12

Swept Away

Look to the future, because that is where you'll spend the rest of your life.
—*George Burns*[1]

INSIDE THIS CHAPTER

- Why software platforms create, destroy, and transform industries
- The enormous potential of Web-based software platforms
- How auction-based and search-based platforms are transforming the retail sector

Introduction

Software platforms march relentlessly across the economic landscape, forming new industries, transforming others, and sometimes shattering old ones in their wake.

Video game platforms have gradually pushed board games to one side. In 2004, people spent twelve times as much on video games as they did on board games; twenty years earlier they had spent more than seven times as much on board games as on video.[2] The PC software platform made the typewriter a museum piece within a generation. In 1985 the

1. http://en.wikiquote.org/wiki/George_Burns.

2. http://money.cnn.com/2005/04/29/news/midcaps/bored_games/; "The Profit In Games People Play," *New York Times*, December 31, 1986; "Video Game Wars Heating Up: Firms Hawk New Generation of Machines," *Los Angeles Daily News*, August 24, 1993; Schelley Olhava, "Worldwide Videogame Hardware and Software 2004–2008 Forecast and Analysis: Predicting the Future" (IDC report no. 31260), May 2004.

typewriter industry had U.S. sales of $1.1 billion; about two decades later it was less than a tenth as large.[3] The destruction isn't confined to old industries. Fax machines started a global revolution in real-time communication in the 1980s. Slowly, email and other document delivery platforms are making the fax machine passé. Nor are software platforms protected from their brethren, as the Palm OS learned from the platforms that have powered smart mobile phones. And, like amoebae, software platforms envelope their own. Regularly, they add features such as Internet communication capabilities that had been provided by stand-alone applications.

The malleability of code makes software platforms particularly adept at moving into old industries and starting new ones. Sometimes this is as simple as adding a block of code to an existing software platform—more or less like adding another section to a newspaper. For example, the Safari browser that Apple added to the Mac OS took up just over 6 megabytes of this 500+ megabyte software platform. Other times it requires more substantial work. Yet both Symbian for mobile phones and MS-DOS for PCs were completed in less than a year by building on code that was created for other purposes, much as a writer might build a novel from a short story.

Creative destruction has been a hallmark of economic progress for millennia, but it has proceeded at a glacial pace for most of history. The Industrial Revolution sped this process up. Even so, it took decades for change to filter through the economy following innovations such as the spinning jenny, steam engine, and electric generator. The information technology revolution has quickened the pace of industrial change greatly. The plummeting costs of computer processing and storage make it possible to create products and industries that were not only infeasible but also unimaginable a few years earlier. Software platforms further accelerate the process of creative destruction, mainly because code is digital and malleable. Think how easy it is to add a new feature to a software platform and distribute that change electronically over the Internet to potentially billions of computing devices around the world.

3. "What's New in Typewriters," *New York Times*, March 9, 1986. "Clack of Typewriter Still Stirring Minds," *USA Today*, June 20, 2003.

As we look ahead, there are signs that the pace at which software platforms transform the economy will accelerate even further in the coming decades. People have speculated about some of these transformations for years.

The control of the living room is perhaps the best example. As of 2006 there is little doubt that software platforms in some form will dominate home entertainment. Many of these platforms have crept into the television ecosystem without much notice.

People usually think of TiVo as a digital video recorder (DVR) maker: it makes a box that records television shows and skips over commercials. In fact, the hardware in that box is built around a hard disk drive and didn't require much inventive effort. The software platform is the secret behind the TiVo service and the strategy that this company has adopted for home entertainment.

To make inroads, TiVo followed strategies with which we are now familiar. It priced the box low to penetrate the market and earned revenues from subscription fees. At first this was a familiar one-sided "give away razor and sell the blades" strategy. However, this pricing approach was designed to create a critical mass of TiVo users. Once developed, those users could be used to attract two other sides.

One is familiar: developers. TiVo is evangelizing its software platform by providing tools and offering prizes for the best applications in several categories, including games, music, and photos.

The other side is perhaps surprising: advertisers. As a verb, "to TiVo" has entered American slang as the process of skipping over commercials—not a development likely to thrill the companies that spend billions every year on television advertising. As a software platform, however, TiVo provides tools that allow television advertisers to provide creative services to users. Viewers can select advertisements they are interested in and can download infomercials and other more detailed product information that they can't get in a 30-second spot. Whether TiVo will succeed with this strategy is not obvious as of this writing. (It is facing stiff competition from low-priced DVRs offered by cable companies.) But using the invisible engine in its DVR, TiVo is trying, at least, to transform the television advertising model, and with it how we consume home entertainment.

Automobiles provide another instructive example. Although we doubt software platforms will transform the auto industry, they will surely take on a more significant role in automobiles. Microprocessors now control more than thirty mechanical systems in automobiles, from power windows to antiskid systems, as well as features that provide entertainment and information such as radio and navigation.[4] Many different operating systems, some of which are dedicated to particular microprocessor-based features, run these computing devices.

The dashboard now comes with separate technologies that are ripe for integration through a software platform. The navigation system, for example, is currently a self-contained computer application. Many developers have created map-based applications on the Web. Drivers will soon relish the ability to download these applications onto their automobiles. At the same time, automobile makers are increasingly incorporating new entertainment technologies into cars, including the ability to use iPods, MP3 players, and satellite radio. These create a demand for a more flexible software platform that can facilitate in-car entertainment. In addition, consumers are interested in more wireless speech-enabled applications in their cars, which is a natural for a software platform.

Not surprisingly, several vendors have been working on developing in-car software platforms. Enea, a Swedish company, for example, offers three operating systems for automobiles. These operating systems share a common set of APIs and environment for writing applications. As of late 2004, automobile companies wrote customized applications to run on in-car software platforms in their vehicles. Over time we would expect more third-party developers to produce applications for these platforms. Moreover, cars are increasingly becoming connected computing devices. One source estimates that sales of global positioning systems (GPSs) have increased 33-fold between 1998 and 2004, and about 60 percent of consumers plan to purchase a GPS system with their next vehicle.[5] More-

4. http://www.epn-online.com/page/18332/embedded-software-platform-for-automobiles-the-replacement-of-mechanical—.html.

5. "Microsoft bolsters auto application software," *Network World*, July 18, 2005. "Enea Introduces Embedded Software Platform for Automobiles," *Business Wire*, December 2, 2004. "Cost of Getting Lost Is Higher Than Ever," *PR Newswire*, May 19, 2005; "You are here," *The Virginian-Pilot & The Ledger-Star*, December 20, 2004.

over, the increased use of wireless technology with cars through mobile phones could enable Internet access and the integration of in-car software platforms and Web services into the dashboard.

Enterprise software is another example of an industry that software platforms will change. Oracle, Microsoft, SAP, and other software developers have written large, complex applications that large enterprises use to handle tasks such as human resources, accounting, and supply chain management. Globally, enterprises spent more than $21 billion on such software in 2004.[6]

Enterprise software is evolving into middleware platforms that themselves support a developer community. Rather than adding their own features to their applications, the large enterprise software makers are making the system services provided by various modules in their products available to third-party developers. That includes publishing APIs and providing software developer kits to help other companies develop applications. SAP, for example, announced in early 2005 that it was making its Netweaver development platform publicly available. That platform includes more than 1,200 services that are available through APIs that developers can use. Later in the year Oracle introduced its competing Java-based Fusion middleware. Both companies are making developer tools available for free to actively encourage developers to "get on board" the platform. SAP includes the core components of Netweaver for free with its mySAP ERP and mySAP Business Suite while Oracle offers Fusion's Java IDE, called JDeveloper, for free.[7]

Over the next decade invisible engines will transform economic life well beyond our living rooms, cars, and offices. They will change how we buy and pay for things. And they will cut a wide swath of destruction across many industries that have heretofore helped buyers and sellers find and do business with each other.

The market capitalization of the newspaper industry provides a leading indicator of what's to come. From December 2004 to December

6. Paul Hamerman and R"Ray" Wang, Forrester Research, "ERP Applications —The Technology and Industry Battle Heats Up," June 9, 2005.

7. http://www.zdnet.co.uk/print/?TYPE=story&AT=39192233-39020466t-20000007c; "Oracle JDeveloper 10*g*," http://www.oracle.com/tools/jdev_home.html; Ellen O'Brien, "NetWeaver for Free? Not Quite," *SearchSAP.com*, September 30, 2004.

2005, each of the ten largest newspaper companies in the United States lost market value; their market capitalization fell 23 percent over this period, while the S&P 500 rose by 4 percent. New York Times Company's market value declined by $2 billion (35 percent), and Gannett, the largest, lost $6 billion (29 percent).[8] A major factor behind this is Internet-based advertising. And behind Internet-based advertising search engine–based software platforms that are using multisided strategies to drive the growth of vast ecosystems. The source of this cataclysmic change is the subject of this concluding chapter.

Web-Based Software Platforms

Many of the software platforms we have discussed are married to a hardware platform. That's the case with PC and video game console platforms. Being a couple enables the software and hardware to play as much as possible to each other's strengths. Specialists talk about "optimizing" the software for the hardware. Other software platforms can't indulge in monogamy. They have to work well with several hardware platforms because the industry is fragmented along hardware lines. That's the case with mobile phones. It became the case for PDAs as software platform makers such as Palm realized that mobile phones provided both opportunities and challenges. A few others are "distributed platforms." Digital media platforms and i-mode have pieces that reside on the devices that people use (often called the client) and other pieces that reside on computers that sit in the backroom (often called the server).

It has always been possible to design a software platform that resides entirely on servers. Indeed, one of the businesses we discuss in the box below, payment systems such as Visa, are server-side platforms. The software platform resides on servers on a network and does virtually all the work necessary to execute transactions. The clients—payment cards with magnetic stripes and terminals at the point of sale—do little (like the computer screens called "dumb terminals," that long ago connected people to mainframes).

8. Pulled from Bloomberg on December 28, 2005 (http://finance.yahoo.com/q?s=%5EGSPC).

Two related developments have made server-based software platforms increasingly attractive. For one, the World Wide Web has grown dramatically. The number of computers sold for use as Web server activities grew 60 percent from 2000 to 2005.[9] For another, communications capacity has grown: more households and businesses globally have broadband connections, and those connections have increasing rates of throughput. As we've noted, this piping has extended to mobile telephones and other handheld devices. Not surprisingly, businesses have developed software platforms that live on the Web.

As consumers, we tend not to think about the code that lies at the heart of many Web-based enterprises. Yet, putting aside whether this code comprises a software platform, it is a major source of value for companies like Amazon. Instead of building factories, they construct software programs that process information on mainframes or, more commonly, massive arrays of server computers. They innovate through adding features in their software, such as Amazon's feature, "Customers who bought this book also bought. . . ."

Most Web-based enterprises do not, in fact, operate what we've called software platforms. They run computer applications that, while innovative and admirable, do not have the defining feature of a platform: the provision of software services to third parties. But others have followed multisided strategies by making software-based services available through APIs and encouraging the growth of developers and other third-party complementors.

This chapter focuses on two software platforms that facilitate transactions on the Web: eBay and Google. Before we discuss them in detail, we need to take a quick, relatively painless detour into economic history.

Driving Down Transactions Costs

Economic textbooks often describe an idyllic world in which everybody has perfect information for free. Pricing is transparent. Quality is known. Search is costless. With intense competition in such a world, resources

9. IDC Server Workload Data, 2005.

flow effortlessly to their highest-valued uses. Adam Smith's "invisible hand" of self-interested behavior inevitably leads to the greatest good.

Like the frictionless plane of introductory physics class, this economic nirvana is unreachable. Yet over centuries, societies have gotten closer to it through the development of institutions that facilitate trade among people and businesses. We have grown so accustomed to some of these institutions that we forget how innovative they were in their times, and the enormous values they created for humankind.

In the Western world, some of the most important innovations of this sort date to very ancient times. The Lydians introduced the first money—standard gold and silver coins that were an easy way to exchange and store value—in the seventh century B.C. The Babylonians held the first auction for which there is a record, in around the fifth century B.C.[10] The village market that brought buyers and sellers together in a central location had been around for millennia before the Roman Forum opened its doors to traders.

Further innovations that facilitated buying and selling arose as Europe came out of the dark ages. The development of checks in twelfth-century Florence was the major monetary one. Stock exchanges—bourses—began several centuries later. The predecessors of today's modern exchanges were firmly in place by the first part of the nineteenth century. The invention of printing with moveable type in 1450 vastly increased the rate at which information could be disseminated throughout the world. London newspapers printed advertisements regularly by the seventeenth century.[11]

During the early nineteenth century, at least in the United States, retailers started allowing customers to buy on credit. People could charge their purchases and pay at the end of the month. Many larger retailers offered installment plans that allowed people to buy durables, such as sewing machines, over time. In 1950, Diners Club—the first payment card that could be used by individuals at many merchants—gave birth to the modern global industry of debit, credit, and charge.[12]

10. "Auctions . . . A History" (http://www.jjmanning.com/selling.htm).

11. *Encyclopedic Dictionary of Semiotics, Media, and Communication*, edited by Marcel Danesi, 2000.

12. David Evans and Richard Schmalensee, *Paying with Plastic*, 2nd ed. (Cambridge, Mass.: MIT Press, 2005).

Over the course of economic history there have also been many innovations in transportation. These have ranged from the early traders, who traversed the known world, to the development of shipping and rail, and eventually automobile and air transportation. The world became a more connected place. We had a global economy that allowed financial and physical resources to move relatively freely well before the birth of the commercial Internet in 1995.

All these innovations have lowered transactions costs and thereby made exchange cheaper and broader. The Lydian invention of coins, for example, made trade cheaper and more secure. It also permitted trade that might not have taken place because buyers or sellers were more likely to have a common medium of exchange and unit of account.

Consider a simple modern-day example. I have a car I am willing to sell so long as I get more than $10,000. You are willing to pay up to $12,000 for my car. Suppose it costs $500 between the two of us to find each other and consummate the deal. Then we can share in $1,500 of value that is created by moving the car from me (the lower-valued user) to you (the higher-valued user). How much we each capture depends on our bargaining power and the sales price we negotiate. But in the end there's $1,500 of hard value to be had net of transactions costs.

Institutions can help consumers and businesses obtain value through exchange in two major ways. They can make it cheaper. If you and I could find each other and consummate our deal for $100 less, we'd have $100 more value to share. They can also make trade that was impossible possible. Suppose you and I couldn't find each other and do a deal without a particular institution—an auction, an advertisement, or a payment card. Then we might not be able to obtain the $1,500 of value at all, or we might end up with less desirable trading partners and obtain less than $1,500.

Summed across all people and businesses and all possible transactions, innovations that can reduce transactions costs or make more trades possible have immense value. A little arithmetic reinforces the insight. An innovation that could reduce the cost of retail transactions in the United States by 0.1 percent of the value of those transactions would result in savings of almost $4 billion annually.[13] That's one of the reasons

13. http://www.census.gov/svsd/advretl/view/adv44x72.txt.

payment systems compete to shave off seconds on transactions at the point of sale.

Payment Cards

The payment card remains both an important vehicle for reducing transactions costs and an intriguing computing device. Even in its magnetic stripe form it provides a primitive interface with a vast global computer network that allows people to obtain cash and conduct other transactions around the world. It is also the most popular computer-related device in the world. There were more than 2.2 billion payment cards in circulation in 2004 compared with 1.5 billion mobile phone subscribers.[14]

The technological basis for making the payment card the most widely available computing device in the world has already been laid. Smart cards, which contain a computer chip with considerable storage capacity, have held the promise of replacing the magnetic stripe technology since they were invented in the early 1970s. Yet virtually all cards in the United States remain based on magnetic stripes. Smart cards are more popular in Europe—they were introduced in France in the early 1990s, and MasterCard and Visa have recently provided strong incentives for all banks in Europe to deploy the next generation of these cards.[15]

Although their considerable hardware intelligence remains largely untapped, it is not for lack of an operating system. The major card networks have sponsored software platforms for smart cards that could promote the development of applications for these cards. MasterCard adopted the MULTOS operating system in 1997, while Visa uses Sun's Java Card, also developed in 1997, and American Express uses both.[16] These software platforms provide software services through APIs and make it possible to create applications that will run on the smart cards' microprocessors. So far, however, relatively few such applications have been created.

Smart cards have not succeeded in generating the sort of indirect network effects between the hardware and software that has helped other platforms overcome their chicken-and-egg startup problems and grow quickly. They have faced several problems.

14. Alex Slawsby, and Allen M. Liebovitch, "Worldwide Mobile Phone 2004–2008 Forecast Update" (IDC report no. 31080), July 2004. Source: *The Nilson Report*, no. 829, March 2005.

15. http://www.cartes-bancaires.com/EN/groupement/historique.html.

16. http://www.javarss.com/java-timeline-10years.html; Kim Min-hee, "MasterCard Takes on Visa in Smart Card," *The Korea Herald*, November 15, 2004; Donald Davis, "Brand Awareness: The Four Big Payment Brands Are Counting on Contactless Chips to Inject Some Excitement into Their Smart Card Programs," *Card Technology*, March 1, 2005.

(continued)

While reductions in microprocessor costs have lowered the cost of smart cards over time, these cards are still significantly more expensive than magnetic stripe cards. Without a killer application that some group—merchants, cardholders, issuers, or other players in the system—values highly, there is little demand for smart cards in the United States. (In Europe they serve to provide enhanced security that is provided in other ways in the United States.[17]) On the other side of the market, without a base of smart cards there is little incentive to develop applications for them, and no killer application for smart cards has emerged to spark their widespread adoption. One possible killer application that we will return to below is "contactless"—these are chip-based cards that use radio waves to connect to a reader at a short distance.

The payment card industry has faced a problem similar to one experienced by mobile telephone producers but even more severe. Banks issue credit and debit cards. Although card associations such as MasterCard and Visa can encourage the development of applications that run on all cards, each bank has a strong incentive to differentiate the cards it issues from those issued by other banks. The cards, after all, are part of the service that banks are providing their depository customers, in the case of debit cards, or often part of a lending relationship, in the case of credit cards. Banks are interested in applications that help them sell these broader relationships in competition with other banks. Unfortunately, that reduces their incentives to promote the development of applications. The card issuers have a further incentive for maintaining a walled garden around their devices: security. Either the card issuer or the cardholder faces significant liability if the card is breached. Two developments could provide the indirect network effects necessary for supporting a software platform–based ecosystem for payment cards. The first involves the marriage of two popular computer devices—the payment card and the mobile phone.

In Japan, NTT DoCoMo has incorporated the ability to pay for goods and services in some of the mobile phones it provides subscribers. It relies on Sony's FeliCa contactless chip. Consumers scan items at merchants that have contactless readers that can communicate with the phones. Based on an early 2005 report, there were a million phones with this payment ability and 13,000 merchants with readers.[18] In Japan, most of these bricks-and-mortar transactions are going through the DoCoMo billing system in addition to the Internet billing system we discussed in Chapter 7. Of course, Japan is one of a kind in so many ways, as we have noted—DoCoMo's success as a payment system partly results from the fact that payment cards were much less popular than cash; DoCoMo got people paying by phone

17. Evans and Schmalensee, *Paying with Plastic*.

18. Matt Richtel, "Momentum Is Gaining for Cellphones as Credit Cards," *New York Times*, January 10, 2005.

(continued)

before it got them to pay by card in effect. And DoCoMo is the largest mobile phone operator in Japan.

The other development is contactless payment as a possible killer application that could appeal to cardholders and merchants. Chip-enabled cards have not become popular because no one had figured out a way to generate significant value for customers that would warrant the initial cost. With few chip-enabled cards around, combined with the walled garden issue raised above, there wasn't much incentive to write applications. Several U.S. payment card issuers have been introducing contactless cards to their cardholders and trying to persuade merchants to install the necessary terminals. It is too early to tell, but the growth of contactless may seed the market with enough chip-based cards to increase the growth of applications.

Multisided platforms were behind most of the institutions that have formed to facilitate transactions. Consider three examples.

- The first recorded auction, at least according to Herodotus, involved the marriage market. Women were auctioned for the highest price, which could be positive or negative. Men paid for some women, while other women had to offer a dowry to attract a mate. What this lacked in romance it made up for in efficiency. It provided good information and transparent pricing. Other exchange platforms followed relying on many variants of the original highest-bid auction.
- The first money had to get buyers and sellers on board in the same way that American Express needs to get merchants and cardholders on board today. Traders could still have bartered their oxen and kettles or continued to use the irregular metal slugs that were then used as stores of value. The Lydian coins became popular because, like modern-day payment cards, buyers and sellers agreed on this means of exchange and store of value.
- The early advertising-supported media had to get advertisers and eyeballs on the same platform. Magazines were, for many years, resistant to advertisers and, in an effort to protect readers, only permitted them on the back. But publishers quickly learned that advertising boosted sales and covered the high first-copy costs. By the end of the nineteenth century, many magazines were in the business of delivering readers to advertisers.[19]

19. James Twitchell, "Media and the Message," *Advertising Age*, March 29, 1999.

The prevalence of multisided platforms in facilitating transactions historically is not surprising. By definition, these platforms solve transaction problems, broadly construed, between different groups of customers that would like to interact with one another. A multisided platform is often the solution to market frictions.

The Internet provides a technology that can help these multisided platforms operate more efficiently. But that flat description, though true, is an immense understatement. The development of Web-enabled software platforms is leading to the creation of new institutions that can dramatically reduce transactions costs and drastically expand the scope of trade among buyers and sellers around the world. It is this revolution to which we now turn.

eBay

For the fun of it, Pierre Omidyar decided to write the code for an auction program over Labor Day weekend in 1995.[20] He started Friday afternoon. By Monday, September 4, he had a program that would allow users connected through the Internet to list, view, and bid on items. He posted AuctionWeb, as he called it, as one of several home pages on his URL—ebay.com—and announced it on several Internet newsgroups to help attract interest. Slowly, it did.

Omidyar had created several categories of things to list. They included computers, antiques, comic books, and a few others. People started placing items. In the first few weeks these included an autographed poster of Michael Jackson, a Toyota Tercel, and a Mattel Nintendo Powerglove. The buyers and sellers were on their own. The successful bidder paid the seller directly and the seller made arrangements to deliver the merchandise to the successful bidder.

Omidyar didn't charge users for AuctionWeb and hadn't originally intended to turn it into a business. However, as the traffic increased, his Internet service provider complained about the amount of capacity ebay.com was taking and decided to increase his monthly fees. To defray these costs, Omidyar decided to charge sellers 5 percent of the sales price

20. Adam Cohen, *The Perfect Store: Inside eBay* (Boston: Little, Brown, 2002).

for items that sold for less than $25 and 2.5 percent for items that sold for more. Buyers didn't pay anything to look, bid, or buy, and sellers only paid when they made a sale.

The growing community of buyers and sellers was self-policing at first. Omidyar encouraged them to behave ethically and to trust one another. Most did. But, as Webmaster, Omidyar was the natural person to appeal to when they didn't. To help govern the community, Omidyar developed the Feedback Forum. He noted,

> Most people are honest. Some people are dishonest. Or deceptive. This is true here, in the newsgroups, in the classifieds, and right next door. It's a fact of life. But here, those people can't hide. We'll drive them away.[21]

He encouraged people to rate those with whom they transacted on a scale of −1, 0, and +1 and to provide any comments they wished. He barred people who had accumulated several negative ratings from the site.

eBay evolved from these beginnings.[22] The story is told well elsewhere, so we will fast forward to 2005. The code for eBay has grown from the few lines Omidyar wrote over a long weekend to about one million lines of C++ in 1999, and now to more than 6 million lines, mainly written in Java EE. It has gone from running along with many other things on Omidyar's home computer to running on an array of over 9,000 servers in multiple locations. As of the third quarter of 2005, it had 168 million registered users around the world exchanging goods in about 50,000 categories. In 2005 it earned $4.5 billion of revenue on sales of more than $25 billion. Its market capitalization was $62 billion as of January 4, 2006.[23]

The invisible engine that powers this transaction platform has grown enormously from the code that Omidyar wrote and patched together

21. The first part of this section is based in large part on Cohen, *The Perfect Store: Inside eBay*, p. 27.

22. The following is based on interviews with Michael Dearing and Chris Donlay of eBay, December 2005.

23. http://www.auctionbytes.com/cab/abn/y04/m06/i26/s01; Jeffrey Schwartz, "Dot Coms Need You," *VARBusiness*, July 22, 2002; "'Bot' Networks on the Rise, According to Symantec Report," *Bangkok Post*, October 19, 2005; www.sec.gov; http://finance.yahoo.com/q?s=ebay.

from whatever freeware he could find to do the job that Labor Day weekend in 1995. As with all the software platforms we have seen, this growth resulted in large part because eBay kept adding features that were valuable to its community. Of course, if computer code was all there was to eBay, it wouldn't be a software platform in the sense that we've used that term in this book.

In fact, eBay decided to make services provided by its code available to others through APIs. These APIs have resulted in eBay creating an ecosystem of developers that create applications for sellers. Moreover, eBay has provided various tools to sellers themselves that better enable them to benefit from the power of its software engine. Before we explore this aspect of the eBay software platform, we first summarize how this huge online marketplace promotes transactions between buyers and sellers.

eBay helps buyers and sellers come together in two main ways. A number of sellers offer merchandise at a fixed price—"Buy It Now." They are like traditional shopping mall retailers. This model accounted for about 30 percent of eBay sales during the third quarter of 2005.[24] Many other sellers use eBay's auction engine. Auctions have been eBay's long-term focus and what we concentrate on here.

Most auctions start with a minimum price and a fixed length of time over which the seller will accept bids. They usually follow a "second-price"—the highest bidder wins but pays the price offered by the second-highest bidder. This type of auction provides a lot of information to buyers and sellers, and it is transparent. In practice, if not in theory, it tends to avoid the winner's curse, which arises when the high bid is based on overestimation of an item's value, since the winner pays only the second-highest bid. Sellers can also opt to have a reserve price at which they can decline to sell the item. This price is higher than the minimum price; the fact that there is a reserve is disclosed to bidders, but the amount isn't.

eBay could make money in various ways. It could charge buyers to get into its marketplace or for bidding in auctions. It could charge sellers for accessing its platform, using the resources available for establishing

24. "EBay Faces Threat from Google on Fixed-Price Business," Dow Jones News Service, October 25, 2005.

stores, for listing items, or for selling things. Although it has progressed beyond the simple commission fees for sellers that Omidyar established when he was pressed for cash, eBay has maintained a relatively simple pricing structure, one designed to encourage certain behavior as well as to raise revenue.

Buyers don't pay anything directly. They can browse, bid, and buy for free. Sellers pay an insertion fee for each item. These fees are similar to the access or fixed fees that we discussed in Chapter 10; they are independent of whether the item sells or how much it sells for. Sellers also pay a commission on items that are sold. This commission is similar to the variable fees we discussed in Chapter 10. The commission is based on a sliding scale: 5.25 percent for the first $25, 2.75 for the next $975, and 1.5 percent for anything at or beyond $1,000. Finally, sellers can pay to have a reserve price, which varies from $1 for less than $50, $2 for $50 to $199.99, and 1 percent for more than $200.

Omidyar's Feedback Forum has evolved into a critical aspect of eBay. Buyers and sellers are encouraged to provide these ratings. The ratings are aggregated and reported for each registered user as a buyer or seller. Our own experience with eBay is that people are fanatical about these ratings. People don't want negative ratings because it affects their ability to do business with a broad community. Sellers in particular value high ratings because it provides buyers assurance for merchandise that they can't see (except in pictures) from a seller who exists mainly as an email address. These buyer and seller ratings are a valuable asset for eBay and for its community.

Like many exchange platforms, eBay has to satisfy both buyers and sellers. It maximizes revenues more by encouraging sales rather than trying to get the highest price for every sale. The second-price bidding scheme encourages buyers who are fearful of the winner's curse—the tendency for winners of high bid auctions to have overpaid. Sellers are encouraged to adopt low minimum bids and reserves. The rating system provides information on the reliability of buyers and sellers.

As a result, eBay—and rivals around that world that have followed similar approaches—has reduced transactions costs and expanded the scope of trade. We're sure economists will examine the social value that eBay and its imitators have created. It is likely to be enormous, for the

reasons we mentioned earlier. eBay has reduced transactions cost for many buyers and sellers. While these savings are likely small relative to transaction values, when accumulated across buyers and sellers and compounded over time they are likely huge. But, more important, some of the transactions that take place on eBay probably wouldn't have taken place at all. The entire value from trade net of transactions costs would have been left unrealized.

If that were all, eBay would be a revolutionary transaction platform but not one that fits into this book. In fact, eBay is fundamentally different from the London Stock Exchange, Sotheby's, and manheimauctions.com because it allows two groups of businesses to use services provided by the software platform: sellers and developers. And, as a result, it lies at the center of an expanding ecosystem of businesses that benefit from eBay and in turn make eBay a more valuable platform for the eBay communities.

eBay provides sellers with a variety of tools that help them run their businesses through eBay's platform. Much of this involves free tips and advice. Sellers can also attend local "eBay universities" and seminars that provide further instruction. eBay has developed a variety of software programs that help these sellers. Turbolister, for example, is a free application that helps sellers list multiple items on eBay, design the listings, and manage the schedules. Other programs are available for small monthly charges after a 30-day trial period. Selling Manager, for example, helps larger sellers manage their entire eBay program, including downloading sales data from eBay.

Moreover, sellers can also obtain software tools from third-party developers that have built programs that rely on the software services eBay makes available through its APIs. Like the other software platforms we have seen, it didn't take long in its evolution for eBay to realize the importance of a vibrant developer community for helping its buyers and sellers. The eBay Developers Program was started in November 2000. It provides developers with

- Software services available through APIs
- Software development kits that facilitate writing applications
- A "developer zone" that provides, for example, access to tools, sample code, and technical support

- A "developer sandbox" that provides a place where developers can test their applications
- Member forums for online discussions with other developers

Web Services comprise an important set of APIs. They enable developers to create Web-based applications that can conduct business on eBay. These applications, which can be written in any programming language that is capable of making Internet data requests, enable users to do all the things they could do from the eBay desktop, including conducting auctions and managing their stores. There were roughly 2.5 billion calls to the Web Services APIs monthly in 2005, and almost half of the traffic on eBay came through these Web-based applications.

As of the end of 2005, about 21,000 developers have registered for the developer program. Thus far they have developed more than 1,600 applications that buyers and sellers can use with eBay. These include tools for managing auctions, productivity tools, and wireless applications.

Terapeak.com, for example, provides users with access to data on the hundreds of millions of eBay listings, along with analytical tools for examining buying habits and trends for specific categories. It is targeted to sellers who want to better understand the marketplace in which they are competing. Vendio's Ticket Manager is an example of software that is designed to help sellers in a particular category. According to its marketing material, it will "increase your listing capacity, manage your live listings to boost sell through, and fulfill orders with powerful post sale management." Auction Wireless Alerts, by Prisma Corporation, is designed for bidders: it will alert you on your mobile phone when an auction is about to end.

As with many other platforms we have seen, eBay has an annual contest for the "best application" for eBay. The grand prize winner of the "eBay Developer Challenge 2006" was UnWired Buyer, which calls a buyer on her mobile and lets her bid by phone. The first place winner was Auction Contact, which helps online publishers place ads for items on eBay. It also holds an annual developer conference.

Initially, eBay charged developers fees for accessing their APIs. These were intended in part to encourage developers to design efficient software that minimized the load on the software platform. In November

2005, eBay decided to eliminate all developer charges. It is providing free use of the APIs (so long as developers use the most recent version of the platform), membership, and certification, as well as live technical support. eBay also provides a place where developers can promote their applications to the eBay community.

The eBay community is transforming the traditional retail industry. It has made the process of buying and selling more efficient and is providing serious competition for everything from shopping malls to used car dealers. Time will tell the extent to which this software platform destroys traditional businesses. But perhaps the most remarkable aspect of eBay is that it has swept into the economy many transactions that wouldn't have occurred without it.

Google

Google is a software platform, with a search engine at its core, that controls a massive array of Web servers. It derives revenue and profit from facilitating transactions between buyers and sellers. It doesn't, as of the end of 2005, facilitate these transactions directly in the way eBay does. Rather, it does so indirectly through advertisements that point users to particular businesses that can meet a need those users seem likely to have. Virtually all of its revenues and profits come from charging businesses for some form of advertising.

It didn't start that way. We briefly summarize Google's evolution from search engine par excellence to advertising-supported search engine–based transaction platform before examining the business model pursued by this software platform.[25]

As the number of sites and amount of content on the Web expanded rapidly in the mid-1990s, it became apparent that people needed tools for finding things. Programmers began writing search engines. These software applications automatically searched (or "crawled") the web to recover its content, indexed this content in some way that facilitated people finding things, and provided a user interface, including a search

25. Much of the following discussion is based on John Battelle, *The Search: How Google and Its Rivals Rewrote the Rules of Business and Transformed Our Culture* (Huntington, N.Y.: Portfolio Press, 2005).

method for recovering information from the index. AltaVista, created by Digital Equipment Corporation in its death throes, was launched in December 1995.[26] One of the most successful of the early search engines, it handled more than 4 billion search queries during its first year. It became a popular portal and made money by selling display advertising. More sophisticated search engines followed AltaVista. They tried to deliver better search results to users by, for example, conducting statistical analyses of word relationships between Web pages.

Google began as an academic research project conducted, famously, by Stanford engineering graduate students Larry Page and Sergey Brin. Page started working on a doctoral dissertation concerning the mathematical characteristics of the Web. The Web can be thought of as a map of interrelated links. Page's idea was to study who links to whom—a simply stated but computationally difficult problem. Brin joined him. They brought to bear a set of tools that were developed initially by Professor Eugene Garfield. Garfield was a pioneer in the field of information science who did groundbreaking work on the analysis of scientific citations.

To see the idea behind this, consider the sorts of ranking that academics obsess over. Who, for example, is the best evolutionary biologist in the world? One can't really answer that question objectively, so let's pose a different one: Whose scientific papers on evolutionary biology are cited most often? One can answer that question objectively by looking at the citations in academic papers to other academic papers. Papers that are cited more often are presumably more influential and therefore more important. One can refine the analysis—and this is where Garfield's breakthroughs came in—by weighting the citations by the importance of the paper making the citation. So citations by papers that hardly anyone cites (and therefore are presumably not very good) count for less than citations by papers that many people cite (and are therefore presumably very good). This analysis can be extended to the ranking of academic departments, journals, and countries by aggregating across the papers relevant for each.

Applying this concept to the Web, one can ask simply how many Web sites are linking to each Web site. That leads to a simple ranking of Web

26. http://www.clubi.ie/webserch/engines/altavist/history.htm.

sites. One can also ask how many "important" Web sites are linking to each Web site. A simple measure of importance, following scientific citation analysis, is how often a Web site that, itself, has many links is linking to a particular Web site. Web sites that are linked often by other Web sites that are linked often are in some sense better, or at least more interesting.

The Stanford team developed a program that crawled the web and documented the links between sites. Using this program, dubbed BackRub, to collect data on the entire Web on an ongoing basis required an immense amount of computer resources. They then developed an algorithm, PageRank, that ranged pages based roughly on the number of other highly cited pages that cited them. These rankings were married to the standard index that emerges from crawling the Web based on words. The index identifies the Web sites that seem to be relevant to a particular word search. The ranking then identifies the importance of these Web sites.

Computer search is about helping people find the best answers to their questions. Page and Brin made a huge leap forward in doing that. They made their search engine available through Stanford. It was a quick success.

They then started their own search—for a financial return on their innovation. For the first 18 months they tried to sell the technology to some other established search-based Internet businesses. The founder of Infoseek, one of the leading portals of its time, says, "I told them to go pound sand." As more or less did everyone else. The problem, according to John Battele's book on Google's early years, *The Search*, was that the Web portals viewed search as a commodity technology and one that only had to be "good enough."

Page and Brin needed money, though. Between the exponential growth of the Web, which had increased the computing demands for constructing their searches, and the exponential growth of people using their search engines, which had increased the computing demands for handling the searches, they needed more hardware and space than Stanford was able to offer for free. They raised start-up funds and incorporated as Google, Inc., on September 7, 1998.

Google initially earned revenue by licensing its search engine to other companies such as Netscape and Yahoo. They eventually turned to

advertising. This was the way that other Web portals with search features supported themselves. Several questions were already apparent. First, should the pricing scheme entail charging for the number of eyeballs that see an ad or the number of people who click on an ad? Second, should the advertising be separate from the search, such as display ads on a portal that happens to have a search engine, or be integrated into the search, so that advertising that is relevant to the search appears? Third, should the search engine alter the search results—in particular, the order in which results are presented—based on payments made by companies?

The answers to these questions may be apparent in hindsight now, but they were not clear around the turn of the twenty-first century to Google's founders or to many others in search of Internet business models. Battelle's book provides an interesting discussion of how the advertising-supported search industry and Google decided to resolve these issues. We fast-forward to Google's answers.

It is useful to begin by recalling what a Google search result page looks like. Consider a search for "BMW series 6." The top of the page (on December 21, 2005) had two sponsored links by automobile Web portals—edmunds.com, which has reviews as well as links to dealers, and southbankleasing.com, which leases cars. Below this, on the left-hand side of the page, are the ranked search results. Not surprisingly, www.bmwusa.com is the first. On the right-hand side of the page are more sponsored links—on the first page of the search results all of these are places where you could find out about leasing or buying a BMW.

Google settled on charging for the number of clicks. The supported links only pay Google if people click on them. While this is a natural approach for Web-based advertising, the fact that it is used instead of the traditional pay-per-eyeball approach distinguishes Google, and similar firms, from the traditional advertising-supported media industry.

Google also decided to have only search-related advertising, and even then only text-based advertising. There's no reason the BMW page couldn't have an ad that isn't directly related to the search—a Pepsi ad, for example—or a display ad for a local BMW car dealer. But it doesn't. (This reflects a typical two-sided trade-off. In this case, Google decided that it was better to give up certain advertising revenue than to degrade

the quality of the product to the users. By making the search more appealing that increased the number of users and therefore the amount of advertising revenue from those users.)

Finally, Google decided not to take payment for altering the search results. There are many ways to climb the Google charts. Some of these are encouraged, such as selecting keywords that help the search engine make the proper linkages. Others are discouraged, and Google, often to the ire of entities that care about where they are displayed, makes changes in algorithms to defeat gaming of the rankings.

Google operates a bidding process for appearing as a sponsored link. Entities bid on a price per click. However, Google cares about the number of times an advertisement is clicked, as well as on the amount it gets per click. As a result, the actual ranking of the sponsored links on the search pages depends on the price-per-click bid as well as the number of times that the link has been clicked on. Thus, if two entities had bid the same per click, the links with the greater number of clicks will be first; if two links have the same number of clicks, the link with the higher bid will be first. As of 2005, 99 percent of Google's $6 billion revenue came from pay-per-click advertisements—its AdWord program for its own site and its AdSense program for external sites.[27]

Google, as just described, is an advertising-supported search engine that competes mainly with other advertising-supported media. Like other advertising-supported media, it is a multisided platform that supports advertisers and eyeballs. Unlike most traditional advertising-supported media (although like the Yellow Pages), it uses valuable search results to attract eyeballs rather than using content such as *Lost* for ABC or Paul Krugman's column for the *New York Times*. If that were all, as we said for eBay, Google would be a fascinating business, but not one for this book.

The invisible engine behind Google is a software platform that provides services through APIs to software developers. As of the end of 2005, Google offered four major sets of APIs. All of these promote the development of applications that either drive traffic to Google, and

27. http://www.sec.gov/Archives/edgar/data/1288776/000119312506056598/d10k.htm.

therefore enable it to obtain advertising revenue, or enable Google to export advertising to other Web sites or devices.

Web APIs give developers access to the Google search engine so that their programs can pull information from the Web. That could include, for example, sending out periodic search requests to update information on a subject. At the end of 2005, this program was still experimental, and developers who wanted to create commercial services needed to get permission.

AdWords APIs permit developers to write programs that interact with Google's AdWords server. They are particularly helpful to advertisers who want to write internal applications for managing their "sponsored links" on Google's search and developers who want to write applications that they can sell to advertisers—both of which will make advertising with Google more attractive.

Desktop APIs concern a search engine program that Google has made available for local use. The Google Desktop applies Google's search technology to the storage contained on computers used by individuals or enterprises. Google makes APIs available that permit developers to write applications that use this search capability on these computers that are under the control of the individual or enterprise (as opposed to computers that are on the Web).

Map APIs allow developers to write programs that use Google's mapping service. Google has constructed a database of maps and satellite images of the world. It made APIs available to developers for creating applications in June 2005.[28] As of the end of 2005, these APIs were available only for applications that were free to the public. Businesses can use this to develop applications to help people find locations so long as they don't charge for it.

Accessing Google's Web and AdWords APIs imposes costs on Google because they result in additional traffic on its servers. That is quite unlike the APIs in software platforms that reside on local computing devices. It doesn't cost Apple anything when a person runs an application on the computer that relies on a software service that Apple has made available through an API in the Mac OS. Not surprisingly, Google limits the use

28. "Google, Yahoo Offer Maps APIs," *CMP TechWeb*, June 29, 2005.

of APIs that result in traffic on its system. In the case of Web APIs, Google limits users to 1,000 queries a day and, since this is still experimental, doesn't provide any method, aside from negotiating with Google, for obtaining more. In the case of AdWords, Google provides advertisers with a quota of "units" with their advertising account. Different operations based on the AdWords APIs consume various numbers of these units. The quota is tied to the amount of advertising spending. In January 2006, Google began providing a mechanism for commercial developers to obtain larger quotas. As of the end of 2005, Google was not charging for accessing its Map APIs, although it was reserving the right to place advertising on Web pages that relied on these APIs.

Seven years after its formation, Google is at an early stage in creating the kind of developer community that has surrounded other computer-centric software platforms, and is also behind eBay in this regard. Nevertheless, the direction is clear. Google Maps, for example, created an enormous flurry of developer activity quickly. Developers have created applications, some of which can be run from mobile phones, for finding cafés or wireless hotspots and showing criminal activity such as burglaries in small neighborhoods. At the end of 2005, there were at least 500 applications written on these APIs.[29] Google's APIs are poised to support a significant ecosystem of application developers. Google stands to benefit from these APIs by charging for access to its search engine and databases, driving traffic to its site from which it derives advertising revenue, and exporting its paid advertising services to linked sites.

Google is not alone in taking these approaches, although it is the largest Internet advertising-based business, with 2005 revenues of $6 billion, and the one with the highest market capitalization, $125 billion as of the end of 2005.[30] Yahoo has taken somewhat different approaches to developing an advertising-supported platform. Microsoft introduced Windows Live in late 2005 as a Web-centric platform that would compete in many dimensions with Google. In all these cases, the platforms are vehicles for lowering search and transactions costs for

29. There were about 500 Google Maps applications listed on the "Google Maps Mania" blog on December 23, 2005.

30. http://finance.yahoo.com/q?s=goog; http://www.sec.gov/Archives/edgar/data/1288776/000119312505065298/d10k.htm.

consumers and businesses. The ultimate driver of revenue and success is facilitating transactions. Not surprisingly, Google and Microsoft are both developing payment systems that, much like PayPal on eBay, can help their users consummate transactions.

These software platforms are sweeping away traditional industries.We noted earlier, to take one example, newspapers' collapsing stock market valuations. There are several Internet-related reasons behind this, but a major one is that Google and similar platforms can eliminate two major inefficiencies in traditional advertising. Think about all of the advertising dollars that beer manufacturers such as Miller in the United States spend. A large proportion of Americans don't drink beer. Miller ads have no chance of resulting in additional sales from them. It is inefficient both for these "eyeballs" and for Miller for them to spend time glancing at or hearing a Miller ad. Moreover, in the end, Miller has a very limited ability to figure out the extent to which its advertising expenditures result in increased sales. There isn't any convenient way to link exposure to an advertisement to a subsequent sale. (This can only be done, imperfectly, through expensive consumer surveys.) Google and its competitors, however, can tailor the insertion of advertisements to signals from users that they are possibly interested in a topic—buying a BMW for example, or being interested in French restaurants in Tokyo, or where to get the cheapest Miller Lite in Boston.

These software platforms also have the promise of altering significantly how people buy things and how stores sell things and how payments are made. First, along with the auction-based platforms, advertising-supported platforms are likely to move more transactions from physical to virtual stores. Second, the search engine platforms are likely to change the way people buy things at physical stores. For example, wireless devices could be used to find merchandise, compare prices, and guide the user to a store. These devices could also quickly capture feedback on the retail experience, which could be fed back to the software platform. That would extend the important online feedback system to off-line. In both these cases, although eBay and Google have made relatively limited advances thus far, we would expect, based on the other software platforms that we have studied, that developers will create a vast number of applications for auction-centric and search-centric

software platforms. It would be hard to overstate the likely importance of these applications for stimulating innovation in the ecosystems supported by software platforms such as eBay and Google.

Back around the turn of the century there was a great deal of talk about the third industrial revolution, the new economy, and how the Internet would transform the economy as we knew it. Then the dot-com bubble was pricked. Trillions of dollars of value disappeared from stock market valuations in a few short months.[31] Those who hyped the new economy were seen as foolish if not in a few cases criminal. With the benefit of hindsight, though, it looks like the bust was at least as much an overreaction as the bubble. We think three things are clear.

We are about 25 years into a third industrial revolution that is built on microprocessors and, what is often overlooked, the software platforms that use the underlying computer technology to provide powerful services to a wide variety of applications. The first industrial revolution lasted from 1760 to 1830. It was based on innovations such as the steam engine and iron production. The second industrial revolution went from 1850 to 1930. Important developments included the invention of the electric generator and the rise of the chemical industry. Although a longer perspective may change our views on this, it appears that the third industrial revolution started around 1980 with the incorporation of the microprocessor into PCs and video games and the subsequent development of software platforms to create vast ecosystems of businesses around these hardware-software platforms. The second leg of this information technology revolution is the invention of the Internet, which had its commercial birth in 1995.

Just as electric generators drove the development of diverse industries in the second industrial revolution, software platforms have been the invisible engines behind the third industrial revolution. That is not to understate the importance of hardware innovations, which have been essential to what this third revolution has accomplished. Nor is it to minimize the importance of the innovations that led to the Internet: these will ultimately go down in history, we suspect, as some of the most

31. http://catablast.blogspot.com/2005/07/kozmocom-relic-of-dotbomb-bust.html.

important organizational innovations economic history has seen. It is to say, though, that software platforms have played a critical role in sustaining businesses based on microprocessor technologies and are likely to do so for those based on the Internet as well. At a purely technological level, these software platforms permit the software industry to obtain vast scale economies by providing application developers, hardware makers, and content providers with services they all need; the software platforms thereby enable the ecosystem to avoid significant duplication of effort. At a business level, these software platforms permit the formation of ecosystems that create value through the symbiotic relationships between diverse communities.

Software platforms naturally lead to multisided businesses. Almost all of them discussed here have done so. By their nature, multisided platforms reduce the cost of doing business for buyers and sellers. From Apple's OS at the early end of the historical spectrum we have considered to eBay at the late end, these businesses have followed similar multisided strategies to get multiple distinct groups on board and generate value for separate communities of users. Their success, generally, has resulted not from following the dot-com hype of building share quickly and at whatever cost, but from nurturing mutually interdependent communities. In the main, that has meant providing software services for free or at subsidized prices to numerous third parties. That results in the software platform being enveloped in a rich ecosystem of complementors who together provide great value for themselves and for consumers.

History teaches us that it takes decades for technological changes to work their way through the economy, destroying, creating, and transforming industries. The third industrial revolution got off to a quick start. We suspect that it will continue through at least the first few decades of the twenty-first century and that our invisible engines will ultimately touch most aspects of our business and personal lives.

INSIGHTS

- Software platforms have powered new industries such as personal computers and mobile phones, destroyed traditional industries such as typewriters, and disrupted industries from music to payment cards.

- Software platforms are powerful engines of change because of the malleability of code, which makes it easy for them to march across industry boundaries, and because their multisided nature enables them to create vigorous ecosystems of complementors.
- Web-centric platforms that facilitate transactions and lower transactions costs are poised to disrupt the retail sector and advertising-supported media. The 24 percent drop in the market capitalization of the major newspaper publishers between 2004 and 2005 is just one signal of the upcoming transformation.
- The leading Web-centric platforms based on auctions (eBay) and search (Google) have developed multisided strategies based on providing services through APIs to developers and other third parties and encouraging the creation of vibrant ecosystems around their platforms.
- Software platforms are critical players in the third industrial revolution that started around 1980. The first leg of this revolution focused on software platforms that run on dedicated computing devices. The second leg, which began around 2000, is focused on software platforms that run on Web servers and that help businesses and consumers buy goods and services.