

Evans, David S.; Hagiu, Andrei; Schmalensee, Richard

Working Paper

A survey of the economic role of software platforms in computer-based industries

CESifo Working Paper, No. 1314

Provided in Cooperation with:

Ifo Institute – Leibniz Institute for Economic Research at the University of Munich

Suggested Citation: Evans, David S.; Hagiu, Andrei; Schmalensee, Richard (2004) : A survey of the economic role of software platforms in computer-based industries, CESifo Working Paper, No. 1314, Center for Economic Studies and ifo Institute (CESifo), Munich

This Version is available at:

<https://hdl.handle.net/10419/18679>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

A SURVEY OF THE ECONOMIC ROLE OF SOFTWARE PLATFORMS IN COMPUTER-BASED INDUSTRIES

DAVID S. EVANS
ANDREI HAGIU
RICHARD SCHMALENSEE

CESIFO WORKING PAPER NO. 1314
CATEGORY 9: INDUSTRIAL ORGANISATION
OCTOBER 2004

PRESENTED AT CESIFO ECONOMIC STUDIES CONFERENCE ON
UNDERSTANDING THE DIGITAL ECONOMY: FACTS AND THEORY, JULY 2004

An electronic version of the paper may be downloaded

- *from the SSRN website:* www.SSRN.com
- *from the CESifo website:* www.CESifo.de

A SURVEY OF THE ECONOMIC ROLE OF SOFTWARE PLATFORMS IN COMPUTER-BASED INDUSTRIES

Abstract

Software platforms are a critical component of the computer systems underpinning leading-edge products ranging from third-generation mobile phones to video games. After describing some key economic features of computer systems and software platforms, the paper presents case studies of personal computers, video games, personal digital assistants, smart mobile phones, and digital content devices. It then compares several economic aspects of these businesses including their industry evolution, pricing structures, and degrees of integration.

JEL Code: L11, L15, L86.

Keywords: software platforms, hardware platforms, network effects, bundling, multi-sided markets.

David S. Evans
LECG LLC
350 Massachusetts Avenue
Cambridge, MA 02139
USA
devans@lecg.com

Andrei Hagiu
Research Institute for the Ministry
of Economy Trade and Industry
1-3-1 Kasumigaseki Chiyoda-ku
Tokyo 100-8901
Japan
[*ahagiu@princeton.edu*](mailto:ahagiu@princeton.edu)

Richard Schmalensee
Sloan School of Management
Massachusetts Institute of Technology
50 Memorial Drive
Cambridge, MA 02142
USA
rschmal@mit.edu

An earlier version of this paper was presented at “Understanding the Digital Economy: Facts and Theory,” in Munich on 2 July 2004. We appreciate very helpful comments received from Pierre Regibeau, our discussant, and other participants. We would like to thank Brendan Reddy and Ori Stitelman for excellent research assistance. Evans and Schmalensee are grateful for research support from Microsoft Corporation; Hagiu would like to thank the RIETI for its hospitality and research support while working on this project.

1 INTRODUCTION

i-mode is a mobile information service offered by DoCoMo in Japan. Based on a mobile telephone handset, it enables users to access content from some 82,000 Web sites and to run a wide variety of software applications on DoCoMo's mobile telephone network. iPod is a digital music player sold by Apple. It is designed to play music downloaded from Apple's iTunes service using Apple's QuickTime's media player.

i-mode and the iPod are two examples of economically important products based in part on computer systems. Every component of computer systems plays a critical role in these businesses. The characteristics of the software platform, however, can be the key to the viability of a computer system, facilitating the creation of software applications and increasing the value to consumers of the integrated system. Moreover, the degree to which software platforms are integrated with other components affects many aspects of the organization and evolution of computer systems.

Some software platforms are tightly integrated with the hardware platform—for example, Apple Macintosh for personal computers and Sony PlayStation for video games. Others are not—for example Windows for personal computers and Symbian for mobile phones¹—and, as a result, have facilitated competition among suppliers of the hardware platform. Integration between the software platform and software applications has also varied. In the early development of video games, manufacturers made the game platform as well as the games; today most platform vendors license their platforms to independent game developers who generate the bulk of applications. Here, we analyze the business dynamics of industries based on computer systems, with special attention to the software platform.

Computer systems are often organized as “multi-sided platform markets.”² A business is based on a multi-sided platform if it serves two or more distinct types of customers who depend on each other in some important way, and whose joint participation makes the platform valuable to each.³ Windows is an example of a three-

¹ Symbian, however, is made by a joint venture of mobile phone companies.

² Rochet and Tirole (2003). Although economists working in this area generally use the term “two-sided” and most of their models are indeed two-sided, it is clear that the insights carry over to multi-sided markets. Since most of the markets we study in this paper have more than two sides, we will use the term “multi-sided” whenever it is appropriate.

³ Rochet and Tirole (2004) use a more formal version of this requirement: for them, a necessary and sufficient condition for a market to be two-sided is that the volume of transactions be sensitive to the distribution of total costs between the two sides. This implies that a platform can improve upon the market outcome through a pricing structure that rebalances costs between the two sides by internalizing (to a

sided platform. The platform provides services to application developers, hardware manufacturers, and end-users. Each group needs the others to participate if the platform is to generate market value. The emerging economics literature on multi-sided platform markets provides interesting observations into pricing and business strategies, offering insights into the workings of industries based on computer systems.

It is worth noting, though, that some markets organized around multi-sided platforms could be viable as traditional one-sided markets.⁴ Firms can choose to integrate vertically into the supply of a component rather than relying on the market. For example, Apple's iPod could, in principle, be operated as a multi-sided market platform, but Apple has decided to build its business model around a fully integrated single-sided product. Apple produces both the hardware and software, and operates its own content-provision service for which it buys the content in competition with others. Some of its competitors may, however, find that it pays to take a multi-sided approach, creating portals through which consumers buy music or other content from third-parties.

Our survey is organized around case studies of five computer-system-based industries: personal computers, video games, personal digital assistants, smart mobile phones, and digital content devices. In section 2, we offer some principles for analyzing business models in these industries. In section 3, we describe the organization and evolution of each of the industries. In section 4, we draw some broad conclusions from the case studies, leaving readers with some puzzles to solve.

2 ECONOMIC AND TECHNICAL BACKGROUND

A number of forces are driving the proliferation of industries that rely on computer systems. Microprocessors have become cheaper, smaller, and more powerful, while memory devices have become much less expensive.⁵ Today, many products come with built-in computer systems complete with microprocessors and operating systems, which in some cases are more powerful than the PCs of less than a decade ago.⁶ Often,

certain extent) the indirect externalities. This condition implies, but is broader than, the failure of the Coase theorem.

⁴ Contrast this with, for example, the payment card industry, where it is simply not plausible to imagine the integration of the two sides of the business—payers and payees—into a traditional, single-sided market.

⁵ Davis et al. (2001).

⁶ Typical Pocket PCs available today have processor speeds between 266 and 400 MHz. It was not until 1997 that Intel released a chip with a clock speed of 233 MHz and not until 1999 that they released a chip of 450 MHz. Note, these numbers are not directly comparable due to the differing architectures of the systems.

these computer systems are integrated with the product and are designed to perform a specific set of tasks.⁷

In the markets in which we are interested, the computer system is the core of the product and supports a multi-sided market by becoming a focal point for the development of complementary products. Smart cards with built-in chips, like the American Express Blue card, are examples. Payment card systems have designed these cards to encourage the development of applications that can run on them.

Industries based mainly on computer systems are quite diverse. Nonetheless, they share economic and technical features that we summarize in the remainder of this section.

2.1 Some Technical Observations

The microprocessor is at the core of a modern computer system.⁸ PCs center around a microprocessor, a central processing unit that carries out instructions, performs computations, and issues commands to the peripheral devices connected to the system. Microprocessors control the logic of digital devices ranging from climate control systems in office buildings to handheld global-positioning appliances. The operating system is equally important to the working of the system. It consists of software code that instructs the microprocessor to perform specified tasks, notably controlling and exchanging information with hardware. The operating system also enables software developers to write applications that use the operating system instructions to interact with the microprocessor rather than directly interacting with it themselves.

Memory is another important category. A variety of mechanical and solid-state devices can store anything that can be reduced to digital information, and then make this information available to the microprocessor and operating system.

Software technology has evolved to facilitate the writing of new code for both applications programs and operating systems. This technology consists of “high-level” languages such as C++. Programs written in these languages can then be “compiled” into binary code that gives the microprocessor the digital 0/1 instructions it understands. An important source of innovation in computer systems involves developing languages and tools that help programmers write applications efficiently.

⁷ The computer systems embedded in cars, often called engine controllers, provide a familiar example. These special-purpose systems are interesting in their own right, but do not raise the sort of business or economic issues on which we focus here.

⁸ It was not until the early 1970s that microprocessors were invented. Early computers relied on various different technologies: in the 1940s, mechanical relays, in the 1950s, vacuum tubes, and in the 1960s, memory chips that included hundreds of transistors. Frieberger and Swaine (2000).

Software platforms contain modules of code that perform various tasks.⁹ These modules provide services to be used by other parts of the operating system or by software applications. Access to these services is provided through applications program interfaces (APIs).

APIs are conceptually similar to a statistical command like `regress(X,Y)`, where the underlying code performs the specified algebraic manipulation on (X,Y) and returns results. Like the user of a statistical command, the user of an API does not typically see the code underlying the command.

In some cases, application developers must use APIs to work on a hardware platform with a particular microprocessor. In most cases, though, application developers use APIs to avoid the tedious process of writing code to perform routine functions. Software platforms also serve as interfaces for hardware manufacturers, allowing them to finesse the problem of writing code specific to the myriad potential permutations in the way systems are configured.¹⁰

Most operating systems facilitate writing and running applications on the underlying hardware, and therefore function as software platforms. However, software platforms do not always provide basic, hardware-related operating system services. When they do not, they are called “middleware” because they logically sit between the operating system and the applications. For example, Java Virtual Machines and the Java class libraries provide a platform for applications to run on a variety of hardware and operating systems.

Two other observations are worth making here. First, the development of many of the key components of computer systems requires large investments in research and development, and embodies significant intellectual capital. An important source of protection for software comes from the fact that the code is typically released only in

⁹ These modules may be in separate files, or many modules may be “linked” together in a single file for distribution.

¹⁰ For example, some hardware, including some external DVD drives like the MCE “Superdrive” that connects to a computer through an industry-standard interface, is designed only to work with a single software platform. This is the case with the Mac; the MacOS shields the drive from the computer’s graphics and sound cards, so that a user plugging the drive into a Mac computer need not worry that the drive will not function. “MCE Desktop Series External 8x DVD±R/RW FireWire/USB 2.0 ‘SuperDrive’,” <http://www.mcetech.com/sd8fwusb2-d.html> (downloaded 5 July 2004). Other peripherals, such as the CanoScan 8000F scanner, use an industry-standard interface to allow the user to connect it to multiple platforms, for which Canon has written code to permit such connections. “CanoScan 800F,” <http://consumer.usa.canon.com/ir/controller?act=ModelDetailAct&fcateoryid=120&modelid=8054> (downloaded 5 July 2004).

binary form, which can only be read by microprocessors. The underlying source code, written in a higher-level language, is kept secret in order to prevent the theft of ideas.¹¹

Second, there are extensive scale economies in creating some of the key components. That is true for microprocessors, where the substantial fixed costs include both the construction of specialized plants for fabricating chips and the R&D embodied in the design and testing of the chip. The same holds true with a vengeance for software, where virtually all the costs come in writing, testing, and debugging the code. What little incremental cost there is consists almost entirely of outlays for marketing and distribution; the cost of physically reproducing copies of the software is minimal.¹²

2.2 Bundling Components

Computer systems are comprised of many components. Firms combine these components to create products that are ultimately sold or licensed to end-users. The contours of these products—what is included or excluded from the bundle of components—are determined by business and design decisions.

The microprocessor, memory, and other components are typically combined to create a hardware platform such as a PalmOne Zire PDA, a Nokia mobile phone handset, or an Xbox game console. Hardware components that are not included in the physical product are termed peripherals and include items such as headphones, pointing devices, and printers. Sometimes one model of a product will integrate a specific component, while another will not—for example, a pointing device is virtually always integrated into a laptop computer but rarely into desktop models.

With time, many peripherals come to be integrated into the hardware platform. Consider the case of the math co-processor, which facilitates number-crunching. Prior to the release of Intel's 486 chip, Intel's microprocessors did not include a math co-processor; customers who wanted one purchased it separately from one of several vendors at substantial cost. Today, one cannot buy an Intel x86 processor without a built-in math co-processor.¹³

Similar economic and technological forces are at work in software. Operating systems, software platforms, and applications all consist of code that accomplishes specific tasks. Where those tasks are accomplished is a matter of business and design decisions. And those decisions may vary among competing products as well as between

¹¹ In addition to trade secrets, software is protected by copyright. Some software code can also be patented.

¹² Campbell-Kelly (2003, pp. 209–212). See also Evans and Schmalensee (2001).

¹³ Meuller, Scott and Mark E., "Microprocessor Types and Specifications," *informIT*, 8 June 2001.

generations of the same product. Many tasks that used to be performed by stand-alone applications have become integrated into other applications (e.g., spell checkers, which were originally sold separately from word processors) or into the software platform itself. Early operating systems, for example, did not include built-in calculators.

A variety of economic considerations go into the decisions on what to include in a product:

- (i) *Integration and coordination benefits.* The value of the whole may exceed the combined value of the parts. Thus businesses may be able to make more competitive products by integrating components. Most hardware platforms sold to end-users are, in fact, assembled from components purchased from a variety of vendors.
- (ii) *Economies of scope.* It may be more efficient to produce and distribute components as a group rather than separately. The choice will depend on the nature of the demand for the components as well as the extent of the economies. Evans and Salinger show, for example, that when there are fixed costs in offering separate products, firms may find it profitable to bundle those products if demand for the separate components is lacking.¹⁴
- (iii) *Demand aggregation.* When there are fixed costs of producing and distributing products but low marginal costs in adding components, it may be possible to lower costs and reduce variation in reservation prices by combining components that appeal to different groups of customers.¹⁵ Hardware and software typically include many features that most consumers never use. However, by including these features vendors expand the number of consumers who find the product valuable at the offered price.

These considerations are not unique to computer systems. They apply to everything from newspapers (a bundle of news, arts, lifestyle, and sports content), to cereals (consider bran flakes plus raisins), to cars (few come these days without air conditioners and rear window defrosters).¹⁶

¹⁴ Evans and Salinger (2004).

¹⁵ Bakos and Brynjolfsson (1999). Reducing variation in reservation prices enables the seller to capture more of the product's value.

¹⁶ Evans and Salinger (2004b).

2.3 Complementarities and Network Effects

System components are generally complements. Moreover, in many cases, systems are characterized by indirect network effects linked to the presence of components.¹⁷ That is, an increase in the number of users for one component often makes that component more valuable as a complement to the other components.

For example, as Sony's Internet-based game center for the Playstation 2 draws more users, more Playstation 2 owners will want to buy games supported by the Internet service and more consumers will prefer Playstation 2 consoles to competitors' models. Likewise an increase in the variety of components (printers for personal computers, for example) often increases the value of other components to end-users.

There may also be direct network effects. An increase in the number of users for a platform makes the platform more valuable to each user—typically because a common platform enables people to work together and share information. The “Palm Economy” provides another interesting example. The popularity of the PalmOne PDAs attracted many developers of small applications and hardware add-ons. There are currently thousands of Palm-related products—hence the term Palm Economy—which, of course, makes the device more valuable to users.¹⁸

2.4 Multi-Sided Market Considerations

The literature on multi-sided markets considers situations in which a platform can internalize externalities present among several distinct customer groups.¹⁹ The primary means for accomplishing this is to set prices in a way that balances the demand of the multiple customer groups (i.e., multiple sides of the market) in order to maximize the value of the platform. A key conclusion of this literature is that, because the price structure serves to internalize externalities, the price levels for each side do not track traditional price–marginal cost relationships.²⁰ Instead, optimal pricing turns on the demand structures and marginal costs of the several sides. Prices on one side may be less than marginal cost—even less than zero.²¹

¹⁷ Katz and Shapiro (1994).

¹⁸ See, for example, “Palmtops/PDAs—PDA Accessories,” <http://palmtops.about.com/cs/palmaccessories/> (downloaded 16 September 2004).

¹⁹ Rochet and Tirole (2004).

²⁰ Rochet and Tirole (2003); Armstrong (2002).

²¹ Evans (2003).

Four types of multi-sided market platform businesses have been identified in the literature.²² *Match-makers* aid members of one or both sides in their quest for a match on the other side (think of real-estate brokers, B2B Web sites, and stock exchanges).²³ *Audience-makers* bring advertisers and audiences together, matching buyers and sellers (yellow page directories, television, newspapers, magazines, and Internet portals).²⁴ *Transaction-based businesses* meter transactions between the two sides of a market (monetary media such as credit cards, advanced-generation mobile phone networks).²⁵

The fourth type, *shared-input platforms*, include the hardware and software platforms discussed in this paper. Participants on at least one side need to obtain access to the one side to provide value to participants on at least one other side. Thus, software developers need to obtain access to the APIs and other features of the software platform in order to write applications for end-users. Likewise, software platform designers and hardware peripheral manufacturers need access to the functionality of the hardware platform to develop products for that platform.²⁶

Many of the firms in the computer-based industries examined here are organized as multi-sided platforms. However, unlike some other cases, the decision to organize computer-based businesses this way seems to be dictated by the same considerations that determine whether a firm follows a vertically-integrated or disintegrated business model. For example, advertising-supported media cannot vertically integrate into advertising, exchanges cannot realistically integrate vertically into the sell-side, and transaction-based platforms cannot also become sellers. Nor is there a way of providing a general-purpose payment card without a two-sided model.

But there is a way of running a computer-system based business in which the firm supplies, and perhaps bundles together, all components. In fact, many computer-based businesses have adopted this vertically integrated model instead of creating multi-sided markets. “Multi-sidedness” is therefore considerably more dependent on the particulars of technology and market environment in the computer-based industries than it is in many other platform-based businesses.

²² Evans (2003b); Hagiu (2004).

²³ Caillaud and Jullien (2003).

²⁴ Baye and Morgan (2001) propose a formal model of Internet monopoly “gatekeepers” connecting advertisers and consumers.

²⁵ Rochet and Tirole (2003).

²⁶ Hagiu (2004).

3 COMPUTER–SYSTEM BASED INDUSTRIES

This section surveys five diverse, computer–based industries. To avoid getting lost in a mass of detail, we focus on two issues related to the degree to which firms have chosen to organize themselves as multi–sided platforms. The first of these is the structural dynamics of the industry, including how the degree of vertical integration varies over time. The second is pricing strategies and their use in exploiting network effects. A particular focus here is on businesses that have adopted multi–sided market models and on how they have used pricing to “get all sides on board” and to maximize the value of their systems. We devote the greatest attention to the computer industry itself, and then compare its development with other industries based on computer systems.

3.1 Computers

Several sorts of computers have emerged since the first commercial machines were introduced in 1951. It is useful to distinguish mainframes, minicomputers, personal computers, workstations, and server computers. These sectors have followed diverse dynamics and employed different business models over time. We begin with an overview of the hardware and software platform relationships, and then focus on personal computers.

3.1.1 Overview

The vendors of many sorts of computer systems have chosen to provide the hardware and software platform together.²⁷ Most computer mainframe and minicomputer companies have produced their own operating systems, which are optimized for the computer hardware, and have bundled these operating systems with the hardware platform. In these circumstances, software applications can only run on that specific hardware and software platform combination.²⁸

Some thought that the Unix operating system, introduced by Bell Labs, would provide an operating system that worked across different hardware platforms. However,

²⁷ There are other dimensions to decisions on vertical integration not discussed here. For example, one of the major differences between personal computers and mainframes is the division between manufacturers of microprocessors and the manufacturers of the hardware platforms that are sold to end users.

²⁸ IBM, for example, developed several operating systems for the computers it introduced in the 1950s and 1960s beginning with the GM–NAA I/O System for its 704 mainframe in 1956. It developed OS/360 for its System/360, which was the leading mainframe–computing platform for many years after its introduction in the late 1960s. Recently, IBM has attempted to standardize its various mainframes on the open source Linux OS.

the computer manufacturers using Unix found that there were advantages to modifying the OS to optimize its performance with the underlying hardware.²⁹

When personal computers were first introduced in the late 1970s, producers generally followed the traditional vertical integration model: firms provided both the hardware and software platforms. That is how Apple did it—not to mention Tandy, Commodore, Texas Instruments, Coleco, Atari, Timex, and Sinclair.³⁰ Some vendors relied on the CP/M operating system, licensed from Digital Research. But CP/M had to be modified for each hardware platform, and applications that ran on one CP/M system generally did not work on others.

When IBM entered the industry in 1981, the company planned to make it possible for its hardware platform to work with several operating systems. These included PC-DOS developed by and licensed from Microsoft, CP/M licensed from Digital Research, and the UCSD p-System licensed from Softech.³¹ As discussed below, the personal computer marketplace soon became a battle between the Apple and IBM platforms.

Beginning in the 1980s, several companies including Sun and Hewlett Packard produced workstations that were designed to perform more processing-intensive tasks than most personal computers. These workstations had operating systems based on the UNIX or OpenVMS operating systems, but were tightly coupled with a specific hardware platform. Vendors typically bundled their hardware and software platforms, much like Apple has done in PCs. Microsoft introduced a version of Windows for workstations in the early 1990s that ran on the increasingly powerful Intel platform.

More recently, the development of networked personal computers led to the creation of an industry to provide “server computers,” which performed various specialized tasks on the network.

Two business models have coexisted in the workstation and server industries for a number of years. Some manufacturers, including Sun, provide integrated hardware and software platforms. Several others, notably Microsoft and Novell, license software

²⁹ Campbell-Kelly (2003, pp. 143–145); See also Salus (1995).

³⁰ Campbell-Kelly (2003, p. 224).

³¹ When IBM released its personal computer, MS-DOS, re-branded PC-DOS when purchased through IBM, was the only complete, fast-running OS available for its hardware. The UCSD p-System was available but ran very slowly. CP/M-86 did not become available until 1982, and had the added drawback of a hefty price tag—\$240, compared to DOS at just \$40 when introduced in 1981. Manes and Andrews (1994, p. 192).

platforms that can be used with Intel-compatible hardware platforms manufactured by others.

3.1.2 Personal Computers

The choice of business models appears to have had a dramatic effect on the evolution of the personal computer industry. Apple bundled its proprietary hardware platform with its proprietary software platform. IBM did not. The company offered several compatible operating systems developed by third-party vendors and permitted vendors to license these operating systems to other hardware vendors. IBM tried to thwart the cloning of its hardware platform, but it did not succeed.³² Microsoft widely licensed the operating system it had developed for the IBM PC, which ultimately led to intense competition among purveyors of the hardware platform. Despite some experiments with licensing its operating system, Apple continued to follow an integrated strategy.³³

With the benefit of hindsight, one tends to think that Apple's strategy was foolish and Microsoft's strategy was brilliant. At the time, however, both strategies had advantages and disadvantages that left it unclear which strategy would prevail. Apple could coordinate the hardware and software platform, realize possible benefits from integrating them tightly, and ensure the quality and reliability of the combined system. Microsoft, for its part, could benefit from hardware competition and perhaps from specializing in what it knew best—writing software. But the company could not test any single hardware/operating system combination as intensively as Apple could, and it could thus not do as much to ensure overall system quality and reliability.

The Apple-Microsoft competition is well-trod territory.³⁴ We consider it here only from the perspective of multi-sided markets theory. Microsoft followed a four-sided strategy in which it tried to bring hardware platform manufacturers, application developers, peripheral manufacturers, and end-users on board its platform. Apple initially followed a two-sided strategy in which it marketed to end-users, integrated into

³² The IBM decision in the eighties to make the XT PC specifications public opened up the PC market to everybody. Total personal computer sales exploded, but IBM lost share. In 1986, IBM tried to reverse the trend by stopping the production of the AT PC and introducing the new Microchannel architecture (MCA). Unlike the previous PC specifications, MCA was proprietary and IBM demanded substantial license fees for its incorporation in hardware. IBM hoped that its position as a market leader would be enough to turn the market in its direction. Vendors, however, largely ignored the MCA and continued to develop the old architecture. By the end of the eighties, when IBM quietly abandoned the MCA technology, Compaq was the largest maker of PCs. Carrol (1994, pp. 120–168).

³³ From 1994 to 1997, Apple licensed its operating system to manufacturers of Apple clones. It stopped because the licensing revenues gains did not begin to cover the engineering costs.

³⁴ See, for example, Carlton (1998); Manes and Andrews (1994); Liebowitz and Margolis (2001); Stross (1996); and Malone (1998).

the hardware platform and peripherals, and tried (though with less gusto than Microsoft) to enlist application developers as partners. Apple's strategy subsequently became three-sided as it loosened its grip on printers and other peripherals.

We begin by examining how Microsoft and Apple dealt with the applications side, and then turn to how Microsoft has dealt with computer manufacturers.

3.1.2.1 Software Applications

From the beginning, the fortunes of the Apple and IBM platforms turned on their ability to win over independent applications developers as well as end-users. Both platforms took off in part because of "killer apps"—novel applications that enable a task that had previously been impossible or too expensive, and thereby drive the adoption of a new technology. Apple sales for personal and business use increased rapidly with the introduction of VisiCalc, the first commercial spreadsheet program, which was originally available exclusively for the Apple II. Likewise, IBM PC sales took off once the Lotus 1-2-3 spreadsheet application was released.

We have found no evidence that these platform providers (Apple, IBM, and Microsoft) actively encouraged the creation of the killer apps. Moreover, it appears the software platforms at the time played much less important roles in providing services to applications than they did later in the history of the personal computer business. VisiCalc for the Apple II and Lotus 1-2-3 for the IBM PC sometimes even found it more efficient to use its own code rather than code embodied in the operating system to perform routine tasks. But in later years, Microsoft and Apple realized that, with so much riding on the quality of applications available, it made sense to make it as attractive as possible to write software for their platforms.

Most importantly, both expanded the set of APIs providing services to developers. This was especially apparent at Microsoft as it moved its software platform from DOS to Windows. According to Bill Gates, Windows XP exposes more than 6,000 APIs for developers through its Win 32 API set.³⁵ Apple has also created an extensive set of APIs for the convenience of its developers. Indeed, it has split these APIs into Cocoa, a set of high-level functions that developers can use to write complex applications using very little code, and Carbon, a class of fine-grain procedural APIs.³⁶

³⁵ Direct Testimony of Bill Gates, 18 April 2002 in *State of New York v. Microsoft*, Civil Action No. 98-1233 (CKK).

³⁶ Apple's Developer Connection Web site, "An Introduction to Mac OS X Development Technologies," 1 June 2004, <http://developer.apple.com/macosx/introdevtech.html> (downloaded 3 September 2004).

Both launched programs for “evangelizing” their respective software platforms among developers, with Apple taking the lead in the early 1980s.³⁷ According to Guy Kawasaki, the greatest Apple software evangelist:

The secondary effects of getting people to believe, who then got more people to believe, is something that was stumbled upon. In my recollection, I was never told, ‘OK, you go get XYZ to write software, and they in turn will get more customers to buy your software and to buy Macs. That’s what happened, but that was not the plan.’³⁸

In 1984, shortly after Apple’s introduction of the evangelist role, Microsoft formed the Developer Relations Group (DRG) to drum up support from third-party application developers for the first release of Windows.³⁹ DRG’s mission was to “Drive the success of Microsoft’s strategic platforms by creating a critical mass of third-party applications.” By 1995, DRG had over 85 employees in the United States and many more in the rest of the world.⁴⁰ Their evangelism process is well defined, and is broken out into four steps that focus on choosing which technologies to evangelize and working one-on-one to support early adopters.⁴¹

Both platforms now provide benefits to software developers at little cost. In fact, both platforms have developer networks that offer online assistance at no cost through their developer Web sites. Members of these networks receive newsletters, documentation, software seeds, system software, and software development kits.⁴²

According to a 1995 Microsoft white paper, the company has long been committed to supporting third-party development:

Microsoft has its roots in developer support. The company was founded to develop languages and tools for PC application developers. As developers themselves, Microsoft’s early employees understood the difficulties of supporting product on the multiplicity of operating systems that existed in the late 1970 and early 1980s. One of their key interests in MS-DOS was to try to create a broadly popular platform that would greatly simplify their own development process. In recognizing that, the company also realized

³⁷ McConnell and Huba (2003).

³⁸ McConnell and Huba (2003).

³⁹ Microsoft (1998).

⁴⁰ Microsoft (1998).

⁴¹ Microsoft (1998).

⁴² Microsoft’s Microsoft Developer Network (MSDN) began in 1997. Apple’s Developer Connection (ADC) began in 1998.

that it needed to marshal other developers and hardware vendors to create a critical mass of support for MS-DOS.⁴³

As of 1995, Microsoft was spending about \$65 million annually to support software developers, and had about 400 technical support engineers on staff. By 1998, Microsoft was spending over \$250 million annually on developer-related activities that included research, events, marketing assistance, training programs, product support, and publications.⁴⁴ In January 1999, Paul Maritz, the Group Vice President of Platforms and Applications at Microsoft, testified that Microsoft was planning to spend \$630 million to help software developers write applications, and that 2,000 of Microsoft's 27,000 employees work full time in helping developers.⁴⁵

It is obvious that Apple and Microsoft would benefit from stimulating the production of applications. It is less obvious that the profit-maximizing strategy involved giving away developer services. The application developers, after all, were getting something of value from the software platforms: by writing for these platforms, they could market their applications to businesses and individuals using Apple or IBM-compatible computers. By 1990, Microsoft had 71 million users worldwide and Apple had 5 million users worldwide.⁴⁶ Therefore, both Apple and Microsoft could, in theory, have charged for access to their APIs and for the other platform services for developers, once they had established a substantial user base. And, in fact, that is just what makers of video game platforms did. (See Section 3.2 below).

In addition to persuading third parties to write applications, both Apple and Microsoft created applications for their own software platforms.⁴⁷ As of 1999, Microsoft applications represented about 20 percent of the sales of applications for its own platform, while Apple had a 10 percent share of applications for its platform.⁴⁸

⁴³ Microsoft (1995).

⁴⁴ Microsoft (1998); and Microsoft (1995).

⁴⁵ Direct Testimony of Paul Maritz, 20 January 1999 in *US v. Microsoft and State of New York, et al. v. Microsoft*, Civil Action Nos. 98-1232 and 98-1233 (TPJ), http://www.courtTV.com/archive/trials/microsoft/legaldocs/maritz_full.html (downloaded 2 September 2004).

⁴⁶ IDC (1993).

⁴⁷ This is not surprising for Microsoft since a major part of its business was writing applications. It supplied some of the biggest selling applications for the Mac—Word and Excel—well before these applications became successful on Windows.

⁴⁸ IDC (2000).

3.1.2.2 Hardware

Apple and Microsoft differed in how they treated the hardware platform. Apple began as an integrated hardware and platform company. Its first computer, the Apple I, was introduced in 1976 and came with a firmware resident system monitor that allowed the user to display, write, debug, and run programs. Except for a brief flirtation with licensing clone manufacturers in the mid 1990s, Apple has always bundled its hardware and software platforms.

Bill Gates did send a now-famous letter to Apple's CEO John Sculley and Apple Products President Jean-Louis Gassée in 1985 in which he argued that Apple should license its operating system software to other manufacturers. Gates advised Apple to consider licensing the MacOS in order to "create a standard," stating that the "significant investment (especially independent support) in a 'standard personal computer' results in incredible momentum for its architecture."⁴⁹

Gates' counsel was not altruistic. Microsoft had a financial interest in the success of Apple because, as noted above, it was a major supplier of applications for the Apple platform.⁵⁰ But many observers argue that Apple's failure to follow his advice contributed to Apple's loss of share to the open Microsoft platform. It is hard to know for sure. After all, much of the appeal of Apple computers is the seamless integration of its hardware and software, and a three-sided model might not have done as well.

Microsoft started as a software company focused on programming languages and only began working with operating systems when it was commissioned by IBM to do so.⁵¹ In its agreement with IBM, Microsoft reserved the right to license the operating system to other companies. And soon after the IBM PC had established itself as a serious contender in the emerging personal computer industry, Microsoft started working with other computer manufacturers who were developing IBM clones. It is generally accepted that the ensuing competition with IBM helped reduce the price of IBM-compatible PCs, thus increasing total PC output and expanding the demand for Microsoft's operating systems.

In some respects, Microsoft is the tail that wags the proverbial PC dog. Computer manufacturers install copies of MS-DOS (and later Windows) on their computers before shipping them to end-users, and end-users are made aware that their PCs have Microsoft

⁴⁹ Linzmayer (1999).

⁵⁰ At this time Microsoft's future success in software platforms was far from ensured and one can view Gates's plea as an effort to hedge Microsoft's bets on its application and software business.

⁵¹ Carrol (1994, pp. 17–25).

operating systems installed.⁵² Manufacturers pay license fees to Microsoft based on the number of personal computers they ship with Microsoft operating systems installed. As in any manufacturer–distributor relationship, these license fees are based largely on the ultimate demand by end–users for Microsoft’s operating systems; for the computer manufacturers, Microsoft’s operating systems are a cost that is passed on in whole or in part to end–users.

In fact, the relationship between Microsoft and the computer manufacturers has been more complicated than the preceding paragraph suggests. Microsoft, Intel, and the major computer vendors have worked with each other (and often with peripheral manufacturers) to prepare new products; in the early days, in particular, the process was driven by advances in microprocessors. Microsoft worked with the computer manufacturers both to make sure that its operating system worked well with the next generation of hardware and to ensure that the next generation of hardware made use of features that Microsoft was developing for its operating system. Over time, Microsoft has created pricing incentives for the inclusion of hardware features that made Microsoft’s operating systems more useful.

Microsoft provided services to computer manufacturers and makers of peripherals in several ways. In the early days of DOS, any developer that wanted its application to work with, say, a large set of printers already in use had to customize its application for each printer. If there are N applications and M peripherals in use, in order for every program to interact with every peripheral, $N \times M$ drivers would have to be written.

With the transition to Windows, however, application programs needed only include a single general Windows driver for each type of peripheral (printer, scanner, etc.) and device manufacturers needed only write Windows drivers to make use of their device’s features; the operating system linked them together.⁵³ Thus, thanks to Windows, in order for every program to communicate with every device, only $N + M$ drivers had to be written. Microsoft continued to put a lot of effort into making it easier for manufacturers of peripheral equipment to write for their operating system. For example,

⁵² This is important. For the end–user, the operating system is a visible, distinct, and important element of the overall system, an element with which they interact directly. Disk drives have also been important over time, but they have generally been viewed as simply commodity inputs used to build the hardware platform. One can view Intel’s “Intel Inside” marketing campaign as an attempt to transform its microprocessors from inputs that are used by manufacturers to products that manufacturers distribute to consumers.

⁵³ “Review Microsoft Windows Software Development Kit,” *Byte*, 1 June 1987.

when Windows 95 was released, the platform supported over 800 different printers right out of the box.⁵⁴

Another example of Microsoft's contribution to the development of hardware peripherals is the company's collaboration in the development of the CD-ROM. In 1986, prior to even having a CD-ROM product in the works, Gates insisted that Microsoft associate itself with CD technology by holding a CD-ROM conference.⁵⁵ The conference was a success and Gates decided to make it an annual event. Gates recognized that to exploit the technology fully, Microsoft would need to acquire new skills, and he was not convinced that a viable business was possible in this arena. In fact, Gates told a colleague that he was willing to invest \$200 million to find out if a profitable market existed for the CD technology.⁵⁶

To jump-start the industry, Microsoft worked with representatives from Sony, Phillips, Digital Research, Apple, and others to accelerate the establishment of a single file format.⁵⁷ Microsoft also evangelized the "religion of CD-ROM" by convincing hardware companies to offer personal computers with pre-installed and configured CD-ROM drives in addition to add-on external drives.⁵⁸

Apple has followed a very different path with respect to peripherals. Because the IBM PC is an open platform with no single vendor controlling the standard, anybody can produce peripherals to work with any PC. By contrast, Apple exercised significant control over its system. The company manufactured and sold its own peripherals—opening the platform to others, the thinking went, would cut into Apple revenues. As a result, Apple was usually late to include industry-standard ports and mostly relied on its own peripherals.⁵⁹ For example, the market for third-party Mac monitors did not exist until 1987. Even today, many Macintosh computers house the monitor and the computer core in a single unit.⁶⁰ Note, too, that the very first Mac was designed to make expansion as hard as possible. It included a bar above its few ports in order to make expansion

⁵⁴ Mace (1995).

⁵⁵ Manes (1994, p. 336).

⁵⁶ Stross (1996, p. 65).

⁵⁷ Stross (1996, p. 65).

⁵⁸ Stross (1996, pp. 65–66).

⁵⁹ IDC (1988, p.2).

⁶⁰ Apple just released the iMac G5, which contains many peripherals within its two-inch thick display. Apple advertises that "The entire computer, including a G5-based logic board, slot-loading optical drive, hard disk, speakers, and even the power supply—dwells inside the enchanting display." "Where did the computer go? The all-new iMac G5," <http://www.apple.com/imac/> (downloaded 10 September 2004).

difficult. What's more, the cover was difficult to remove—the better to discourage users from opening it.⁶¹

Even after Apple opened the Macintosh to third-party peripheral manufacturers, it retained a significant advantage for its own equipment through the tight integration of the Macintosh with Apple-made peripherals. For example, at the end of the 1980s, Apple remained virtually the only supplier of printers for the Macintosh platform; not surprisingly, 95 percent of the printers attached to Macintoshes were made by Apple.⁶²

3.1.2.3 Pricing Structures

Multi-sided market theory is built on the premise that pricing is a critical element in any strategy to maximize platform value. Microsoft has earned virtually all of its revenue from end-users—either directly through the retail channel, or indirectly through computer manufacturers who pass on the licensing charges to end-users. It has given away access to its APIs as well as to other valuable services for writing applications, and thus earned only minor revenue from licensing software tools. It has also given away valuable services to computer manufacturers and hardware peripheral manufacturers.⁶³ Apple has adopted a similar pricing structure, though as noted before, it has vertically integrated the production of hardware.

3.2 Video Games

The video game industry has been remarkably dynamic since its birth, starting with Pong in 1975. Comparing its evolution with that of computers provides useful insights concerning choices of structure and business models.

3.2.1 Evolution of business models

Video game consoles were introduced in the 1970s as devices that played a single game that was hardwired into the console's circuitry. The most successful ones were versions of already popular Arcade games, such as Atari's Home Pong (1975) and Coleco's Telestar (1976). With the release of Fairchild's Channel F game console in 1976, a new business model emerged. Channel F did not hardwire games, but rather played games stored in interchangeable cartridges. Atari expanded on Fairchild's approach with

⁶¹ Kunkel (1997).

⁶² IDC (1988, pp. 16–17).

⁶³ It is important to distinguish here between OEMs as Windows licensees and OEMs as the third side of the market. In the computer market, *it just happens* that OEMs, in addition to building the hardware, are also the major distribution channel for Windows. The payments from OEMs to Microsoft are exclusively for the Windows license and are entirely passed on to end users; they do not reflect the services that Microsoft offers OEMs in order to help them build the hardware.

the release of the Video Computer System (VCS) in 1977.⁶⁴ Neither Fairchild nor Atari moved quickly to write contracts for new games with independent developers, but they did establish the technological separation of the hardware platform and the game software.⁶⁵

A new dimension of the business model started taking shape in 1980. Some Atari programmers defected to found Activision, the first independent software company in the home video game market.⁶⁶ The company's initial goal was to create games for the VCS model 2600. Activision soon released its first games and was an overnight success. That encouraged other startups in independent game production, which represented a turning point for the industry. From then on, a manufacturer entering the console market had to attract both gamers and developers to the platform to ensure its success.

Today's platform business model originated with the introduction of Nintendo's Nintendo Entertainment System (NES) in the mid-1980s. Nintendo actively pursued licensing agreements with game publishers in order to support the NES, and was the first console maker to introduce a security chip that locked out unlicensed games. Both practices have been since adopted by virtually all console manufacturers.

Just as computer operating systems must attract applications software and users to succeed, video game platforms must attract game publishers and gamers: in this sense, both markets are two-sided. Platform competition in the video game industry is extremely fierce: a new round of competition takes place every 4–5 years when new models are released; the two most successful platforms typically end up with more than 80 percent of the market, and one platform generally achieves clear dominance.⁶⁷ Moreover, 50 percent of console sales come within 3 months of the introduction of a new model, and fully 80 percent within 9 months.⁶⁸ Thus a platform either succeeds or flops quickly, and platform owners understand that having a solid library of games available at launch is a necessary condition for success.

3.2.2 Attracting game developers

It is interesting to compare the personal computer application industry and the game publishing industry. Both came into being in order to exploit the popularity of the

⁶⁴ Kent (2001, p. xiii). The Video Computer System was also known as the Atari 2600.

⁶⁵ Kent (2001, pp. 94–98).

⁶⁶ Still other Atari programmers left the company to form Imagic. Like Activision, Imagic enjoyed success very quickly. See Kent (2001, pp. 192–193, 197).

⁶⁷ Clements and Ohashi (2004).

⁶⁸ Coughlan (2000, p. 12).

early platforms in their respective industries, the Apple II and the Atari VCS. Moreover, killer applications have played important roles in both industries. It is noteworthy, however, that unlike killer apps for PCs, virtually all killer games have been produced by console makers themselves—Pong (Atari), Mario Bros and Pokemon (Nintendo), and Sonic the Hedgehog (Sega). Note, too, that although the video game industry had become two-sided by the 1985 release of NES in the United States, first-party (or in-house) games continued to account for a large share of total games developed for consoles as late as 1995. In that year, Nintendo was still developing 57 percent of Nintendo 64 software in-house.⁶⁹

The first console to break with this tradition was Sony's Playstation, for which only 23 percent of the games were developed in-house.⁷⁰ Neither the original Playstation nor its successor, Playstation II, could boast of any killer game in the sense Sonic and Mario had been killer games for Genesis and NES respectively. Instead, Playstation achieved supremacy by attracting the support of game companies that developed a wide variety of popular offerings.

Key to Playstation's popularity with game developers were Sony's excellent development tools, which earned its platform a reputation for being very easy to program in comparison to Nintendo's Nintendo 64 and Sega's Saturn.⁷¹ Furthermore, Sony made life easier for developers by opting to store games on CD-ROMs, while Nintendo continued to use cartridges. CD-ROMs were substantially cheaper to make in quantity than cartridges,⁷² and the medium also allowed for much faster adjustments in production to meet changes in demand.⁷³ These two factors enabled Sony to earn the same per-unit profit as Nintendo while charging substantially lower licensing fees.⁷⁴

The importance of developer support in the video game industry, as in the computer industry, became clear when Microsoft launched the Xbox console in 2001. The company announced two programs, the Independent Developer Program and the Incubator Program, to encourage smaller developers by providing free software tools and

⁶⁹ Coughlan (2001).

⁷⁰ In fact, even among the 23 percent of games produced in-house, a large part came from recently acquired publishing companies such as Psygnosis, creator of the hit game *Lemmings*.

⁷¹ Sega also launched its 32-bit platform, the Saturn, in 1995. But at launch time this platform had already developed a reputation for being very difficult to program. Kent (2001).

⁷² At the time it cost about \$1.50 to press a 640-megabyte CD and about \$12 to manufacture an 8-megabyte cartridge.

⁷³ Campbell-Kelly (2003, pp. 287–88).

⁷⁴ Kent, (2001, p. 513).

waiving normal pre-publishing requirements.⁷⁵ Furthermore, Microsoft had extensive meetings with developers before the hardware specs for the console were set and incorporated many of their suggestions into the final design. Microsoft also made it easier for developers with PC experience to develop games for the Xbox by relying on DirectX (a collection of Windows APIs that serves as the foundation for most PC games) in the design of the console.

3.2.3 Hardware

In contrast with personal computers, the hardware platform has remained tightly integrated with the software platform in the video game industry. Console makers design their own operating systems to take advantage of technological advances in hardware. It is also noteworthy that the most recent consoles contain operating systems whose functionality extends beyond simple support for video games. For example, the recently released Playstation X (PSX) includes a DVD player and Internet connectivity.⁷⁶

3.2.4 Pricing structures

Perhaps the most interesting contrast with the personal computer industry is the pricing structure adopted by game console manufacturers. The pricing model that emerged with the NES in the mid-1980s (and is still prevalent today) is based on selling the console to end-users at (or below) the marginal cost of production. Console makers then rely on revenue from games produced in-house along with royalties from games sold by independent developers to recover fixed development costs.⁷⁷ In multi-sided market jargon, users pay a direct access fee for the right to access the platform—but a fee that is less than the marginal cost of serving them.

Developers pay no access fees. Their access is, in fact, subsidized through the provision of numerous development tools at little or no cost. But they are charged usage

⁷⁵ Microsoft PressPass, "Microsoft Embraces the Worldwide Independent Video Game Developer Community," 7 November 2000, <http://www.microsoft.com/presspass/press/2000/Nov00/XPKPR.asp> (downloaded 21 June 2004).

⁷⁶ "The Complete Home Entertainer," *The Economist*, 27 February 2004.

⁷⁷ For example, Nintendo sold the original NES for \$130 and charged 20 percent plus \$14 per game sold; Nintendo 64 was sold at \$250 with \$20 or more per-game royalties. When first released, Sony's Playstation 2, Nintendo's GameCube and Microsoft's Xbox all sold for around \$300 and charged \$8–\$10 royalties per game to independent game publishers. In each case, competition and falling component prices prompted console makers to cut hardware prices. In 1996, Panasonic's REAL 3DO, Sony Computer's PlayStation and Sega's Saturn, all priced at \$199, were sold at a loss of between \$50 and \$80 per unit. In September 2003, Microsoft had dropped the price of the Xbox to \$179, for a loss of around \$100 per console. C. Thomas Veilleux, "Costly Battle for Share in Video-Game Arena," *HFN*, 3 June 1996, p. 4; Wachovia Securities, "Company Note—MSFT: Hit By Double Whammy—Nintendo and HPQ Announcements," 24 September 2003, p. 2.

fees in the form of royalties that depend on the number of end-users who buy their games.⁷⁸ This pricing model is the exact opposite of the one used by computer software platforms—a fact that is all the more remarkable since video game consoles are technologically equivalent to computer operating systems integrated with a hardware platform.

3.3 Personal Digital Assistants

Personal Digital Assistants (PDAs) provide users with electronic applications such as address books, calendars, schedule organizer, memo-writing capabilities, etc. Like computers, PDAs are based on a software platform—the best-known and most widely used being the Palm OS. Palm’s history and business strategies are well documented elsewhere.⁷⁹ We therefore focus on elements directly relevant to software platforms.

3.3.1 Evolution of business models

The Palm Pilot, introduced in 1996, was not the first PDA. It was the first to achieve widespread popularity, however: after selling 360,000 units in its first year, Palm reached cumulative sales of 2 million in 1998 and 18 million in 2002.

The first Palm was a fully integrated device for consumers, integrating the hardware, an operating system, and a major application. Its early success seems to have been linked to ease-of-use, low prices,⁸⁰ good handwriting recognition, and PC connectivity. Subsequently, Palm’s popularity with users attracted a large community of developers of complementary software and hardware,⁸¹ which Palm actively encouraged and which has become known as the Palm Economy. Thus, much like video game consoles and PCs, Palm evolved from a one-sided, consumer product into a two-sided

⁷⁸ This is true if we define a unitary interaction between users and game developers as the purchase of a game. One could alternatively consider that a unitary interaction occurs each time a game is being played by a user. In this case the royalties are positive *access* fees for users, since according to this definition they do not depend on the intensity of usage of the console, i.e., on how often users play games.

⁷⁹ See Gawer and Cusumano (2002, chapter 6).

⁸⁰ To this day, Palm charges less than its competitors. The lowest priced Palm PDA, the Zire, costs only \$99 while the lowest priced Pocket PC (the second most popular PDA operating system) costs \$199. See “Zire 21,” <http://www.palmone.com/us/products/handhelds/zire21/> and “Dell Handhelds,” <http://www1.us.dell.com/content/products/compare.aspx/handhelds?c=us&cs=04&l=en&s=bsd> (downloaded 6 June 2004).

⁸¹ Today Palm has more than 320,000 registered software developers. Palmsource, “Developers,” <http://www.palmsource.com/uk/palmos/developer.html> (downloaded 19 August 2004).

market platform. The process took just two years, compared to 10 years for video games and close to three decades for the computer industry.⁸²

The next step in Palm's platform strategy was to license the Palm operating system to manufacturers of other consumer devices, most importantly cell phones. Then, starting in 2000, Palm decided to start licensing the Palm OS to competing handheld manufacturers. The goal was to expand the user base of its operating system platform, whose dominance had begun to come under pressure from Microsoft's Pocket PC OS. Thus, although Palm allowed the cloning of its hardware platform, it kept the development of the software platform in-house.

In 2003, Palm split into PalmSource, its operating system-division which continues to license the Palm OS to a broad range of handheld devices (including smart-phones), and PalmOne, the hardware division, which at the same time acquired rival hardware maker Handspring⁸³.

3.3.2 Complements: applications and hardware add-ons

Graffiti, Palm's hand-recognition application, is generally regarded as the killer application that drove adoption of the Pilot, much like VisiCalc drove the success of the Apple II. It made PDAs popular because it gave users an easy and appealing way to enter data into their address-books, to write short memos, etc.

However, Graffiti was produced in-house, like virtually all early applications for the Palm. Although Palm's founders were fully aware of its potential to become a software platform at the outset, the Pilot was initially conceived as a vertically integrated consumer device. This strategy reflected lessons learned from the early failures in the PDA business⁸⁴ and the initial difficulties encountered in attracting independent developers.

Palm held its first developer conference in late 1997.⁸⁵ Since then, it has been aggressively involved in the development of the Palm Economy, with efforts in this

⁸² Of course, this difference has a lot to do with the relative stages of advancement in hardware technology at the time each of these industries was created. It is arguably much easier to create a software platform today than it was in the 1960s.

⁸³ John G. Spooner and Richard Shim, "Palm to acquire rival Handspring," *news.com*, http://news.com.com/Palm+to+acquire+rival+Handspring/2100-1041_3-1012981.html?tag=nl (downloaded 14 September 2004)

⁸⁴ Famous ones include Palm's own Zoomer and Apple's Newton.

⁸⁵ Palm Computing Press Release, "3Com Rolls Out New Developer Program to Bring Internet Content to the Palm VII™ Connected Organizer," 2 December 1998, <http://www.palmone.com/us/company/pr/1998/devpr.html> (downloaded 21 June 2004).

direction very similar to Microsoft's. Like any computer operating system vendor, Palm realized that more software and hardware complements make devices running the Palm OS more attractive, and therefore allow Palm to sell more copies of the operating system.

Palm helped its developer community stabilize interface standards and facilitated business and technical exchanges among them. For example, Palm worked with five other firms to found the SyncML Initiative, aimed at developing a standard open protocol for data synchronization. The company also regularly offered Palm OS development classes, and it encouraged other activities among its community of users through developer portals.⁸⁶ In addition, Palm provided business resources to developers, including joint development, marketing, and bundling. The most important resources offered were its software development forums, which helped third-party developers start their own businesses. Palm even created a \$50 million venture capital unit called Palm Ventures to support businesses focusing on Palm OS applications. Today, PalmGear, the leading Internet provider of software and hardware products for the Palm OS platform, offers more than 22,000 applications.⁸⁷

3.3.3 Hardware platform

Initially, Palm sold a fully integrated consumer device. But after recognizing that most of the value of its platform resided in the operating system, it chose to license it to manufacturers of other hardware including competing PDA makers—among them Handspring, Sony, IBM, Qualcomm, and Supra.

3.3.4 Pricing structure

Given the similarity of the relationship between Palm and the Palm Economy to the one between Microsoft and the PC applications industry, it is perhaps not surprising that the Microsoft and Palm platform pricing structures are virtually identical. Software and hardware developers are charged little or nothing for support from Palm, and there are no variable fees parallel to the royalties collected in the video game industry.⁸⁸ Palm's

⁸⁶ Gawer and Cusumano (2002, p. 203–206).

⁸⁷ PalmGear, "Education and Enterprise," <http://www.palmgear.com/index.cfm?fuseaction=software.enterprise&SID=E35F1560-CF18-AC90-8352FED820EF5049&PartnerREF> (downloaded 14 September 2004).

⁸⁸ One could, however, argue that the venture capital support to developers possesses a variable fee dimension if, for example, Palm retains an equity position in the newly created ventures. This would mean that it capitalizes on successful complements, much like video game console makers capitalize on successful games through royalties. We were unable to find any detailed data regarding the funding deals made by Palm Ventures, but we suspect that net revenues from venture capital deals are small compared to revenues from licensing the Palm OS.

revenues come from licensing its operating system to device vendors and from direct sale of its own devices.⁸⁹

3.4 Smart Mobile Phones

The market for smart phones has exploded during the past two years; indeed, in 2003, sales of smart phones overtook those of PDAs.⁹⁰ Smart phones are the most sophisticated category of wireless devices, possessing advanced capabilities and data services (calendar, address book, computer-like applications, wireless access to content, etc.),⁹¹ as well as mobile phone capacities. The leading software platform for smart-phones is the Symbian operating system,⁹² on which we focus here.

3.4.1 Evolution of business model

Symbian was established as an independent software licensing company in June 1998 by Motorola, Ericsson, Nokia, and Psion (a PDA vendor), with the goal of creating a common operating system for wireless information devices based on Psion's EPOC 32 OS.⁹³ Subsequently, Matsushita (owner of the Panasonic brand), Sanyo, Sony, Fujitsu, Siemens, and Samsung all became Symbian licensees, and some of them also became Symbian shareholders. The majority of Symbian's shareholders are among the top 10 handset vendors.

3.4.2 Applications

Initially most Symbian applications were developed in-house and distributed to its hardware licensees, or directly developed and installed by the hardware licensees themselves. Symbian provides its licensees with customizable user interfaces, which allow them to differentiate their products, as well as color support and advanced Internet connectivity.⁹⁴ Handset manufacturers integrate the Symbian OS with their hardware and

⁸⁹ As mentioned above, this was true until 2003. After the split, PalmSource became the operating system licensor and PalmOne the hardware vendor (one of many other PalmSource licensees).

⁹⁰ 12 million vs. 11 million. "PDA, RIP," *The Economist*, 16 October 2003.

⁹¹ It is unclear where the line between smart phones and PDAs with integrated phone capability should be drawn. According to some, the difference is that PDAs use touch-screens and pen for input while smart phones use keyboards. See, for example, Phone Scoop, "Smartphone," www.phonescoop.com/glossary/term.php?gid=131 (downloaded 16 September 2004).

⁹² In 2003, of 12 million smart phones shipped, 6.6 million were based on the Symbian OS. SmartPhoneToday, "Symbian Doubles Smartphone Shipments," 6 May 2004, <http://www.smartphonetoday.com/articles/2004/5/2004-5-6-Symbian-Doubles-Smartphone.html> (downloaded 16 September 2004).

⁹³ "Ericsson, Motorola, Nokia & Psion Create Symbian Alliance," *Newsbytes*, 24 June 1998.

⁹⁴ "Ericsson, Motorola, Nokia & Psion Create Symbian Alliance," *Newsbytes*, 24 June 1998.

sometimes their own applications, and then market their mobile phones through network operators such as Vodafone and T-Mobile.

However, as mobile phone capabilities developed, a market for third-party applications for smart phones emerged and expanded spectacularly, with products ranging from productivity applications (communication, data and file management) to entertainment (games, media players). The number of Symbian applications offered by third-party vendors rose from 1,323 to 2,954 between 2003 and 2004.⁹⁵ Noteworthy examples include Access' NetFront browser, RealNetworks' Mobile RealPlayer and Macromedia's Flash Player for Symbian OS.

Symbian seems to have succeeded in creating a "Symbian Economy" similar to the Palm Economy. The company supports this economy very actively. Through its Web site, Symbian offers SDKs, programming languages (Java, C++, OPL,⁹⁶ and Visual Basic), development tools, APIs and documentation to independent developers. It also holds industry-wide events⁹⁷ and even organizes application contests.⁹⁸ In addition, Symbian relies heavily on joint efforts in making improvements to its platform, setting up a partnership initiative in 2002 "to promote the creation of innovative software and hardware technology solutions."⁹⁹ It also develops/licenses products with/from software companies including Oracle (wireless access to data residing in Oracle databases), IBM (wireless enterprise applications for users of Symbian-based smart phones and communicators), Sybase (SQL anywhere and UltraLite deployment technology for smart phones), Metrowerks (CodeWarrior software development tools), Infomatrix (file format converters for MS Office e-mail attachments), Opera Software (Opera browser), RealNetworks (RealPlayer Mobile), and ARM (Jazelle technology software).

⁹⁵ "Symbian H1 2004 results," <http://www.symbian.com/press-office/2004/pr040804.html> (downloaded 14 September 2004).

⁹⁶ Open Programming Language.

⁹⁷ Symbian Feature, "At the Heart of Smartphone Evolution," 11 March 2002, <http://www.symbian.com/news/2002/devexpo-feat.html> (downloaded 15 August 2003); "Symbian Exposium03—Invitation to Press," 10 April 2003, <http://www.symbian.com/press-office/2003/pr030410.html> (downloaded 15 August 2003).

⁹⁸ Symbian Press Release, "Nokia names top Symbian applications globally," 15 April 2002, <http://www.symbian.com/news/2002/pr020415b.html> (downloaded 16 September 2004).

⁹⁹ Symbian Press Release, "Symbian opens its source code to third party software and hardware developers," 23 April 2002, <http://www.symbian.com/press-office/2002/pr020423.html> (downloaded 4 August 2003).

3.4.3 Hardware

What sets Symbian apart from all other software platforms is its status as a joint venture by competing mobile phone handset manufacturers. No such entity exists in any other industry. In fact, it is interesting to note the stark contrast in platform strategy compared to Windows CE, Symbian's main competitor in the market for smart-phone operating systems. Microsoft's approach has been to partner with low-cost original design manufacturers (ODMs) to produce Windows-powered, network-operator-specific phones,¹⁰⁰ thereby bypassing the brand-name handset makers.¹⁰¹ The idea is to commoditize the hardware and make the industry revolve around a proprietary software platform controlled by the operating system vendor and the telecom network operator. That, of course, is similar to the way the PC industry has evolved, with the operator playing a similar role to that of the microprocessor manufacturer (i.e. Intel) in PCs. This platform would have three sides: users, application developers, and hardware makers. In contrast, leading hardware makers would like to see the software commoditized instead—which is why Symbian champions open standards—and have the industry revolve around hardware platforms that share an open software platform not controlled by any one firm. This platform would be two-sided: users and application developers. The model resembles the PC industry before the emergence of the IBM PC and DOS, with CP/M as a more or less open software platform that hardware manufacturers could customize and integrate into their machines.

3.4.4 Pricing structure

Symbian's pricing structure is very similar to the ones adopted by PalmSource and the software platforms in the computer industry. Symbian invests considerable effort and financial resources in developer support, but application developers pay little if anything in return. Symbian then receives royalties¹⁰² on each Symbian OS-based phone shipped by hardware licensees, as well as consulting fees for helping licensees implement the Symbian OS. Thus, there are no usage fees, negative access fees for developers, and,

¹⁰⁰ The best example of this approach is France's Orange operator, which sells phones built by HTC, one of many Taiwanese ODMs, and running Windows. In September 2003, Microsoft finally managed to engage one big handset maker, though not exclusively: Motorola will base its MPx200 phone on Windows CE, but it continued licensing Symbian and using its own OS, MotoJUIX. See "The Third Way," *The Economist*, 18 September 2003.

¹⁰¹ It is noteworthy that the latter also use ODMs extensively, by outsourcing parts of their manufacturing processes.

¹⁰² From Symbian OS v7.0 onwards, the royalty has been set at \$7.25 per unit for the first 2 million units shipped by a licensee and \$5 per unit thereafter.

assuming royalties charged to hardware licensees get passed on to consumers, positive access fees for end-users.

3.5 Digital Content Devices

As the Internet and wireless telecommunications networks have become faster and capable of transmitting larger chunks of data, the market for digital content has expanded greatly. Access to digital content—including music, multimedia clips, and games—is controlled by software platforms, which provide the technology that enables users to “consume” the content.

At this early stage of development, different digital content platforms have adopted different business models. In particular, Apple’s iPod is a one-sided consumer product, while NTT DoCoMo’s i-mode is a two-sided platform, and Real’s RealPlayer is something of a mix of the two. Here we consider these platforms, focusing on the first two.

3.5.1 Description

Apple’s iPod is a portable digital device for listening to music stored in Apple’s AAC digital format. A regular iPod can hold 5,000 songs on average, which users can purchase and download via the Internet at Apple’s online music store, iTunes.

i-mode is a “third-generation” mobile-phone service introduced by NTT DoCoMo in 1999¹⁰³ that is still the market leader in Japan. It is an always-on wireless service, which allows consumers equipped with i-mode-powered handsets to exchange e-mails and access a wide variety of content—including news, personal finance, games, and ring-tones—through i-mode’s portal.

3.5.2 Content provision

Perhaps the most interesting difference between these two is their relationship with content providers. iPod delivers music¹⁰⁴ from the iTunes music store, an online shop run by Apple, where consumers buy songs for 99 cents individually (or entire albums for \$9.99), which they download onto their computer, manage using the iTunes software and Quicktime media player, and then transfer to the iPod portable device.¹⁰⁵ Rights to distribute the approximately one million songs available on iTunes have been

¹⁰³ At the time it was 2.5 generation.

¹⁰⁴ At least for now. There is speculation that future versions will include games and other applications.

¹⁰⁵ Songs come wrapped in Apple’s digital rights management (DRM) technology. The DRM is intended to limit the use of AAC files to the QuickTime media player (installed on the user’s computer), iTunes and the iPod as well as to limit to ten, the number of times a particular play list can be burned onto a CD.

purchased by Apple from record labels, which receive a percentage of revenues for songs downloaded. Thus, Apple's iPod/iTunes business model is no different from that of supermarkets, which must first acquire supplies before selling them to consumers.

By contrast, none of the content that users can access through the i-mode portal is owned in any sense by DoCoMo. The company simply provides a digital interface through which users and content providers interact. This business model is two-sided, as the value of the service to users depends on the variety of content available (i.e. the number of content providers signed up) and the value to content providers depends on the number of subscribers to i-mode. It is noteworthy that two-sidedness was one of the major strategic differences between i-mode and another early failed attempt in the 3G market, Vizzavi, which relied exclusively on its own content.¹⁰⁶ Two of the main reasons behind i-mode's success in attracting content providers¹⁰⁷ were the choice of the c-HTML language over WAP (wireless application protocol), which greatly reduced costs for providers already possessing an Internet Web site, and the creation of an advanced billing system enabling content providers to charge small monthly fees to users.

3.5.3 Hardware

Apple's iPod-iTunes business model is vertically integrated, with Apple providing both the software (Quicktime and iTunes) and the hardware (iPod). By contrast, handsets designed specifically for i-mode are supplied by a number of manufacturers with close relationships to DoCoMo—in particular Matsushita (owner of the Panasonic brand), NEC, Fujitsu, and Sharp.

3.5.4 Pricing structure

As explained above, Apple's iPod-iTunes business model is one-sided, so there is nothing unusual about it. Like any other merchant, Apple buys music from record labels and resells it to users, along with the iPod device. According to most analysts, Apple barely breaks even on the music part of the business, making its money on the hardware/software bundle.

The i-mode pricing structure issue is more interesting. DoCoMo does not charge fixed fees to content providers for making their content accessible through i-mode, even though there are reportedly hundreds of new providers queuing up every month for this privilege. It charges relatively modest subscription fees to users (including significant

¹⁰⁶ Vizzavi was a joint venture between Vodafone and Vivendi, the former owner of Universal Studios.

¹⁰⁷ Today the service has more than 60,000 official (i.e., directly accessible through the i-mode portal) and non-official sites.

handset subsidies) but uses its sophisticated billing system to charge them variable fees based on their intensity of content usage.¹⁰⁸ Furthermore, it offers content providers the option of joint billing (which makes life easier for everyone, since consumers receive a single unified bill) in exchange for 9 percent of the content revenues.

4 REGULARITIES AND PUZZLES

The four, computer-based industries surveyed here have notable similarities, as well as striking differences that give rise to interesting puzzles—some of which we highlight in this concluding section.

4.1 Integration and Multi-sidedness

Within individual industries, we have seen significant differences in the extent of integration across companies and over time. The less integration, generally the greater the extent of multi-sidedness, the greater the complexity of the task of “getting all sides on board,” and the more the market is relied upon to lower cost and improve quality. On the other hand, greater integration provides greater control over the total system as experienced by the end-user.

One can see a broad tendency toward less integration over time in these industries, reflecting the development of markets that can be relied upon for low-cost, high-quality complementary products. For example, the early video game console manufacturers wrote their own games and operating systems—nobody else was going to do it. A robust industry of game developers has since emerged, and console vendors have found it useful to rely more on the market to supply desired games. They have moved from offering a consumer product to a two-sided platform.

Similarly, in the absence of a capable set of outside suppliers, Palm initially wrote all the applications for its PDA and sold a software platform/hardware platform/applications bundle as a consumer product. As outside applications suppliers developed, Palm moved to a two-sided strategy relying in part on their efforts. Currently, the Palm Economy also includes independent suppliers of hardware platforms and peripherals, and PalmSource was spun off to focus tightly on its software platform, the Palm OS.

Computers offer the most extreme example of this sort of evolution. Early users of mainframe computers bought the hardware platform, operating system, applications, and peripherals from the same firm. As the mainframe computer industry matured after

¹⁰⁸ Indeed, the system does not charge based on the time spent on the network (it is an always on service), but rather based on the amount of data transferred.

the introduction of the IBM System/360, independent vendors of applications and peripherals emerged. By the time the IBM PC was introduced in 1981, there were numerous independent software applications vendors. Today, Microsoft's Windows software platform plays a central role in personal computers. But Microsoft is not integrated into hardware platforms, and it is one of many vendors of applications programs and, to a very minor extent, peripherals (mice, keyboards) that work with the Windows platform.

Some important differences in integration within industries reflect differences in business strategies, which in turn reflect different perceptions of competitive advantage and/or market trends. The differences between Apple and Microsoft are classic. Apple's greater integration facilitated delivery of quality and reliability, and eased the task of coordinating changes in hardware and operating systems. Microsoft's reliance on a competitive market for hardware led to both hardware innovation and low hardware prices—and, with hindsight, this seems to have been the better business strategy. As the creator of the leading handset manufacturers, Symbian's strategy aims toward a smart-phone industry with a cheap, open software platform and differentiated hardware platforms. By contrast, software platform vendor Microsoft aims to create a PC-like smart-phone industry in which relatively undifferentiated hardware platforms all use a proprietary software platform. In setting up iPod, Apple made a choice to distribute only content it owned, while NTT DoCoMo chose to own none of the content that flows through its i-mode system. Purely as a technical matter, each could have made the other choice.

4.2 Pricing Strategies

As noted at the outset, most vendors of multi-sided platforms seem to earn the bulk of their profits from only one side. This is itself something of a puzzle. The pattern characterizes all the industries we have discussed here. In PCs and PDAs, all the money is earned from end-users. Neither Palm nor Apple nor Microsoft (in either sector) charges applications writers or peripherals manufacturers for the right to access to its software platform. In fact, all expend resources to facilitate access. Microsoft profits by licensing its operating systems to end-users through hardware manufacturers; Apple profits by selling hardware/software bundles to end-users. The two companies created from the original Palm Computing collectively employ both channels.

As we have noted, pricing strategies in the video game industry are dramatically different. Video game platforms are sold to end-users as integrated bundles of hardware and operating systems, but these sales yield little or no profit. As with personal computers,

video game platform vendors subsidize access by applications (i.e., game) developers. In stark contrast with personal computers, however, game developers pay platform vendors fees for use of the platform in the form of royalties for each game sold.

When pricing strategies differ so sharply between industries using broadly similar technologies and integration strategies but are nearly identical within those industries, it seems clear that industry-level cost and/or demand differences (rather than, say, firm-level differences or strategic choices) are playing the leading role in shaping strategies. As noted, however, the literature does not yet contain a rigorous explanation of the differences in pricing between PCs and PDAs, on the one hand, and videogames on the other. This is perhaps the most interesting puzzle we have encountered in this research.

Differences in pricing strategies within the other industries discussed here seem mainly to reflect strategic choices. Symbian's owners want to make money by selling smart-phone handsets, not by licensing the Symbian OS, while Microsoft wants to make money licensing its operating system. At the end of the day, of course, end-users purchase hardware/software bundles, and neither Symbian nor Microsoft intends to earn revenues from applications developers. Apple makes money on the iPod device (a hardware/software bundle), not on the music it buys and resells to iPod owners. NTT DoCoMo makes money on the network traffic generated by i-mode services, not by charging high prices for i-mode handsets or by levying access or usage fees on content developers. In these last two cases, alternative pricing strategies would be possible and, arguably, plausible—though they may not be optimal.

4.3 Bundling and Feature Accretion

There is a single, simple story to tell here: driven by advances in hardware technology, hardware and software platforms (as well as applications programs) have steadily added features and functionality. While this process has been most visible in PC operating systems, thanks to antitrust cases brought against Microsoft in the United States and elsewhere,¹⁰⁹ it is not obviously the most dramatic. Mobile phones, once capable only of placing and receiving voice telephone calls, have become smart phones with cameras, Internet browsers, instant messaging capability, and a host of other features and functions. Today's PDAs are much more capable than the Apple Newton or even the first Palm Pilot. This advance has been supported by increasingly complex and capable operating

¹⁰⁹ In 1998, the United States Department of Justice and several states lodged complaints against Microsoft for bundling its Internet Explorer Web browser with the Windows operating system. More recently, the European Commission issued a similar complaint with regard to the bundling of Windows Media Player with Windows. *United States v. Microsoft Corporation*, Civil Action Nos. 98-1232 and 98-1233; *EC Commission/Microsoft*, Case COMP/C-3/37.792.

systems and applications programs, which are in turn made possible by increases in the power of the handheld platform. Video game platforms now include the ability to play DVDs and access the Internet. The market for PC modems has almost vanished, since modems are built into most new PCs—often along with Ethernet ports and wireless networking capability.

As discussed at the outset, there is little mystery about what is going on here. Increases in hardware power make it possible for software to do more without undue sacrifice in speed. The earliest PCs were too slow to support a graphical user interface, for instance, while current models do so with ease. As disk storage has become cheaper, the cost of adding new features to an operating system or an applications program has fallen rapidly, and new features broaden product appeal. Many end-users agree that software is too complex. But there is much less agreement on what features should be dropped, and most users acknowledge that they simply ignore features they do not find valuable. Similarly, antitrust authorities who view product integration with suspicion are standing in the path of what seem to be strong and broad technology-driven trends.

4.4 Concluding Observations

As a general matter, forecasting the evolution of technology is hazardous business. Many technological wonders forecast with confidence a few decades ago have yet to appear on the market, while our everyday lives have been profoundly affected by innovations, like email and GPS navigation, that entered the world of commerce with little fanfare. Still, it is hard not to be confident that computer-based industries will grow in importance over at least the next few decades. Hardware performance is almost certain to rise even as machinery size falls; most information and entertainment can be reduced to digital content; and software is arguably the most flexible of creative media.

This is good news for consumers—and, we would argue, for economists. The computer-based industries discussed here exhibit fascinating, rapidly changing patterns of product design, integration, and pricing. Many firms pursue interesting multi-sided platform strategies. As these industries evolve and others like them emerge, there will be much for economists to study and important lessons for us and for antitrust authorities to learn.

References

- Armstrong, Mark (2002). "Competition in Two-Sided Markets," *Nuffield College, Oxford University*, Working Paper.
- Bakos, Yannis and Eric Brynjolfsson (1999). "Bundling Information Goods: Pricing, Profits, and Efficiency," *Management Science*, vol. 45, pp. 1613–1630.
- Baye, Michael R. and John Morgan (2001). "Information Gatekeepers on the Internet and the Competitiveness of Homogeneous Product Markets," *American Economic Review*, vol. 91, pp. 454–474.
- Caillaud, Bernard and Bruno Jullien (2003). "Chicken and Egg: Competition among Intermediation Service Providers," *RAND Journal of Economics*, vol. 34, pp. 521–552.
- Campbell–Kelly, Martin (2003). *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry*, (The MIT Press: Cambridge, MA).
- Carlton, Jim. (1998). *Apple*, (HarperBusiness: New York, NY).
- Carrol, Paul. (1994). *Big Blues The Unmaking of IBM*, (Crown Publishers: New York, NY).
- Clements, Matthew and Hiroshi Ohashi (2004). "Indirect Network Effects and the Product Cycle: Video Games in the US 1994–2002," mimeo, University of Tokyo.
- Coughlan, Peter J. (2000). *Note on Home Video Game technology and Industry Structure*, (Harvard Business School: Boston, MA), p. 12.
- Coughlan, Peter J. (2001). *Competitive Dynamics in Home Video Games (K): PlayStation vs. Nintendo64*, (Harvard Business School: Boston, MA).
- Davis, Steven J., Jack MacCracken and Kevon M. Murphy (2001). "Economic Perspectives on Software Design—PC Operating Systems and Platforms," in *Microsoft, Antitrust and the New Economy: Selected Essays*, ed. David S. Evans, pp. 361–419.
- Evans, David S. (2003). "Some Empirical Aspects of Multi-Sided Platform Industries," *Review of Network Economics*, vol. 2, pp. 191–209.
- Evans, David S. (2003b). "The Antitrust Economics of Multi-Sided Platform Markets," *Yale Journal on Regulation*, vol. 20, pp. 325–381.
- Evans, David S. and Richard Schmalensee (2001). "Some Economic Aspects of Antitrust Analysis in Dynamically Competitive Industries," in *Innovation Policy and the Economy*, vol. 2., eds. A. Jaffe, J. Lerner and S. Stern, pp. 1–50.
- Evans, David S., and Michael Salinger (2004). "The Role of Cost in Determining When Firms Offer Bundles and Ties," http://papers.ssrn.com/sol3/papers.cfm?abstract_id=555818.
- Evans, David S. and Michael A. Salinger (2004b). "Why Do Firms Bundle and Tie? Evidence from Competitive Markets and Implications for Tying Law," *Yale Journal on Regulation*, forthcoming, <http://ssrn.com/abstract=550884>.
- Freiberger, Paul and Michael Swaine (2000). *Fire in the Valley*, (McGraw–Hill: New York, NY), pp. 11–17.

- Gawer, Annabelle and Michael A. Cusumano (2002). *Platform Leadership: How Intel, Microsoft, and Cisco Drive Industry Innovation*, (Harvard Business School: Boston, MA).
- Hagiu, Andrei (2004). "Platforms, Pricing, Commitment and Variety in Two-Sided Markets," Princeton University Doctoral Dissertation.
- IDC (1988). "Macintosh Peripherals Markets: High-Resolution Monitor and Printers Review and Forecast, 1987-1992," Report #3713.
- IDC (1993). "Desktop Operating System Review and Forecast: The Second Phase of the Industry Begins," Report #7445.
- IDC (2000). "Worldwide Software Market Forecast Summary, 2000-2004," Report #22766.
- Katz, Michael L. and Carl Shapiro (1994). "Systems Competition and Network Effects," *Journal of Economic Perspectives*, vol. 8, pp. 93-115.
- Kent, Steven L. (2001). *The Ultimate History of Video Games*, (Prima: Roseville, CA).
- Kunkel, Paul (1997). *AppleDesign: The Work of the Apple Industrial Design Group*, (Watson-Guptill Publications: NY).
- Liebowitz, Stan J. and Margolis, Stephen E. (2001). *Winners, Losers & Microsoft*, (The Independent Institute: Oakland, CA).
- Linzmayr, Owen W. (1999). *Apple Confidential*, (No Starch Press: San Francisco, CA), pp. 193-206.
- Mace, Jeff (1995). "Printers (The Drivers Manual)," *PC Magazine*, vol. 14.
- Malone, Michael S. (1998). *Infinite Loop*, (Currency Doubleday: New York, NY).
- Manes, Stephen and Paul Andrews (1994). *Gates*, (Touchstone: New York, NY).
- McConnel, Ben and Jackie Huba (2003). *Creating Customer Evangelists*, (Dearborn Trade Publishing: Chicago, IL), p. 13.
- Microsoft (1995). "Microsoft Developer Relations; Proving Microsoft's Commitment to Third Parties," *Microsoft Corporation White Paper*.
- Microsoft (1998). "Microsoft Developer Relations: Microsoft's Commitment to Third-Party Developer Success," *Microsoft Corporation White Paper*.
- Rochet, Jean-Charles and Jean Tirole (2003). "Platform Competition in Two-Sided Markets," *Journal of the European Economic Association*, vol. 1, pp. 990-1029.
- Rochet, Jean-Charles and Jean Tirole (2004). "Two-sided Markets: An Overview," IDEI Working Paper, http://www.frbatlanta.org/filelegacydocs/ep_rochetover.pdf.
- Salus, Peter H. (1995). *A Quarter Century of UNIX*, (Addison-Wesley Publishing Company: Reading, MA), pp. 209-212.
- Stross, Randall E. (1996). *The Microsoft Way*, (Addison-Wesley Publishing Company: Reading, MA).
- Takahashi, Dean (2002). *Opening the Xbox*, (Prima: Roseville, CA), p. 319.

CESifo Working Paper Series

(for full list see www.cesifo.de)

- 1253 Ngo Van Long and Gerhard Sorger, Insecure Property Rights and Growth: The Roles of Appropriation Costs, Wealth Effects, and Heterogeneity, August 2004
- 1254 Klaus Wälde and Pia Weiß, International Competition, Slim Firms and Wage Inequality, August 2004
- 1255 Jeremy S. S. Edwards and Alfons J. Weichenrieder, How Weak is the Weakest-Link Principle? On the Measurement of Firm Owners' Control Rights, August 2004
- 1256 Guido Tabellini, The Role of the State in Economic Development, August 2004
- 1257 François Larmande and Jean-Pierre Ponssard, EVA and the Controllability-congruence Trade-off: An Empirical Investigation, August 2004
- 1258 Vesa Kanninen and Jenni Pääkkönen, Anonymous Money, Moral Sentiments and Welfare, August 2004
- 1259 Panu Poutvaara and Andreas Wagener, Why is the Public Sector More Labor-Intensive? A Distortionary Tax Argument, August 2004
- 1260 Lars P. Feld and Stefan Voigt, Making Judges Independent – Some Proposals Regarding the Judiciary, August 2004
- 1261 Joop Hartog, Hans van Ophem, and Simona Maria Bajdechi, How Risky is Investment in Human Capital?, August 2004
- 1262 Thomas Eichner and Rüdiger Pethig, Efficient Nonanthropocentric Nature Protection, August 2004
- 1263 David-Jan Jansen and Jakob de Haan, Look Who's Talking: ECB Communication during the First Years of EMU, August 2004
- 1264 David F. Bradford, The X Tax in the World Economy, August 2004
- 1265 Hans-Werner Sinn, Migration, Social Standards and Replacement Incomes. How to Protect Low-income Workers in the Industrialized Countries against the Forces of Globalization and Market Integration, August 2004
- 1266 Wolfgang Leininger, Fending off one Means Fending off all: Evolutionary Stability in Submodular Games, August 2004
- 1267 Antoine Bommier and Bertrand Villeneuve, Risk Aversion and the Value of Risk to Life, September 2004

- 1268 Harrie A. A. Verbon and Lex Meijdam, Too Many Migrants, Too Few Services: A Model of Decision-making on Immigration and Integration with Cultural Distance, September 2004
- 1269 Thomas Eichner and Rüdiger Pethig, Economic Land Use, Ecosystem Services and Microfounded Species Dynamics, September 2004
- 1270 Federico Revelli, Performance Rating and Yardstick Competition in Social Service Provision, September 2004
- 1271 Gerhard O. Orosel and Klaus G. Zauner, Vertical Product Differentiation When Quality is Unobservable to Buyers, September 2004
- 1272 Christoph Böhringer, Stefan Boeters, and Michael Feil, Taxation and Unemployment: An Applied General Equilibrium Approach, September 2004
- 1273 Assaf Razin and Efraim Sadka, Welfare Migration: Is the Net Fiscal Burden a Good Measure of its Economics Impact on the Welfare of the Native-Born Population?, September 2004
- 1274 Tomer Blumkin and Volker Grossmann, Ideological Polarization, Sticky Information, and Policy Reforms, September 2004
- 1275 Katherine Baicker and Nora Gordon, The Effect of Mandated State Education Spending on Total Local Resources, September 2004
- 1276 Gabriel J. Felbermayr and Wilhelm Kohler, Exploring the Intensive and Extensive Margins of World Trade, September 2004
- 1277 John Burbidge, Katherine Cuff and John Leach, Capital Tax Competition with Heterogeneous Firms and Agglomeration Effects, September 2004
- 1278 Joern-Steffen Pischke, Labor Market Institutions, Wages and Investment, September 2004
- 1279 Josef Falkinger and Volker Grossmann, Institutions and Development: The Interaction between Trade Regime and Political System, September 2004
- 1280 Paolo Surico, Inflation Targeting and Nonlinear Policy Rules: The Case of Asymmetric Preferences, September 2004
- 1281 Ayal Kimhi, Growth, Inequality and Labor Markets in LDCs: A Survey, September 2004
- 1282 Robert Dur and Amihai Glazer, Optimal Incentive Contracts for a Worker who Envy his Boss, September 2004
- 1283 Klaus Abberger, Nonparametric Regression and the Detection of Turning Points in the Ifo Business Climate, September 2004

- 1284 Werner Güth and Rupert Sausgruber, Tax Morale and Optimal Taxation, September 2004
- 1285 Luis H. R. Alvarez and Erkki Koskela, Does Risk Aversion Accelerate Optimal Forest Rotation under Uncertainty?, September 2004
- 1286 Giorgio Brunello and Maria De Paola, Market Failures and the Under-Provision of Training, September 2004
- 1287 Sanjeev Goyal, Marco van der Leij and José Luis Moraga-González, Economics: An Emerging Small World?, September 2004
- 1288 Sandro Maffei, Nikolai Raabe and Heinrich W. Ursprung, Political Repression and Child Labor: Theory and Empirical Evidence, September 2004
- 1289 Georg Götz and Klaus Gugler, Market Concentration and Product Variety under Spatial Competition: Evidence from Retail Gasoline, September 2004
- 1290 Jonathan Temple and Ludger Wößmann, Dualism and Cross-Country Growth Regressions, September 2004
- 1291 Ravi Kanbur, Jukka Pirttilä and Matti Tuomala, Non-Welfarist Optimal Taxation and Behavioral Public Economics, October 2004
- 1292 Maarten C. W. Janssen, José Luis Moraga-González and Matthijs R. Wildenbeest, Consumer Search and Oligopolistic Pricing: An Empirical Investigation, October 2004
- 1293 Kira Börner and Christa Hainz, The Political Economy of Corruption and the Role of Financial Institutions, October 2004
- 1294 Christoph A. Schaltegger and Lars P. Feld, Do Large Cabinets Favor Large Governments? Evidence from Swiss Sub-Federal Jurisdictions, October 2004
- 1295 Marc-Andreas Mündler, The Existence of Informationally Efficient Markets When Individuals Are Rational, October 2004
- 1296 Hendrik Jürges, Wolfram F. Richter and Kerstin Schneider, Teacher Quality and Incentives: Theoretical and Empirical Effects of Standards on Teacher Quality, October 2004
- 1297 David S. Evans and Michael Salinger, An Empirical Analysis of Bundling and Tying: Over-the-Counter Pain Relief and Cold Medicines, October 2004
- 1298 Gershon Ben-Shakhar, Gary Bornstein, Astrid Hopfensitz and Frans van Winden, Reciprocity and Emotions: Arousal, Self-Reports, and Expectations, October 2004
- 1299 B. Zorina Khan and Kenneth L. Sokoloff, Institutions and Technological Innovation During Early Economic Growth: Evidence from the Great Inventors of the United States, 1790 – 1930, October 2004

- 1300 Piero Gottardi and Roberto Serrano, Market Power and Information Revelation in Dynamic Trading, October 2004
- 1301 Alan V. Deardorff, Who Makes the Rules of Globalization?, October 2004
- 1302 Sheilagh Ogilvie, The Use and Abuse of Trust: Social Capital and its Deployment by Early Modern Guilds, October 2004
- 1303 Mario Jametti and Thomas von Ungern-Sternberg, Disaster Insurance or a Disastrous Insurance – Natural Disaster Insurance in France, October 2004
- 1304 Pieter A. Gautier and José Luis Moraga-González, Strategic Wage Setting and Coordination Frictions with Multiple Applications, October 2004
- 1305 Julia Darby, Anton Muscatelli and Graeme Roy, Fiscal Federalism, Fiscal Consolidations and Cuts in Central Government Grants: Evidence from an Event Study, October 2004
- 1306 Michael Waldman, Antitrust Perspectives for Durable-Goods Markets, October 2004
- 1307 Josef Honerkamp, Stefan Moog and Bernd Raffelhüschen, Earlier or Later: A General Equilibrium Analysis of Bringing Forward an Already Announced Tax Reform, October 2004
- 1308 M. Hashem Pesaran, A Pair-Wise Approach to Testing for Output and Growth Convergence, October 2004
- 1309 John Bishop and Ferran Mane, Educational Reform and Disadvantaged Students: Are They Better Off or Worse Off?, October 2004
- 1310 Alfredo Schclarek, Consumption and Keynesian Fiscal Policy, October 2004
- 1311 Wolfram F. Richter, Efficiency Effects of Tax Deductions for Work-Related Expenses, October 2004
- 1312 Franco Mariuzzo, Patrick Paul Walsh and Ciara Whelan, EU Merger Control in Differentiated Product Industries, October 2004
- 1313 Kurt Schmidheiny, Income Segregation and Local Progressive Taxation: Empirical Evidence from Switzerland, October 2004
- 1314 David S. Evans, Andrei Hagiu and Richard Schmalensee, A Survey of the Economic Role of Software Platforms in Computer-Based Industries, October 2004