Valley?

According to Disclosure, Inc., in 1994 there were 333 publicly traded US companies whose primary product was packaged software; as a group, they earned over \$28 billion in gross profit that year. Microsoft, of course, was the most profitable of them all; its 1994 gross profit of \$3.89 billion gave it 14% of the US total. Eighty-nine of the 333 companies were located in

¹⁰ SIC Codes 7371 and 7373, respectively. Packaged (or "pre-packaged") software is SIC 7372.

Northern California; the great majority of these, in the San Francisco Bay Area. These 89 companies combined for a total gross profit of \$7.0 billion, or another 25% of the national total¹¹. In other words, one metropolitan region, plus one company, accounted for almost 40% of the total profit earned by public companies in packaged software in 1994, and Microsoft had over half of Northern California's total¹².

Of the companies I mentioned earlier as possible threats to Microsoft's growth, almost all of them, with the exception of the now-acquired Lotus, and the ailing Novell, are located in the Bay Area: Intuit, Oracle, Sybase, Netscape, Apple, Sun, Silicon Graphics, and Hewlett-Packard. At least a dozen other Bay Area companies, in smaller market segments, could be added to the list. Even IBM develops most of its software in San Jose, at its Santa Teresa Labs. The fact that all of these companies, in their differing specializations, are, or once were, extremely innovative and profitable is telling evidence of the reality of a spatial hierarchy in the software industry.

Yet Microsoft has been able to outsell almost all of them, in software, partly by counting on the network externalities in the segments it controls to give it time to create versions of tightly integrated products that are technically state-of-the-art, and then beginning the cycle again. However, in this paper, I have argued that the movement towards client-server networking in the computer industry is very likely to re-orient the location and degree of network externalities within the computer system. This re-orientation is the product of technological innovation in software and related industries: innovations which Microsoft, pointedly, has not yet made. It must respond to competitors' strategic innovations with strategic innovations of its own; it has acquired key technologies, and invested heavily in basic research and development. The key question is whether its traditional wait-and-see approach has been a strategic decision or a locational imposition.

¹¹ These figures come from the SEC 10-K and Annual Reports of publicly traded software companies, released by Disclosure, Inc. The Disclosure database does not include companies which have less than \$5 million in assets.

¹² Other publicly traded packaged software companies in Washington State accounted for less than 2% of the US total.

References

- Alexander, M. 1995. Another stellar year for workstations. *Datamation* (June 1): 61.
- Angel, D. P. 1991. High-technology agglomeration and the labor market - the case of Silicon Valley. *Environment and Planning A* 23(10): 1501-16.
- Bacon, G. C. 1985. Innovation in the systems business: dynamics of autonomy and cooperation. *Journal of Product Innovation Management* 2(2): 107-112.
- Caldwell, B., and S. Stahl. 1995. Gerstner's gamble. *Informationweek* (June 19): 14-20.
- Clancy, H. 1995. Drive for realtime rendering. *Computer Reseller News* (March 13): 62.
- Computer Graphics World*. 1994. Microsoft takes major step toward interactive entertainment, acquires SoftImage. 17(4): 11.
- Computer Retail Week*. 1995. Industry. (April 17): 11.
- Cortese, A. 1995. Once again, software is seething. *Business Week* (January 9): 76.
- Cox, J. 1995. Microsoft embraces the mainframe. *Network World* (November 6): 12.
- Cringely, R. X. 1992. *Accidental Empires*. Reading, MA: Addison-Wesley.
- Cusumano, M. A., and R. W. Selby. 1995. *Microsoft Secrets: How the World's Most Powerful Software Company Creates Technology, Shapes Markets, and Manages People*. New York: Free Press.
- Daly, J. 1992. Microsoft makes Fox-y move. *Computerworld* (March 30): 4.
- Economides, N. 1994. *The Economics of Networks*. New York: NYU Salomon Center Working Paper S-94-24.
- Egan, E. 1994. Spatial concentration and networking in the US software industry. Paper presented at the 1994 Association of Collegiate Schools of Planning Conference, Phoenix, AZ, November.
- Fischer, F. M., J. W. McKie, and R. B. Mancke. 1983. *IBM and the U.S. Data Processing Industry: An Economic History*. New York: Praeger Publishers.
- Flynn, M. K. 1992. Database mergers: what happens to the developers? *PC Magazine* 11(13): 29-32.
- Foley, M. J. 1995. Microsoft gains intellectual property rights from management software maker. *PC Week* (July 24): 50.
- Fortune*. 1995. The Fortune 500 largest U.S. Corporations. (May 15): F1-F22.
- Friedman, A. L., with D. S. Cornford. 1989. *Computer Systems Development: History, Organization, and Implementation*. Chichester, UK: John Wiley & Sons.
- Hawkins, J. L. 1992. Microsoft bags Fox! *Data Based Advisor* 10(5): 10-12.
- Johnston, S. J. 1995. Microsoft scrambles to find 'brightest'. *Computerworld* (February 20): 32.
- Keizer, G. 1992. How spreadsheets changed the world. *Lotus* 8(6): 28-34.
- Leibs, S., et. al. 1995. Top 50 software vendors: only the strong survive. *Informationweek* (May 22): 44-58.
- Lieber, R. B. 1995. Here comes SAP. *Fortune* (October 2): 122-4.
- Lundmark, M. 1994. The location of computer services in Sweden. Paper presented at the Association of American Geographers Annual Meeting, San Francisco, CA, April.
- Manes, S., and P. Andrews. 1993. *Gates: How Microsoft's Mogul Reinvented an Industry- And Made Himself the Richest Man in America*. New York: Touchstone Books.
- Marion, L. 1995. Tempo rubato. *PC Week* (May 22): A1-2.
- Microsoft Corporation. 1995. *Fast Facts About Microsoft*. Bellevue, WA.
- Morrissey, J. 1993. Web acquisition boosts Microsoft plan. *PC Week* (October 4): 18.

- Morrissey, J. 1994. Microsoft acquires Altamira en route to information highway. *PC Week* (September 26): 132.
- Pare, T. P. 1995. Why the banks lined up against Gates. *Fortune* (May 29): 18.
- Park, S., and L. T. Lewis. 1993. Developments in the location of selected computer-related industries in the United States. *Growth and Change* 22(2): 17-35.
- Reback, G. L. 1995. Why Microsoft must be stopped. *Upside* 7(2): 52-67.
- Rupley, S. 1995. 3-D graphics twist. *PC Magazine* (April 25): 30.
- Ryan, M. 1989. Microsoft picks up software house Bauer. *Electronic Engineering Times* (July 17): 14.
- Saxenian, A. 1994. *Regional Advantage: Culture and Competition in Silicon Valley and Route 128*. Cambridge: Harvard University Press.
- Scott, A. J. 1989. *New Industrial Spaces: Flexible Production Organization and Regional Development in North American and Western Europe*. London: Pion.
- Scott, A. J. 1993. *Technopolis: High-Technology Industry and Regional Development in Southern California*. Berkeley: University of California Press.
- Senn, A. 1995. *Open Systems for Better Business: Something Ventured, Something Gained*. New York: Van Nostrand Reinhold.
- Siwek, S. E. and H. W. Furchtgott-Roth. 1993. *International Trade in Computer Software*. Westport, CT: Quorum Books.
- Sharpe, W. F. 1969. *The Economics of Computers*. New York and London: Columbia University Press.
- Swartz, J. 1995. Power Macs fuel record apps sales. *MacWeek* (March 20): 22.
- Tanaka, D. 1994. Microsoft division shut down. *Computing Canada* (April 16): 39.
- Upside*. 1995. Why Microsoft (mostly) shouldn't be stopped. 7(4): 34-53.
- Webb, M. 1988. Who's betting on SAA? *Systems/3X World* 16(1): 28-32.
- Zachary, G. P. 1994a. Big bundles; consolidation hits the software industry. *Wall Street Journal* (March 23): A1.
- Zachary, G. Pascal. 1994b. *Show-stopper! The breakneck race to create Windows NT and the Next Generation at Microsoft*. New York: Free Press.