11

When Bigger Is Better

"I want a pair of jeans—32–28," I said. "Do you want them slim fit, easy fit, relaxed fit, baggy, or extra baggy?" she replied. "Do you want them stonewashed, acid washed, or distressed? Do you want them button-fly or zipper-fly? Do you want them faded or regular?" "I just want regular jeans. You know, the kind that used to be the only kind."

—*Barry Schwartz, "The Paradox of Choice"*[1]

INSIDE THIS CHAPTER

- How and why software platforms have expanded over time
- The software platform value proposition
- The economics of bundling features into software platforms

As we first noted in Chapter 2, when measured by lines of code, software platforms have grown steadily and substantially over time.

This pattern holds across software platforms for the same computing device. Linux, the Mac OS, and Windows have all grown rapidly. And it is true across different computer-based systems. Software platforms have gotten larger for handheld devices, mobile telephones, and video game consoles as well as for personal computers.

Table 11.1 shows the growth in the number of lines of code of various software platforms over time. Although numbers are not available for every year, the data show a consistent pattern. The average annual growth rate is about 50 percent. That means that the number of lines of code doubles about every two years.

1. http://www.aarp.org/bulletin/yourlife/many_choices.html.

Table 11.1
Size of Operating System by Year

OS	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	Compound Annual Growth Rate (%)
Measured in millions lines of code															
Red Hat Linux						9		16	27		42	50			33
Windows	3				15		18			35	40				30
Windows CE									0.4			2	2.5		58
Linux Kernel			0.1	0.2	0.5			1.1		2.2		3.8			50
Measured in megabytes															
Macintosh						70	200	250		1,500				4,000	66

Source: Red Hat Linux source code (http://research.microsoft.com/projects/SWSecInstitute/DIMACS-report.pdf); Linux Kernel source code; Apple system requirements.

This might not seem surprising, since hardware is becoming more powerful at a rapid rate, and hardware and software platforms are tightly coupled. The processing power and memory of computing devices have risen because the costs of producing microprocessors and memory have declined rapidly as a result of technological change and scale economies. But those forces cannot explain the growth of software platforms. Innovations such as object-oriented programming and the open-source model have made it more efficient to write platform code, but not dramatically so. There is certainly no Moore's Law operating in the labor-intensive process of computer programming. And while there are scale economies from using the same software platform across an increasing number of devices, there are, as Chapter 2 noted, *diseconomies* from expanding the size of the software platform itself.

Besides, there is something fundamentally different between what's happened with hardware and with software. Hardware has generally gotten smaller and more powerful. In contrast, software platforms have gotten bigger.

Increases in computer speed and storage capacity have indirect effects that help explain some of the growth in software platforms. More code is needed to control more complex and capable hardware. With more memory available, software platform architects and programmers have less incentive to economize on code. Neither effect is large enough, however, to explain the historical growth of software platforms.

The main driver is clear: software platforms have added code primarily to provide *more features* to end users and application developers. They have done this in tandem with the hardware platform that has provided *more features* with which the software platform can work. This chapter documents this expansion over time of the scope of software platforms and describes the economic and technological forces behind it.

Feature Accretion

Three patterns emerge from our study of software platforms across different computing devices:

• Every release of each software platform contains significant new features, and some features are even introduced between releases.

- Over time, software platforms incorporate many features that had been provided by third parties on a stand-alone basis.
- Software platforms generally include the code for all available features; they are seldom offered with a list of optional features from which customers can pick and choose.

More Features

Over time, software platform producers add features that appeal to end users, to application developers, and sometimes to both. They have to do this in part to get end users to buy another version of the software platform, since software platforms, like diamonds, don't wear out. They also have to do this to get software developers to write or modify their applications and thereby increase the value of the software platform. Applications don't wear out either, and new services made available through Application Programming Interfaces (APIs) will enable and entice developers to write new ones or upgrade existing ones.

The Palm OS is a good example. As we discussed in Chapter 6, when the original version was introduced in 1996, it included a feature called Graffiti that recognized a special kind of script. This enabled users to convey written information to the device by stroking a "pen" (a stick) over the screen. It also included a calculator, notebook, address book, calendar, and a utility to synchronize files and contacts in the PDA with desktop computers. The fourth version, released in 2001, added support for Bluetooth wireless technology and bundled America Online and utilities for reading and editing Microsoft Office documents. At the same time, it added many software services for developers, including some for telephony-based applications.[2] By 2004, the Palm OS, now in its sixth version, included APIs that enabled developers to take advantage of modern wireless networks, as well as headset and hands-free support for users, who were increasingly using their PDAs as telephones. It also includes a Web browser and supports advanced Web technologies, multimedia, and sophisticated security.

2. Cameron Crouch, "Palms Gain Expansion Options, Keep Popular, Sleek Design," *PC World*, May 2001; "Palm Revamps Operating System, Adds APIs," *Network World*, July 2, 2001.

Apple's Mac OS has also added features that have made it more valuable to end users and developers. The very first Mac OS in 1984 included a graphical user interface (GUI), calculator, notebook, a simple puzzle game, and a clipboard tool called Scrapbook to move text between applications. By 1988 it also included a color user interface that could be displayed on multiple monitors. Three years later it bundled AppleTalk and AppleShare, which enabled users to share files and printers across a network. In 1994, now on to Mac OS 7.5, Apple included Stickies—a desktop application that provided an electronic version of Post-It Notes. Version 9.0, released in 1999, had an updated version of Sherlock, Apple's search engine, which searched the user's hard drive and the Internet. Video chatting was included in 2003.[3] The most recent version, released in early 2005, includes Dashboard, a visually appealing utility that lets the user run many useful mini-applications called widgets. Among the widgets included are a stock ticker, a weather forecast, a flight tracker, a dictionary, and a translation tool. Over this period Apple also added more APIs that developers could use. These included a sophisticated set of media APIs associated with the QuickTime media platform (discussed in Chapter 8) that have been used by a large number of media applications available on the Mac, including Adobe's Premiere, a popular video editor.[4] Other APIs let developers take advantage of new technologies like Bluetooth and the visual technology behind the Mac OS's good looks.

All software platforms have added features over time that helped users and developers avail themselves of the new opportunities made available by the rapid development of the World-Wide Web. Table 11.2 shows when each of the major platforms on which we have focused added Internet communication protocols, a Web browser, an email client, and a

3. http://www.macos.utah.edu/Documentation/MacOSXClasses/macosxone/macintosh.html.

4. http://www.mackido.com/History/EarlyMacOS.html; http://www.macos.utah.edu/Documentation/MacOSXClasses/macosxone/macintosh.html; Gene Wilburn, "Some Bugs to Iron Out in Mac OS," *The Toronto Star*, January 5, 1995; Steve Wood, "Mac OS O9. I Think I Like It!" *A View from the Classroom*, November 8, 1999 (http://lowendmac.com/macinschool/991108.html); http://adobe.es/aboutadobe/pressroom/presskits/pdfs/premiere50/PREaag.pdf.

Table 11.2
Feature Accrual Across Platforms

	Platform						
Feature	Windows	Mac OS	Palm OS	Windows CE	Symbian	Xbox	Play-Station
Networking	1995	1994	1997	1996	2000	2001	2000
Web browser	1995	1998	2003	1996	1999	NA	NA
Email	1993	2001	1997	1996	1999	NA	NA
Media player	1991	1993	2002	2000	2000	2001	2000

media player.[5] In some cases these additions came in the form of code that was integrated into the operating systems. Microsoft, for example, wove various browser-related features (such as an HTML-rendering engine) into Windows 98. In other cases this came through bundling an application with the software platform. For example, Palm OS 3.0 included a stand-alone Expense application that eased the task of tracking business trip expenses.

5. Bernard J. Reddy, David S. Evans, and Albert L. Nichols, "Why Does Microsoft Charge So Little for Windows?" (National Economic Research Associates paper), October 9, 1998; "Windows For Workgroups 3.11 Launched" (Network Week APT Data Services no. 94), October 15, 1993; "Microsoft Ships Windows with Multimedia Extensions 1.0," *Business Wire*, August 21, 1991; http://kb.iu.edu/data/abmc.html; "What Changed in Mac OS X Version 10.1?" Mac OS History, http://www.macos.utah.edu/Documentation/MacOSXClasses/macosxone/macintosh.html; David Flynn, "New Pilots Fly Higher," *Sydney Morning Herald*, June 3, 1997; "Where is Palm OS 6?" (http://www.palminfocenter.com/view_story.asp?ID=6393); Ian Cuthbertson, "Multimedia on the Move: Sony Clie NX70VG," *The Australian*, February 11, 2003; Marty Jerome, "Put Windows in Your Pocket," *PC/Computing*, January 1, 1997; Jack Schoefield, "Third Strike for Windows CE in Palm Territory," *The Guardian*, April 19, 2000; "Symbian OS Version 6.x Detailed Operating System Overview" (http://www.symbian.com/technology/symbos-v6x-det.html); "Symbian Releases Latest EPOC Technology for Future Smartphones and Communicators," Symbian press release, June 15, 1999 (http://www.symbian.com/press-office/1999/pr990615.html); http://reviews.cnet.com/Microsoft_Xbox/4505-6464_7-7853769-3.html?tag=top; "Playstation2 Launch Drawing A Crowd," *Sun-Sentinel-Ft. Lauderdale*, March 19, 2000; "kelly s. i.," *Mirror*, March 4, 2000.

Features Already Provided by Others

Many features were available, in some form, to users or developers as third-party "add-ons" before they were incorporated into software platforms. This is most obvious in the case of PCs. Internet browsers, file encryption and compression, firewalls, and many other applications were available to end users before they were incorporated into either Mac OS or Microsoft Windows. Similarly, third-party media players preceded integrated ones on both Palm and Symbian operating systems. Third-party console accessories enabled network connectivity and online game downloads as early as 1983, nearly two decades before such functionality was integrated into the major consoles.

Some new features were provided first to developers rather than to end users, sometimes by third-party vendors offering libraries and tools that enable developers to take advantage of new technologies or innovations. In 2000, for instance, Extended Systems released a set of tools that enabled manufacturers to add Bluetooth support into their Palm OS–based devices, a year before Palm added this capability into the Palm OS. Similarly, SoftConnex provided USB connectivity APIs a couple of years before they were incorporated into the Symbian OS.[6]

Several dynamics are at work here. Platform releases need to occur at discrete intervals, and they can only include features that have been fully tested by the release data. For this reason, some features are not released promptly. Other vendors sometimes fill in the resulting gaps by offering add-ons that improve the platform in some dimension. Independent software developers also come up with ideas that the platform developers hadn't even thought about. Many of these attract very few interested users, but others become quite popular. If the original developer doesn't have intellectual property rights on her idea, the software platform vendor can incorporate these innovative features into its own product.

In the late 1980s, for instance, Apple and Microsoft decided to develop their own scaleable font technology instead of using Adobe's PostScript.

6. "Extended Systems Ships Bluetooth Software Development Kit for Handheld Devices," *M2 Presswire*, April 4, 2000; "USB On-The-Go Frees Digital Devices for Direct Connectivity Without A PC," *Business Wire*, November 18, 2002; "SoftConnex Joins Symbian Platinum Partner Program and Announces USBLink Host Software Solution," *Business Wire*, February 18, 2003.

Each was concerned about Adobe's per-font royalties and chose to bypass Adobe and develop their own technology. They collaborated on TrueType, which was included with both firms' respective operating systems in the early 1990s. Other times, though, the software platform has had to license technologies developed by others or acquire these firms in order to build in the feature promptly. That was the case with Internet Explorer, Microsoft's Web browser, which was originally built with technology that Microsoft licensed from Spyglass in 1995.

Of course, this means that making third-party add-ons for software platforms can be bit like making souvenirs for the latest Olympics. The market opportunities may be fleeting and might disappear as soon as the platform vendor catches up. The add-on developer can survive only by providing more value than the platform vendor. Many software products have disappeared after a short life when the platform vendor caught up.

One such example is the Watson browser add-on for Mac OS X. Watson, released in 2001 by Karelia Software, enabled users to access the Web in a novel and efficient way. Apple named it 2002's "Most Innovative Mac OS X Product." However, a year later, Apple's new Sherlock 3 search tool essentially supplanted it, and Watson was discontinued in 2004. Similarly, when ARM released MPEG-4 codecs for the Symbian platform in late 2000, the Symbian OS did not include that functionality. Newer versions of the OS, however, have come with support for MPEG media included, and ARM does not offer the codecs any more.

However, inclusion of a feature in the platform does not mean certain death for related third-party offerings. Third-party utilities have been able to distinguish themselves in various ways from the corresponding platform functionality. One such example is the Norton Utilities suite of disk and system repair and diagnosis utilities. Since its introduction in 1982, the Norton suite has had to adapt continuously as the functionality it offered has been added to both major desktop operating systems. Over time, disk compression, disk repair, defragmentation, disk optimizing, encryption, and other tools have all been added to both Mac and Windows platforms. Nevertheless, Symantec still offers Norton Utilities as a part of its SystemWorks bundle.

Versions and Options

Unlike automobiles and Chinese restaurants, software platforms tend not to offer different models or options. Consider the original Mac OS, introduced in 1984. It contained many fairly obscure features, but users and developers had to take all of them—or none. The same is true for the 2005 version. You can buy only one version of Mac OS X Tiger, and you cannot get it without Safari, the Apple browser, or Spotlight, Apple's search engine. Similarly, your Palm or Symbian-based smart phone will include a Web browser, a calendar, and an address book. You must take the option to play CDs on your Xbox or PlayStation 2 or to browse the Web via your i-mode phone even if you don't ever want to exercise it.

There are two exceptions to this pattern.[7] Software platforms for servers—the computers that serve as nodes and perform specialized tasks on networks (including "serving" content requested over the network)—often come in several versions for different uses. These versions are offered mainly to enable pricing to be tailored to differences in user requirements. More advanced versions are almost always supersets of the basic versions and usually come with better software support. The other exception concerns software platforms that are embedded in devices, such as cash-dispensing machines. Here, memory, performance, and power consumption concerns drive the market to small systems with targeted functionality. Often different versions target different sets of devices, or the manufacturer can pick and choose from the different components that make up the platform. Manufacturers can use utilities such as the Windows Embedded Studio to select components and build a system.

7. A third exception exists by order of the European Commission: since January, Microsoft has made two versions of Windows XP available to computer manufacturers. One makes Windows Media Player available to end users and makes the corresponding APIs available to developers; the other does not. Since they cost the same, no major computer manufacturer has licensed the second version. "Microsoft to Release Windows XP Home Edition N and Windows XP Professional Edition N in Europe," Microsoft press release, June 8, 2005 (http://www.microsoft.com/presspass/press/2005/jun05/06-08XPNEuropePR.mspx).

The Economics of Bundling

Software platforms differ from most other products in the number of features that are included in the product and in the growth of that number over time. But the difference is a matter of degree. Most products are bundles of components that could be provided separately and sometimes are. You cannot buy this chapter alone; you must buy the whole book. Men's laced shoes always come with laces, although it is possible to buy laces separately. Airlines cannot purchase the Airbus A380 without also buying the software system for flying it. And numerous products besides software platforms include features that others used to provide as add-ons.

"I haven't the slightest idea who he is. He came bundled with the software."

Many years ago automobiles were designed so that customers could purchase an air conditioning unit and have it installed after they bought the car. Today, almost all automobiles sold in the United States come with a factory-installed air conditioner. In the 1980s, PC users had to purchase an additional chip, a math coprocessor, if they wanted to do significant numerical calculations. By the 1990s these numerical capabilities had been integrated into all microprocessors used in PCs.

In all these cases, firms made two related decisions. One concerned product design and scope. What should be included, and how should the

parts interrelate? The other decision concerned the firm's product line. Should the firm offer only one product, or should it offer several with different combinations of features? Economists have examined these questions, as we describe in this section, and the answers help explain the scope and bundling patterns we observe for software platforms, as we show in the next section.

Product Design and Product Lines

Almost all products consist of components. Take something simple such as salt. Few consumers purchase pure salt. If you buy Morton's iodized salt in the United States you get salt, iodine, and a box. More complex products are combinations of even more components. The typical personal computer has hundreds of separate parts. A credit card provides two major distinct features: the ability to pay for things and the ability to finance those things.

These products could be designed differently—and historically were. Long ago salt did not contain iodine and did not come in easy-to-use containers. If you wanted a CD-ROM drive for your computer in the late 1980s you had to buy it separately and attach it with a cable. Charge and debit cards allow people to pay but not to finance.

To illustrate the decisions that firms make about how to design their products and what products to offer to consumers, consider a simple case in which there are two components, *A* and *B*, each valuable to consumers in its own right. The possible products are listed in Table 11.3. Three cases are particularly important:

- *Components selling* occurs when the firm offers *A* and *B* separately (cars and bicycle racks).
- *Pure bundling* occurs when the firm only offers *A* and *B* together as a single bundled product, *AB* (men's laced shoes).
- *Mixed bundling* occurs when the firm offers the bundle *AB* and either or both of its components, *A* and *B* (such as the Sunday *New York Times* and the *New York Times Book Review*).

With two components, there are three possible "products" and seven possible product lines, as shown in Table 11.3. The number of products and product lines increases dramatically as the number of components

Table 11.3
Products That Can Be Sold Based on Two Components

	A	*B*	*AB*
Components selling	×	×	
Components selling	×		
Components selling		×	
Pure bundling			×
Mixed bundling	×		×
Mixed bundling		×	×
Mixed bundling	×	×	×

increases. Thus, with three components there are seven possible products and 127 possible product lines, while with five components there are thirty-one possible products and over 2 trillion possible product lines.[8]

Firms make different decisions on product designs and product lines within the same industries. Some may offer only components, while others may offer only bundles, and still others may engage in mixed bundling. Consider the most popular midsize automobiles sold in the United States, the Ford Taurus, Honda Accord, and Toyota Camry. The Accord comes in six models that have between zero and two options. The Camry has three models with between nine and twelve options. And the 2004 Taurus had four models with between three and thirteen options. Across car segments there is even greater variation. For example, Porsche is famous for having an enormous number of options that allow purchasers to customize their cars. All of these automobile makers

8. Mathematically, the simplest way to formulate the general problem is to ask ourselves how many different subsets of K objects you can conceive. Line up the objects in whatever order you like: for each of them, there is a simple binary decision, to include it in the current subset or not, and one needs to make this decision for all objects. Note that modifying the decision for one object results in a different subset; therefore there is a total of 2 to the power K (2^K) different subsets. This includes, of course, the empty set, which is obtained by opting for noninclusion for each individual object; therefore the total number of distinct nonempty product lines based on K products is $2^K - 1$.

include tires on their cars. They all purchase tires from third parties, and none of these automakers sells tires separately.[9]

Minimizing Producer and Consumer Costs

Bundling decisions affect costs for both producers and consumers. In both cases it is useful to divide these into costs that vary with each unit (marginal costs) and costs that are lumpy over a range of units (fixed costs). There may be diseconomies of scope from producing multiple separate products that raise both sorts of cost.

For example, studies of automobile manufacturing have found that making many options available increases what are called "complexity costs," which do not vary much with sales. Similarly, maintaining and managing different SKUs (Stock-Keeping Units) costs money, regardless of sales volume. Separate products require separate packaging and shelf space, each of which raises costs. To offer multiple versions of its Linux distribution, Red Hat Linux would have to create distinct packages and probably obtain additional shelf space at software retailers to display all versions. Marginal costs may also rise with product variety. It is cheaper to produce and distribute one pill that contains both cold and headache medicine than two separate products. Likewise, it is less expensive for operating system vendors to distribute a single CD with both an operating system and Internet communication functionality (for example, support for TCP/IP protocols) than to distribute these separately.

It is also possible, of course, that combining features may increase fixed or marginal costs directly by making products more complex and harder to make. And complexity may have costly indirect effects as well, such as raising the likelihood of products breaking down, raising support costs for customers, and increasing the costs of repair. As software platforms have gotten larger, it has become harder to manage their production, the likelihood of bugs has gone up, since more modules interact with each other in ways that are difficult to anticipate entirely, and security problems have escalated. Likewise, combining drugs together

9. David S. Evans and Michael Salinger. "Why Do Firms Bundle and Tie? Evidence from Competitive Markets and Implications for Tying Law," *Yale Journal on Regulation* 22 (Winter 2005): 37–89.

increases the risks of unintended and unanticipated side effects. The marriage of computers and automobiles provides other examples of the potential disadvantages of bundling. Owners of Dodge 2001 minivans have, according to the *New York Times*, "posted anguished cries . . . about electronic gremlins that stop windows from rolling all the way up, that unexpectedly dim the interior lights, that drain batteries or that make engines sputter."[10]

Unless they dislike the components that are bundled, consumers are likely to realize savings from bundling. If you like to read about sports and arts every day, it is cheaper to get a newspaper with both than to have to buy two papers, even if you have to throw away the style section. And if you have both a cold and a headache, it is more convenient to get a single package of pills. Letting the producer make choices for you can save you time as well. When we go to the hospital for surgery, most of us would prefer to leave most of the choices of most of the components to the experts rather than make them ourselves. Downloadable music lets us pick individual songs for our collections. But many might prefer the bundles the artists and publishers put together and distribute as albums. Choice is costly because it takes time and effort to make informed decisions, ones that others may be able to do more efficiently, and bundling reduces consumers' transaction and search costs.

But bundling may also impose costs on consumers. Consumers may prefer to mix and match components—a common strategy in building home entertainment systems and increasingly popular for music collections. Although automobile manufacturers have reduced variety over time, many car buyers like having some choice, and some no doubt resent option packages that require them to take a moon roof to get a more powerful engine.[11]

These sorts of costs help explain how businesses choose the finite set of products they actually do offer from among the essentially infinite number they could offer. Firms must weigh the demand for a particular product offering against the costs of making it available as a stand-alone

10. "What's Bugging the High-Tech Car?" *The New York Times*, February 6, 2005, p. 14.

11. Example of an options package for Ford Taurus from Evans and Salinger, "Why Do Firms Bundle and Tie?"

product or as part of another product. Many products are not offered at all because there is not enough demand to cover the costs of producing and distributing them. Some men would no doubt prefer to get their shoes without shoelaces because they have a favorite shoelace brand or color they like to use. But there are probably so few shoelace aficionados that it would not pay to offer this option. Other products are offered only separately because few people want them as a system. Although this is changing, many families buy their own ingredients for dinner rather than prepackaged meals. And in other cases there is enough demand for the components and the bundle for producers to offer both—to engage in mixed bundling.

In some cases, it isn't profitable for producers to offer bundles versus the individual components. Consider a simple example. One hundred consumers would pay up to \$10 for *A*; fifty different consumers would pay up to \$6 for *B*, and a third group of ten would pay up to \$20 each for *AB*. It costs \$1 to produce each unit of *A* and *B* and \$2 to produce each unit of *AB*. It costs \$200 to make each of these three products available at all; these might be the fixed costs of creating and stocking any one of these products. In this case the average per-unit cost, if all demand is met, of *A* is \$3 (= \$1 + \$200/100), of *B* is \$5 (= \$1 + \$200/50), and of *AB* is \$22 (= \$2 + \$200/10).

Both *A* and *B* could be provided separately for a profit, since the consumer willingness to pay for each unit is greater than the average cost of producing it (\$10 vs. \$3 for *A* and \$6 vs. \$5 for *B*). However, the bundle cannot be provided profitably because the unit costs exceed what people will pay; it costs \$22 to make *AB* on average, but consumers will only pay \$20. The problem here is lack of demand. Not enough people want the bundle to make it profitable to provide, given the significant fixed cost involved.

On the other hand, firms sometimes offer *pure bundles* because, even though some consumers do not value portions of the bundle, it is cheaper to sell the components together. To see the intuition, consider the extreme case in which each of several types of consumers wants one component but none of the others. If the fixed cost of providing each of the components is high enough, it may nonetheless pay to combine them all together. It may be cheaper to give consumers a component they do not

want than to provide the component they do want separately. The manufacturer then saves money, and the consumer often gets a lower price than she would otherwise.

A simple example illustrates this. There are two consumers. Person 1 is willing to pay \$5 for *A* and nothing for *B*; person 2 is willing to pay \$5 for *B* but nothing for *A*. It costs the manufacturer \$2 per unit to make the components *A* and *B*. The per-product fixed cost of offering a product at all is \$1. The manufacturer could sell a unit of *A* and a unit of *B* separately for \$5 each, collect \$10 in revenue, incur \$4 in manufacturing cost and \$2 in product-offering cost, and make a profit of \$4. Or it could sell a bundle *AB* to both consumers for \$5 each, collect \$10 in revenue, incur \$4 in manufacturing cost and \$1 in product-offering cost, and make a profit of \$5.

Bundling is the best strategy in this example: it saves \$1 of fixed cost. In this example the manufacturer pockets the difference, but some of the cost savings would get passed on to the consumer in a competitive market. Moreover, if the fixed cost of offering a product were \$5, it would not be profitable to offer *A* or *B* separately (the additional \$4 in fixed cost would wipe out the profit of \$4), but it would be profitable to offer *AB* (the manufacturer would earn \$1 of profit).

Although these examples are contrived, they illustrate why firms offer only a fraction of the products—defined by combinations of components—that they could. The examples above involve just two components, for which there are three possible products. As we noted above, with three components there would be seven possible products (*ABC*, *AB*, *AC*, *BC*, *A*, *B*, *C*); with ten there would be 1,023. Even minimal fixed costs of offering individual products would encourage producers to reduce the number of products in their product lines to those for which there is significant demand. If you think about the products you buy, while you may have a great deal of choice, you have infinitely less than you could if firms offered all possible combinations of components that some customers might like.

Exploiting Demand

Firms bundle components because it enables them to sell more and usually make more profits. That can be true for demand-related reasons, as well as to save costs.

One obvious reason to add features to a product is to increase the demand for it. Perhaps surprisingly, this does not necessarily lead to a higher price. Speaking a bit loosely, it all depends on what sort of new buyers are attracted by the new features. Features that attract price-sensitive buyers—perhaps because they are particularly eager to save the cost of buying a separate product with those features—will tend to reduce the profit-maximizing price. Conversely, features that attract price-insensitive buyers will tend to raise the seller's profit-maximizing price. In the case of software, it is common for firms to add features without increasing the price. Since it introduced QuickTime in 1991, Apple has added many new features such as streaming audio and video, and support for new formats, yet Apple continues to offer the QuickTime player free of charge. Similarly, RealNetworks has added DVD playback and CD ripping to its player, which it still offers for free.

It is common to bundle together products that are complements, such as automobiles and tires, but firms may find that it pays to bundle products that aren't complements. We already saw an example of this above. Bundling persuaded two consumers to buy a product even though each wanted only a single component. This saved the manufacturer costs.

The idea that bundling of noncomplements can be used to enhance profits goes back to a classic paper by Nobel Prize winning economist George Stigler. Stigler tried to explain why movie distributors at one time required theaters to take bundles of pictures.[12] Suppose for movie *A*, theater 1 is willing to pay $8,000 and theater 2 $7,000; for movie *B*, theater 1 is willing to pay $2,500 and theater 2 $3,000. If the distributor rents the films separately, it would charge $7,000 for *A* and $2,500 for *B* to attract both theaters and collect $9,500 from each, for a total of $19,000. But consider how much the exhibitors would pay for a bundle of both movies: theater 1 would pay $10,500 and theater 2 would pay $10,000. Thus, if the distributor charged $10,000 for the bundle, it would collect $20,000 and make more money.

More generally, businesses can exploit the law of large numbers when they are producing products that have many components. Consumers place different valuations on the various features available to them. You

12. George J. Stigler, "*United States v. Loew's Inc.*: A Note on Block Booking," *Supreme Court Review* 152 (1963): 152–157.

value the arts section of the newspaper highly, while your spouse does not care much for it; your spouse values the sports section highly, while you do not care much for that section. The valuations for any component can be quite dispersed across consumers with different tastes. If you combine all these components into a single product, the variations tend to cancel each other out, and, relative to the corresponding average value, there will be less dispersion in the value consumers place on the product than on the individual components. This makes it easier for the firm to sell to a large fraction of the market at a price that captures a large share of the product's economic value.[13]

This of course means that many people are getting components that they do not value. But if it does not cost much to provide these components, if it costs little or nothing for consumers to ignore or dispose of these components, and if it is expensive to offer multiple product versions, bundling components together into a single product typically expands demand efficiently. These assumptions are especially likely to hold for software and other information goods for which the marginal cost of providing the product (and any component of it) is approximately zero, and the cost of developing and distributing the product is high.

Newspapers are a good example. They provide many features—from crossword puzzles to astrology tables, stock market quotes, and dance reviews—that only a portion of their readers care about. But, relative to the cost of producing and distributing a newspaper, these features are not that expensive to add. By including them the newspaper brings in more readers at its typical price, sells more copies, and therefore covers more of the fixed costs of producing the paper. Consumers who don't want to read these features can easily ignore them. Such bundling can benefit consumers by providing products that either would not be produced or would be more expensive absent bundling.

As is often the case, firms make money bundling this way because they are providing a service to consumers. Consumers get to pick and choose what they want. They can ignore choices they don't care about at little

13. For formal analyses, see Richard Schmalensee, "Gaussian Demand and Commodity Bundling," *Journal of Business* 57, no. 2 (January 1984): S211–S230; Yannis Bakos and Erik Brynjolfsson, "Bundling Information Goods: Pricing, Profits, and Efficiency" Management Science 45 (December 1999): 1613–1630.

cost. Few people care that their eyes may wander over horoscopes in the daily newspaper or that the paper weighs a bit more from the extra newsprint or that a software program takes up a smidgeon more memory because of code for a feature they'll never use.

Aggregating Demand

> Suppose that the first tenth of the population of 100 persons would be willing to pay \$10 for component 1, the second tenth would pay \$10 for component 2, and so forth up to component 10. Each would be willing to pay only \$2 for each of the other nine components. Costs are zero. If the firm sells each component separately, it could charge \$2 for each, sell all ten to all customers, and thereby make \$2,000. Or it could charge \$10 for each but sell each to only ten customers, and thereby make \$1,000. However, every consumer would pay \$28 ($\$10 + 9 \times \$2$) for the bundle of all ten components. By bundling, the firm could get all 100 consumers to buy the bundle and make \$2,800. Bundling this way can make consumers better off because they can get choices they wouldn't otherwise get. Moreover, producers of information goods can use this approach to cover the fixed costs of developing and offering products.

Bundling can be used in a different way to facilitate price discrimination, which we discussed in the preceding chapter.[14] That is, if different groups of consumers place different values on groups of components, bundles can be designed so that those with stronger demand pay more. The idea is possible to design bundles of components that cause consumers to sort themselves by the bundles they choose into groups with different willingness to pay. (Marketers call this "segmentation.") In the case of autos, some will want the car with the sports package, while others will want only the basic package. The seller can then charge a premium to groups that have a particularly high demand for a particular package and offer an especially aggressive price to consumers that are very sensitive to price but are also willing to take the no-frills deal. For this to work, there must be a predictable correlation

14. For a different potential use of bundling for price discrimination, see Richard Schmalensee, "Commodity Bundling by Single-Product Monopolies," *Journal of Law and Economics* 25, no. 1 (April 1982): 67–71.

between combinations of components and demand (for example, price-sensitive consumers generally have a low demand for frills). A number of studies have found, for example, that automobile companies have much higher markups on luxury models than on base models.[15]

Multisided Platforms

Bundling decisions by multisided platforms are particularly complex because they have to take into account all customer groups.

All of the considerations discussed so far still apply to multisided platforms. The principles just have to be adjusted to take into account the fact that there are several distinct groups of customers linked by indirect network effects. Newspapers—advertiser-supported media platforms—include style sections that appeal to younger women, who are valuable to advertisers. Video game console manufacturers may bundle joysticks not only because players want them, but because developers can produce cooler games if they know that all players will have joysticks.

Multisided considerations affect bundling decisions in three other ways.

Bundling customers. In some cases one can think of a platform provider as bundling customers together on one side to offer them to customers on the other side. Take shopping malls. Mall developers rent space to stores. But they are selective as to which stores they allow in the mall. They try to offer a diverse group of shops that match their intended customers. That means choosing particular quality levels of stores and limiting duplication. Most mall developers would reject a second bookstore even if it offered to pay the same rent as the first bookstore. That is in part because having greater diversity attracts more shoppers and therefore makes the mall more valuable to merchants, which in turn will pay more for more foot traffic. The same considerations apply to the content that mobile telephone operators offer to their subscribers and the types of articles that magazines offer their readers.

15. Steven Berry, James Levinsohn, and Ariel Pakes, "Automobile Prices in Market Equilibrium," *Econometrica* 63 (July 1995): 841–890.

Bundling negatives. Multisided platforms may also bundle components in ways that harm side 1 directly but create value to side 2 and, by attracting more customers on side 2, benefit side 1 indirectly. A shopping mall again provides a good illustration. You may have noticed that some malls, especially vertical ones, are not designed to minimize the amount of time it takes the customer to walk between stores. Instead, they are sometimes designed to increase the distance customers need to walk and therefore the number of stores they pass en route. That increases the foot traffic that passes by each store.[16] Payment cards are another example. Merchants that agree to accept cards from a system generally have to agree to take cards from all customers who present cards from that system and refrain from charging card customers more than customers who pay in other ways. Both rules impose costs on merchants and reinforce the bundling of all customer cards. But each rule benefits cardholders directly and merchants indirectly. (These rules have been challenged under the competition laws in various countries. In the United States, merchants can now take credit cards without taking debit cards, while in the United Kingdom, merchants can surcharge card transactions.)

Bundling for externalities. Multisided platforms pay particular attention to harvesting externalities among customer groups. Some features may be bundled because doing so promotes interactions between the two sides. Singles-oriented clubs often bundle drinks with admission; an example is the two-drink minimum. One explanation for this is that it promotes social interaction. Similarly, i-mode has signed a deal with Macromedia that enables it to include (bundle) the Flash Player plug-in in the platform it offers content providers, in order to encourage them to build Web sites with enhanced visual effects that are presumably more appealing to users. And, as we have noted before and will discuss in more detail below, including APIs in software platforms helps developers provide services to end users.

16. Beyard, Michael D, *Shopping Center Development Handbook*, 3rd ed. (Washington, D.C.: Urban Land Institute, 1999).

What Is Bundled, What Is Not; Why and Why Not?

As we have noted, the steady growth in the size of software platforms (as measured by lines of code) has been driven by the steady addition of features—most of which could have been provided by separate applications, and many of which once were—for developers and users. And customers don't get to pick and chose their features. Models are few and options are rare.

But it isn't as if software platforms have an irrational appetite, like PacMan, for absorbing everything in sight. The Sony Playstation doesn't have a word processor, although some of its customers might like to have one included. And Microsoft has kept Windows and Office for Windows separate, even though all Office customers need Windows too.

Then again, platforms' appetites do seem pretty voracious. As we mentioned earlier, i-mode phones can substitute for payment cards at stores in Japan. You just wave the phone over a sensor and press your thumb for further verification. The software for PDAs now helps people to make telephone calls, as well as to send emails and manage their calendars. Video game consoles and PCs are both racing to become media hubs that will help people manage and play music, videos, and television programs.

One reason software platforms have added more features is quite simple: they can.

Technology

Changes in the hardware for computing devices have made it possible to include more features in the software platform. The mobile phone operating systems of today, for example, simply could not fit in the memory that was available on mobile phones ten years ago. But advances in the underlying hardware have also created more opportunities for software platform vendors to devise features that can attract one or more groups of customers.

The cost of storage has declined for all computing devices. Figures 11.1 and 11.2 show the trends in the cost of random access memory (RAM) and hard disk storage for PCs. Between 1984 and 2005, the price per megabyte of RAM declined from $1,000 to less than 25 cents. Over the same period, the price per megabyte of hard disk storage declined from

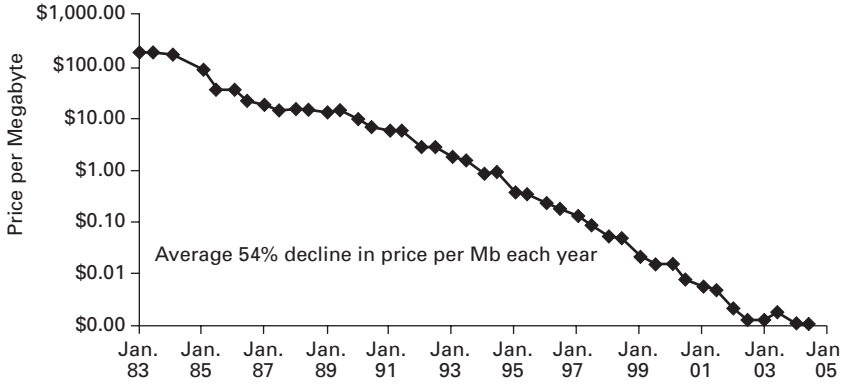

Figure 11.1
The price of hard disk storage in personal computers, 1983–2005, log scale. (Source: Data from 1983 to 2001 are from Steven J. Davis, Jack MacCrisken, and Kevin M. Murphy, "Economic Perspectives on Software Design: PC Operating Systems and Platforms," in *Microsoft, Antitrust and the New Economy: Selected Essays*, ed. David S. Evans [Boston: Kluwer, 2002]. Data from 2002 and beyond are from archived prices taken from compusa.com.)

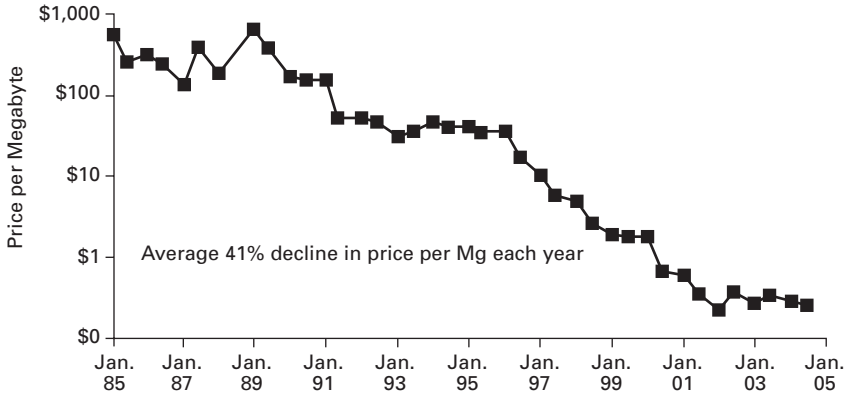

Figure 11.2
The price of random access memory (RAM) in personal computers, 1984–2005, log scale. (Source: Data from 1983 to 2001 are from Steven J. Davis, Jack MacCrisken, and Kevin M. Murphy, "Economic Perspectives on Software Design: PC Operating Systems and Platforms," in *Microsoft, Antitrust and the New Economy: Selected Essays*, ed. David S. Evans [Boston: Kluwer, 2002]. Data from 2002 and beyond are from archived prices taken from compusa.com.)

slightly more than $100 to less than a penny. Although other computing devices use different memory and storage components, they have all experienced similarly dramatic cost reductions.[17] For instance, the memory component used in many mobile phones, "NOR flash memory," fell 42 percent in price just between 2003 and 2004. Similarly the memory used in MP3 players, "NAND flash memory," experienced a 20 percent price drop over the same two years.[18] As a result, it is possible to put more code on the hardware and do more things with more RAM.

The price of processing power has also declined dramatically, as shown in Figure 11.3. Between 1993 and 2001 the price per million instructions per second (MIPS) declined from slightly more than $12 to less than 10 cents for Intel microprocessors. Today this cost has fallen to about a nickel for Intel's Pentium 4 chips. Similar changes occurred for other microprocessors. These changes have allowed software platform developers to provide complex new features that perform quickly enough to be of value to developers and customers. Many of the visual effects for today's video games could have been programmed in the early 1990s, but they would have had no commercial appeal then because games using them would have played too slowly.

Interactions among users of computing devices have also become easier and cheaper. Broadband connections have become cheaper and more widely available. The average cost of monthly DSL rental fell 30 percent, from $42 to $30, from 2000 to 2005.[19] Most significant businesses have broadband connections that facilitate wide-area networks of computers as well as connection to the Internet. The percentage of households with broadband connections has increased in most industrialized

17. Steven J. Davis, Jack MacCrisken, and Kevin M. Murphy, "Economic Perspectives on Software Design: PC Operating Systems and Platforms," in *Microsoft, Antitrust, and the New Economy*, ed. David S. Evans (Boston: Kluwer, 2002), fig. 1.

18. Takuya Inoue, Mario Morales, and Soo-Kyoum Kim, "Worldwide Flash Memory Forecast 2005–2008" (IDC report no. 32854), February, 2005.

19. Point Topic, "Long Term Trends in Broadband Pricing: 2000–2005," May 18, 2005 (http://www.point-topic.com/content/operatorSource/dslreports/Longtermtrendsinbroadbandprices2000-2005.htm).

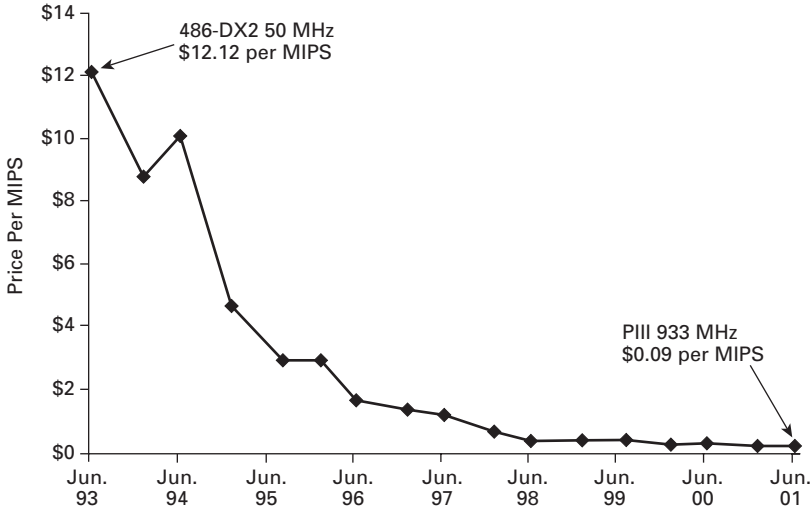

Figure 11.3
The price of processing power in personal computers, 1993–2001, Intel Processor MIPS. (Source: Steven J. Davis, Jack MacCrisken, and Kevin M. Murphy. "Economic Perspectives on Software Design: PC Operating Systems and Platforms," in *Microsoft, Antitrust and the New Economy: Selected Essays*, ed. David S. Evans [Boston: Kluwer, 2002].)

countries, as shown in Table 11.4, with almost one-third of U.S. households connected through cable or DSL in 2004.

In addition to these general trends, most computing devices have experienced decreases in the prices of other important components. The cost to computer assemblers of a CD-ROM drive declined from about $500 in 1991 to about $30 in 2005. The cost to mobile telephone manufacturers of a SIM card declined 25 to 30 percent from 2002 to 2003.[20]

The hardware and software platforms have a symbiotic relationship. These incredible advances in the hardware platform make it possible for the software platform to do far more than it could previously. Software platform makers see more capacious hard drives, faster microprocessors,

20. A SIM card is a component of most mobile phones that carries identifying information about the mobile customer as well as some address book information. "High-end cards and growing applications enable smart card manufacturers to leave behind a troubled 2002," *M2 Presswire*, August 6, 2003.

Table 11.4
Percentages of Households with Broadband Services

Country	2002	2003	2004
France	5.40%	11.00%	23.10%
Germany	7.20	10.20	14.80
Italy	4.10	8.60	17.00
Spain	5.90	13.60	22.40
UK	5.10	12.80	22.30
Western Europe[21]	8.20	12.70	20.40
US	15.70	23.10	29.90
Canada	28.10	35.80	42.70

Source: eMarketer, "Europe Broadband," April 2005.

and bigger broadband pipes, and think they can now develop new and improved features that consumers want. Hardware platform makers understand this. They recognize that it makes sense to invest in these hardware improvements in part because they can depend on software platforms, and the applications they support, to make use of the increased capability those improvements will produce. These positive feedbacks reinforce each other and lead to the addition of features—and tremendous innovation—through the various ecosystems based on computer hardware and software platforms.

Feature Accretion

One of Microsoft's lawyers once remarked, famously and flippantly, that he thought that Microsoft should be allowed to bundle a ham sandwich with Windows if it wanted to.[22] Why doesn't it—with mustard and pickles on the side, for that matter? Or, more seriously, why has it included Windows Media Player but not Office? On the other hand, why do some mobile phones, supported by their software platforms, come with cameras, email, instant messaging, and games, not to mention the ability to make a dinner reservation?

21. Austria, Belgium, Denmark, Finland, France, Germany, Italy, Netherlands, Portugal, Spain, Sweden, and the United Kingdom.

22. Todd Bishop, "Microsoft Loses Crucial EU Ruling; It Must Split Off Media Player While Appealing," *Seattle Post-Intelligencer*, December 23, 2004.

Most developers and users rely on only a portion of the features included in the software platform they are using. For example, any given software program typically would call on a small percentage of all the APIs provided by Windows. But different developers use different ones: game developers make much more use of the platform's graphics capabilities than developers of personal finance programs, for instance. Most consumers use only a few of the features included in Windows. Have you ever used the on-screen keyboard? Or explored the fonts installed on your system using Character Map? We haven't, and we suspect few others have either.

Similarly, few individuals read newspapers from cover to cover. Most pick and choose articles that interest them. The most popular section of U.K. newspapers, sports, is read by fewer than half of newspaper readers, while most sections are read by fewer than a third.[23] Some of us never read the sports section, while others never read the marriage announcements or obituaries. Similarly, a typical basic cable television package comes with seventy channels,[24] but we suspect most people watch a handful or two and ignore all the rest. There probably isn't much overlap between the regular viewers of Comedy Central, Home and Garden TV, the Discovery Channel, and ESPN2. This isn't surprising. As we noted above, bundling different things together is a particularly good business strategy for information goods, for which the marginal cost of adding and distributing another feature is typically very small.

There is a simple explanation for the steady feature accretion we see in software platforms. Technological advances in hardware have given software platforms more to work with and have all but eliminated hardware-related constraints on their size, at least for the time being. Software platforms add features in the hope that more users will find the platform worth its price because they can find the particular features they want and that more developers will write to the platform because they

23. PR Week Media Snap—"National Newspaper Readership Patterns," *PR Week*, May 10, 1990.

24. Comcast standard cable package comes with seventy channels in the Boston area, RCN "Full Basic" comes with seventy-five channels (http://www.comcast.com/Support/ChannelGuide.ashx); http://www.rcn.com/cabletv/lineupDetails.php?lineupID=13.

can find some subset of APIs that helps them write profitable programs. Because platforms are multisided businesses, these additional users and developers increase demand indirectly as well as directly. A platform is more attractive to end users if it has more applications, and it is more attractive to developers if it has more users.

But if more is always better, why no ham sandwich? The answer lies in comparing the additional consumers brought in by adding new features with the cost of adding them. Take Office. There were 63 million Microsoft-licensed copies of Office in use in 2004.[25] The average price of an Office upgrade was upward of $250, and a new license for Office for businesses was upward of $350. And there were about 515 million Microsoft-licensed copies of Windows in use in 2004. Users interested in upgrading their version of Windows can purchase the latest edition from retailers—an upgrade of XP is $90 on average, or a new license is upward of $150.[26] Many business customers do not need Office on their computers because they use specialized software. Insurance company employees, for example, typically spend their days using customized software for dealing with claims and other insurance-specific matters. And many households do not need the firepower in Office either. By keeping Office separate, Microsoft can charge companies that do not need word processing and the other Office features a lower price and companies that do need those features a higher price. In this case the fixed cost of offering separate products is probably fairly small relative to the additional profits that can result from selling Office and Windows separately. But what if most customers who bought Windows also wanted Office? Then it might well make sense, in terms of both Microsoft's profits and total cost to society, to bundle Windows with Office.

This example highlights the fact that the features that get bundled into software tend to be of two extreme sorts. It makes business sense to bundle features that are used by relatively few users, as long as those

25. http://www.microsoft.com/msft/speech/FY05/Raikes_CaposselaFAM2005.mspx.

26. Survey of Office products on Amazon.com; AI Gillen and Dan Kusnetzky, "Worldwide Client and Server Operating Environments 2004–2008 Forecast: Microsoft Consolidates Its Grip" (IDC report no. 32452), December 2004, tables 1 and 2; survey of Windows products on Amazon.com.

users value the features in question highly enough, because it will generally cost little to add these features relative to the additional sales brought in. (Of course, at the far extreme we have features that are simply not worth developing in the first place because so few end users care about them that there is no way to cover their development cost. We are ignoring those features here—as software platform vendors try hard to do in practice.) And it makes sense to bundle features that are used by most users. If most users want a calendar with their PDA, there is nothing to be gained by incurring the extra cost of selling it as a standalone product.

Between these two extremes, it could make business sense to offer the components separately or to offer multiple versions, some of which don't have certain features. Looking across software platforms, however, it appears that this sort of mixed bundling is seldom used. Software platforms either include a feature in the platform or they don't. Unlike cars or cereals, there are almost never multiple versions of the platform to choose from. (Of course, as with everything, this can change with developments in technology [innovations in making modular software, for example], consumer demand [segments develop that want a specific feature set], and competition policy [some competition policy authorities have argued that Microsoft should offer multiple versions of Windows].)

The multisided nature of software platforms helps explain this. Users want to know that the applications they license will run on their version of the software platform, while developers want to know that their applications will work for customers who have the software platform to which they are writing. This assurance is particularly important, since developers and end users are making decisions at different points in time. If there are multiple versions of a software platform on the market, the developer may not be able to conduct the advertising necessary to tell you which is the right version. Thus, standardizing software platforms tends to help both end users and applications developers.

Most commercial vendors of Unix have made their versions proprietary, and more than twenty versions are currently available. Applications written for one version might not run on another. Given its roots in Unix, Linux has been particularly careful to prevent similar fragmentation or

"versioning" of its software platform. The GPL prevents Linux vendors from appropriating the source code to build a proprietary version of Linux (or for any similar purpose).

Innovation Through Bundling

One only has to take a look at ads for Apple's Mac OS X Tiger to see that a major source of innovation in software platforms comes from bundling new features. Although Apple has made lots of improvements to the core of this operating system over the years, consider what it is highlighting for consumers in Figure 11.4: a search program (Spotlight), a Web browser (Safari), video and audio conferencing (iChat), a media player (QuickTime), and an email program (Mail). All in all Apple says there are more than 200 new features in Tiger.

Some of these features were included in previous versions of the operating system but have undergone considerable improvement. Others are new, such as Spotlight. And while products similar to each of these features are available from independent application developers (Firefox for browsing and RealPlayer for media, to take two examples), Apple users benefit from having these features available to them as part of a single integrated platform. Consumers, for example, don't have to find and install their own browser, media player, and email client. They can just trust Apple to provide a good package. And a number of reviewers have commented that these additional features make Tiger an innovative and desirable software platform. They claim that "even casual Mac users will immediately see the difference,"[27] because "Tiger is the best version of Mac OS X yet. . . . The performance improvements are immediately noticeable. Every major bundled application has been improved. There's an unprecedented number of substantial, totally new features and technologies: Spotlight, Core Image and Video, Quartz 2D Extreme, Dashboard, and Automator, just to name a few."[28] Of course, if an Apple-supplied feature is a dog, end users always have the option of ignoring it and using something else.

27. "Apple Mac OS 10.4 Tiger," *Cnet Review*, April 29, 2005 (http://reviews.cnet.com/Apple_Mac_OS_10_4_Tiger/4505-3673_7-31256837-2.html?tag=top).

28. http://arstechnica.com/reviews/os/macosx-10.4.ars.

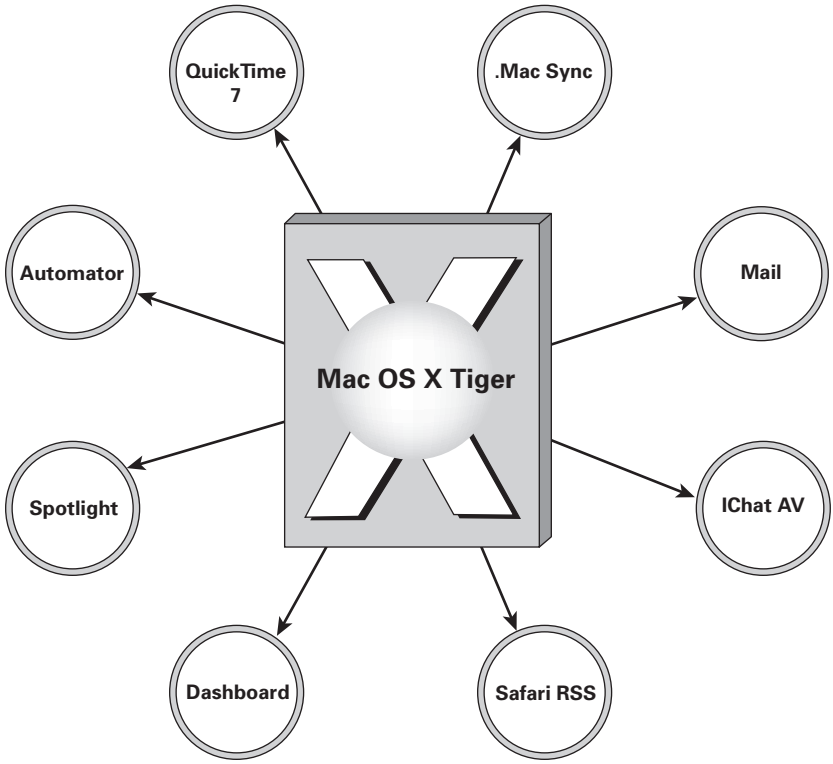

Figure 11.4
Diagram based on a screen shot of Apple's online promotion of Mac OS X Tiger. (Source: http://www.apple.com/macosx/.)

Mac OS X Tiger doesn't just provide innovative new features to end users. Apple marketing highlights the new features it has provided to developers. For example, Apple's QuickTime 7 technology "features an ultra-efficient new video codec . . . that delivers stunning video quality,"[29] while Core Image "unlocks the performance of today's powerful graphics hardware for ultra-fast, pixel-accurate image processing."[30] Reviews geared toward developers have also noted the value of these additions. Reviewers have described Mac OS X Tiger as "a milestone in Mac OS

29. http://www.apple.com/macosx/features/quicktime/.
30. http://www.digitalhub.com/macosx/overview/advancedtechnology.html.

X's development process."[31] This system has some developers saying things like "being a Mac developer was a fun and rewarding experience before Tiger, but now with all of these new technologies, our jobs got even easier."[32]

Comparing the Mac OS 7, introduced in 1991, to the Mac OS X Tiger, introduced in 2005, highlights the pattern of technical advance. Table 11.5 lists some of the features added during this period. Some of the things you couldn't do in 1991 but could do in 2005 were the result of other information technology innovations. Thus, you couldn't browse or stream audio and video in 1991. But today, not only can you do those things, you can do them "right out of the box" with your new Macintosh, without buying any applications. For most people that's a benefit. Other things could have been done in principle in 1991, but no one had thought of them or didn't know how to do them very well, or there was no use for them. Sophisticated searching was unnecessary, for instance, since few people stored a large number of documents or multimedia files on their PCs. The sophisticated compression technology that enabled streaming media did not exist until the mid-1990s, and even then, people did not realize how popular streaming media would become.

These same sorts of observations could be made for any software platform. Although bigger isn't always better, the growth we documented at the beginning of the chapter has enabled users and developers—and, we should note, makers of hardware and peripheral equipment—to do more with their software platforms.

Convergence

There seems to most of us to be a qualitative difference between Palm OS including a browser and DoCoMo turning its phone into a payment device. The former seems like a natural expansion within a relatively well-defined category, while the latter seems like one category setting out unexpectedly to conquer another. But there is really nothing unexpected about it. Increasingly, computer platforms—sometimes led by the software platform, at other times working hand-in-hand with the software platform—have invaded nearby, and not so nearby, categories.

31. http://arstechnica.com/reviews/os/macosx-10.4.ars.
32. http://maczealots.com/reviews/tiger/developers/.

Table 11.5
Some of the New Features Added to Mac OS Since System 7

Multimedia functionality (QuickTime and iTunes)
DVD support and recording
Email and Internet functionality
Disk and Internet searching
Java support
Handwriting recognition
Stickies—Post-It-like application
Bluetooth
Power management
Better disk management
Encryption
Support for multiple users
Password management, voice passwords
Modern multiprocessing

Mobile telephones, for example, are starting to compete with digital music devices to download, store, and play music. And increasingly these devices are able to play television—for now, specially designed short soap operas—to help entertain subway and train commuters, among others. We've gotten used to seeing them used for email and instant messaging.

Game consoles are competing with other home entertainment technologies. They include DVD drives and therefore compete with manufacturers of DVD players; they have the ability to store, manage, and play music, and thus compete with a variety of music-related devices; and they can download, store, and edit television, and are therefore competitive with products like TiVo.

The fact that computer platforms can combine all these features does not necessarily mean that consumers will embrace them. Many fortunes have been lost by those who believed the 1990s' hype about digital convergence—witness the AOL Time Warner merger. Companies like Microsoft have been working on ways to get PCs into the living room for more than a decade. Thus far consumers seem to like their PCs and their home entertainment systems to be separate boxes in different rooms. Yet convergence may occur slowly and by stealth.

The home entertainment system of the future, like the automobile of the present, may not look or feel like a PC. But at its heart it is likely to

have a microprocessor and a software platform. It may look like an obvious product, but in fact it will probably be the result of the accretion of features in various software and hardware platforms over time. Bundling drives innovation and creates industries.

INSIGHTS

- The ability to select bundles of features to sell helps firms segment their customers, control costs, and enhance profits. Bundled products offer consumers convenience, lower costs, and products tailored to their needs and wants.
- Bundling decisions by multisided platforms, such as software platforms, are more complex since they must take into account the effect on all customer groups. Multisided businesses must consider both the additional customers they get on one side as a result of including a new feature and the additional customers they will get on the other side from having those additional customers. They may also include features that harm one side directly but benefit the platform overall by getting more customers on board on another side.
- Bundling makes sense for businesses whenever the cost of adding additional features is lower than the additional sales generated thereby—even if most purchasers do not value or use all the features in a product bundle.
- Software platforms double in size roughly every two years mainly as a result of adding new features; all software platforms have attracted new users and innovated in this way. This behavior is a response to demand that has been made possible by the plummeting costs and rapidly increasing capabilities of computer hardware.
- Software platforms tend not to offer models or options. They come bundled with features that users and developers have to take in total, even if those features are not widely used.