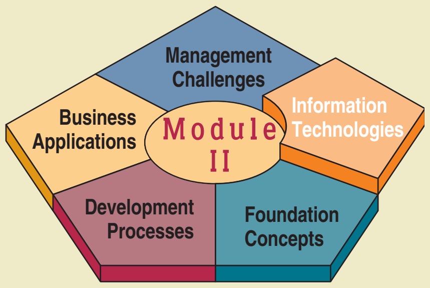
CHAPTER 4



COMPUTER SOFTWARE

Chapter Highlights

Learning Objectives

After reading and studying this chapter, you should be able to:

1. Describe several important trends occurring in computer software.
2. Give examples of several major types of application and system software.
3. Explain the purpose of several popular software packages for end user productivity and collaborative computing.
4. Define and describe the functions of an operating system.
5. Describe the main uses of computer programming software, tools, and languages.

SECTION I Application Software: End User Applications

Introduction to Software

This chapter presents an overview of the major types of software you depend on as you work with computers and access computer networks. It discusses their characteristics and purposes and gives examples of their uses. Before we begin, let's look at an example of the changing world of software in business.

Read the Real World Case on small business software. We can learn a lot about the challenges and opportunities in the small business software market from this example. See Figure 4.1.

What Is Software?

To fully appreciate the need for and the value of the wide variety of software available, we should be sure we understand what software is. **Software** is the general term for various kinds of programs used to operate and manipulate computers and their peripheral devices. One common way of describing hardware and software is to say that software can be thought of as the variable part of a computer and hardware the invariable part. There are many types and categories of software. We will focus our attention on the different types of software and its uses in this chapter.

Types of Software

Let's begin our analysis of software by looking at an overview of the major types and functions of **application software** and **system software** available to computer users, shown in Figure 4.2. This figure summarizes the major categories of system and application software we will discuss in this chapter. Of course, this is a conceptual illustration. The types of software you will encounter depend primarily on the types of computers and networks you use and on what specific tasks you want to accomplish. We will discuss application software in this section and the major types of system software in Section II.

Application Software for End Users

Figure 4.2 shows that **application software** includes a variety of programs that can be subdivided into general-purpose and function-specific application categories. **General-purpose application programs** are programs that perform common information processing jobs for end users. For example, word processing, spreadsheet, database management, and graphics programs are popular with microcomputer users for home, education, business, scientific, and many other purposes. Because they significantly increase the productivity of end users, they are sometimes known as *productivity packages*. Other examples include Web browsers, electronic mail, and groupware, which help support communication and collaboration among workgroups and teams.

One additional common way of classifying software is based on how the software was developed. **Custom software** is the term used to identify software applications that are developed within an organization for use by that organization. In other words, the organization that writes the program code is also the organization that uses the final software application. In contrast, **COTS software** (an acronym which stands for *commercial off-the-shelf*) is software that is developed with the intention of selling the software in multiple copies (and usually for a profit). In this case, the organization that writes the software is not the intended target audience for its use.

Several characteristics are important in describing COTS software. First, as stated in our definition, COTS software products are sold in many copies with minimal changes beyond scheduled upgrade releases. Purchasers of COTS software generally have no control over the specification, schedule, evolution, or access to either the source code or the internal documentation. A COTS product is sold, leased, or licensed to the general public but, in virtually all cases, the vendor of the product retains the intellectual property rights of the software. Custom software, on the other hand, is generally owned by the organization that developed it (or paid to have it developed),

REAL WORLD CASE 1 Microsoft and Others: Developing Software Products for How Companies Do Business

Microsoft, whose fortune has been built around the Windows operating system, is gaining influence on how things get done in an operating room. For the past few years, the software company has been hiring doctors, nurses, and other health-care professionals in an effort to establish internal expertise about the medical industry's IT needs. The strategy is paying off in new accounts and an expanding footprint within the sector.

Already, Matt Maynard, CIO of Pathology Associates Medical Laboratories, credits Microsoft with understanding his business "better than lots of health-care vendors." Yet others see risk in Microsoft stretching too far. "The less Microsoft knows about health care, the better it is for all of us," says Craig Feied, director of the Institute for Medical Informatics at MedStar Health. "The last thing we want is an over-engineered set of solutions built around yesterday's or today's problems."

What Microsoft is doing in health care is a sign of a major strategic shift, one that raises questions in other industries as well. From the time it was founded 28 years ago, Microsoft's focus has been on the software that goes inside computers. Increasingly, however, the company is assessing the business processes of specific industries—and writing software products to support them.

Now Microsoft is expanding the number of industries it targets, injecting industry-specific code directly into its core software platforms and hiring business-technology professionals steeped in the sectors at which it's aiming. Earlier this month, it hired Stuart McKee, the CIO of Washington State, to be U.S. national technology officer of its public-sector and education practice, joining a two-star general and former Coast Guard and Department of Homeland Security officials on that team.

FIGURE 4.1

Small businesses (companies with 1 to 250 employees) are finding mission-critical support for their day-to-day activities through software designed specifically for their size and needs.

Source: Chuck Savage/Corbis.

Microsoft CEO Steve Ballmer describes a two-pronged strategy of selling customizable applications directly to small and medium companies via Microsoft's Business Solutions division, while serving larger companies through partnerships with other technology companies. In both cases, Microsoft engages its wide network of independent software vendors to build apps (applications) that run on top of its own software. What they haven't done in the past is provide business solutions to specific industry (often referred to as *vertical*) segments.

But that's changing. Microsoft engineers are creating software add-ons, called *accelerators*, aimed at business processes common to companies in a given industry. For financial-services companies, Microsoft has an accelerator to help with the trend toward straight-through processing, an automated means of moving a transaction through multiple stages. For health-care companies, they have an accelerator to facilitate information sharing.

And Microsoft Business Solutions has begun inserting what it calls "industry-enabling layers"—software that serves the needs of a broad base of companies in a particular sector—into its enterprise applications. Its latest addition, acquired in April from Encore Business Solutions Inc., is bookkeeping software to deal with the idiosyncrasies of not-for-profits, schools, and other public-sector organizations. The new functionality will be added to the next release of Microsoft's Great Plains applications suite and is the first time Microsoft will integrate technology for the public sector into its software. It has created similar software layers for manufacturing, wholesale distribution, retail, and professional services.

The Microsoft unit that works with independent software vendors has reoriented around vertical industries, too. Until 18 months ago, Microsoft determined its relationships with those vendors by the type of horizontal applications they developed—say, business intelligence or enterprise resource planning. Now Microsoft identifies needs in a particular sector, recruits partners that can fill the need, and jointly creates so-called solution maps of software and services. "We've got a huge pipeline," says Mark Young, Microsoft's general manager of ISVs.

Those partners that aren't threatened by Microsoft's ISV strategy sense an opportunity. Microsoft has so far dealt with manufacturing as one sector, with the exception of specialized service for automotive companies. It's in the early stages of slicing the sector into smaller pieces, planning internal teams for sales, marketing, technology, and support to serve the aerospace, chemical, consumer packaged-goods, high-tech manufacturing, and oil and gas industries. "I'm thrilled," says Ira Dauberman, a VP at UGS PLM Solutions, of Microsoft's plans to target aerospace companies. Boeing is among UGS PLM Solutions's top accounts. "Putting a focus on aerospace is long overdue."

Microsoft is far from the only software company with a vertical strategy. IBM realigned its software division around vertical industries in December. Since then it has delivered preconfigured software bundles for 12 industries. SAP and Oracle go further, developing entire suites of vertical applications. Oracle, for instance, sells applications for real-estate companies and airports. Oracle President Charles Phillips says Microsoft "talks vertical with a horizontal product and tries to package it, twist it, and tweak it." SAP CEO Henning Kagermann contends that Microsoft's heavy reliance on independent software vendors increases complexity. "If you have too many partners in vertical applications, it's always a risk," he says.

Yet Microsoft has a major edge in extending its strategy: Call it Microsoft's Foot In The Door advantage.

When Cooper Tire and Rubber Co., a 90-year-old maker of after-market tires, set out 18 months ago to create a product-life-cycle-management system for designing and building new products, it assessed software from PLM specialists, custom software, and Microsoft. Cooper Tire chose a Microsoft approach, using the company's SharePoint portal software, Project project-management application, and Visio diagramming program.

It was a pragmatic decision: Cooper Tire's license agreement with Microsoft already covered the products needed, so the tire company faced development costs but no added application expense. The other approaches would have cost at least $1.5 million, and Cooper did it for less than half that. But what does Microsoft know about tire manufacturing? "That's what we were wondering, too," says Todd Wilson, project manager of technical systems in Cooper's tire division.

Microsoft brought in a systems integrator—Avanade, a joint venture between Microsoft and Accenture—and bore some of the cost. "The people they've brought in have been experienced manufacturing people. We haven't had to teach them," Wilson says. Microsoft and Avanade spent three months developing a prototype to prove its tools could meet Cooper's needs.

The resulting system helps the company get new tire designs to market in about nine months, half of what it used to take. That scored points with management because speed to market is key to Cooper's strategy of developing high-performance and racing tires to compete with Chinese tire companies. "We were a fast follower. We want to be more of a leader," Wilson says.

Another industry in which Microsoft has well-established customers is retail. It estimates 70 percent of the computing infrastructure in stores runs on Microsoft software. Yet the company is depending on creative thinking to convince retailers to use its software in more—and more strategic—ways.

An initiative called Smarter Retailing, launched in January with 17 partners, sets the lofty goal of revolutionizing the shopping experience through emerging technologies such as using a fingerprint reader in lieu of a credit-card swipe, or a smart phone for one-to-one marketing in a store. Retailers have earned a reputation for treating their best customers the worst, says Janet Kennedy, managing director of Microsoft's retail and hospitality sector, which employs 170 sales, consulting, technical, and service staff to serve 105 top accounts. "The bread, milk, beer customer gets the fast lane," Kennedy says, "and the mother with three kids and $300 worth of groceries gets the slow lane."

Early participants include the A&P supermarket chain and Smart & Final, which operates a chain of warehouse stores. Smarter Retailing could serve as a model for similar undertakings in other areas. "You may see a Smarter Financial Services initiative or Smarter Manufacturing," says Gerri Elliott, VP of worldwide industry solutions.

Ballmer sums up his company's overall vertical push this way: "We've made great progress. We have great capability today that we didn't have a couple of years ago. Yet I believe there's a whole lot more that we can and need to be doing."

Microsoft has already made its mark on the technology industry. The real question is: What industry is next?

Source: Adapted from John Foley, "Strategy Shift: Microsoft Is Looking at How Companies Do Business—and Writing Software Products to Support Those Processes," *InformationWeek*, May 31, 2004. Copyright © 2004 CMP Media LLC.

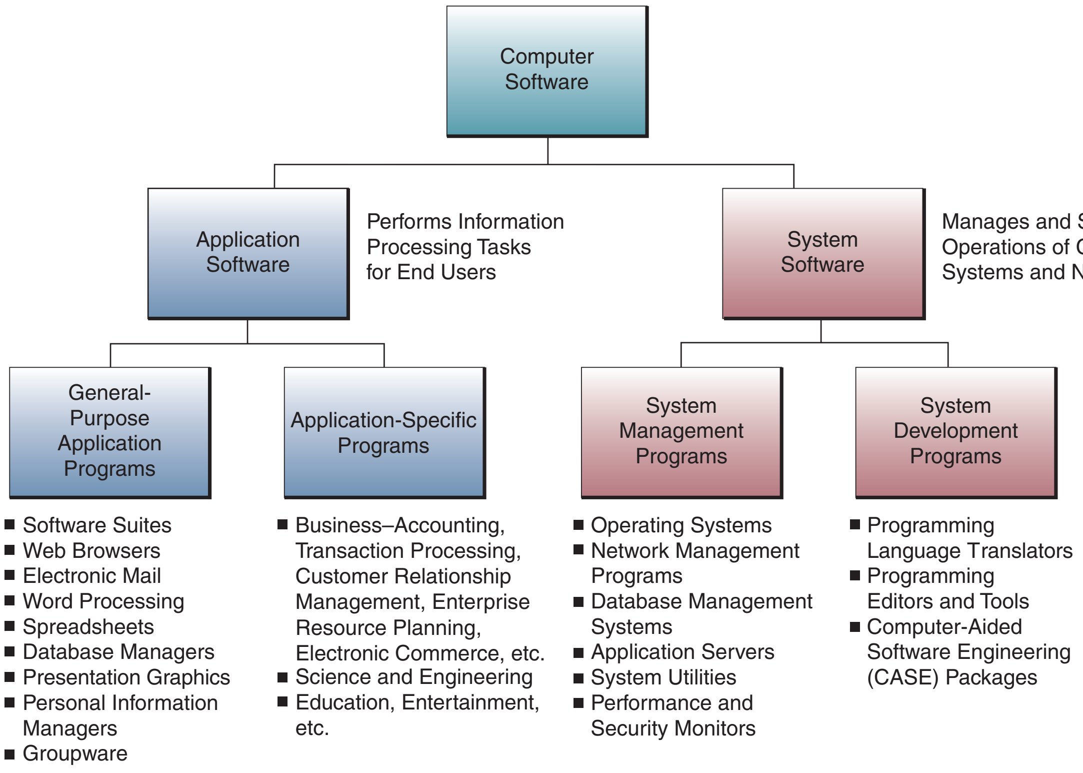
CASE STUDY QUESTIONS

1. A common phrase among IT professionals is "The world views its data through Windows." Why does Microsoft dominate the desktop and networked software market? Visit its website at www.microsoft.com, and review its broad range of software products and services to help with your answer.
2. How successful will Microsoft be in competing with software vendors who specialize in specific market applications like health care, retail, and other specialty services? Why?
3. Do you agree with Microsoft's strategy to develop industry-specific partners to capitalize on opportunities in both large and small business sectors? Is there an advantage or a disadvantage to being one of Microsoft's partners in this type of relationship? Explain.

REAL WORLD ACTIVITIES

1. Industry-specific software applications are everywhere. Despite this, many industries still do not have a wide variety of software applications to support their needs. Using the Internet, see if you can find one example of an industry that has a wide variety of vertical applications and one industry that does not have a variety of software solutions to choose from.
2. Using the industries who do not have a wide variety of support applications you found from the first activity, break into small groups with your classmates, and discuss what types of applications would be valuable to your industries. Why do you think the applications you come up with have not been developed?

FIGURE 4.2 An overview of computer software. Note the major types and examples of application and system software.


Computer Software
Application Software
Performs Information Processing Tasks for End Users
System Software
Manages and Supports Operations of Computer Systems and Networks
General-Purpose Application Programs
Application-Specific Programs
System Management Programs
System Development Programs
■ Software Suites
■ Web Browsers
■ Electronic Mail
■ Word Processing
■ Spreadsheets
■ Database Managers
■ Presentation Graphics
■ Personal Information Managers
■ Groupware
■ Business–Accounting, Transaction Processing, Customer Relationship Management, Enterprise Resource Planning, Electronic Commerce, etc.
■ Science and Engineering
■ Education, Entertainment, etc.
■ Operating Systems
■ Network Management Programs
■ Database Management Systems
■ Application Servers
■ System Utilities
■ Performance and Security Monitors
■ Programming Language Translators
■ Programming Editors and Tools
■ Computer-Aided Software Engineering (CASE) Packages

and the specifications, functionality, and ownership of the final product are controlled or retained by the developing organization.

The newest innovation in software development is called **open-source software.** In this approach, developers collaborate on the development of an application using programming standards which allow for anyone to contribute to the software. Further, as each developer completes his or her project, the code for the application becomes available and free to anyone else who wishes to use it. We will discuss this new approach to software development in greater detail in Section II of this chapter.

Business Application Software

Thousands of **function-specific application software** packages are available to support specific applications of end users in business and other fields. For example, business application software supports the reengineering and automation of business processes with strategic e-business applications like customer relationship management, enterprise resource planning, and supply chain management. Other examples are software packages that Web-enable electronic commerce applications, or in the functional areas of business like human resource management and accounting and finance. Still other software empowers managers and business professionals with decision support tools like data mining, enterprise information portals, or knowledge management systems.

We will discuss these applications in upcoming chapters that go into more detail about these business software tools and applications. For example, data warehousing and data mining are discussed in Chapters 5 and 10; accounting, marketing, manufacturing, human resource management, and financial management applications are covered in Chapter 7. Customer relationship management, enterprise resource planning, and supply chain management are covered in Chapter 8. Electronic commerce is the focus of Chapter 9, and decision support and data analysis applications are explored in

FIGURE 4.3 The business applications in Oracle's E-Business Suite software illustrate some of the many types of business application software being used today.

ORACLE E-BUSINESS SUITE

Advanced Planning	Business Intelligence	Contracts
e-Commerce	Enterprise Asset Management	Exchanges
Financials	Human Resources	Interaction Center
Manufacturing	Marketing	Order Fulfillment
Procurement	Product Development	Professional Services Automation
Projects	Sales	Service
Training	Treasury	

Source: Adapted from Oracle Corp., "E-Business Suite: Manage by Fact with Complete Automation and Complete Information," Oracle.com, 2002.

Chapter 10. Figure 4.3 illustrates some of the many types of business application software that are available today. These particular applications are integrated in the Oracle E-Business Suite software product of Oracle Corp.

Visa International: Implementing an e-Business Suite

Visa International is well known and respected all over the world for the innovations it has brought to global commerce with its sophisticated consumer payments processing system. Surprisingly, however, until recently Visa had many outdated systems managing some of its most critical internal business processes. After an analysis by KPMG in 1999, it was determined that many of Visa's internal systems were becoming a risk to the organization.

The KPMG analysis found Visa's internal systems to be unnecessarily complex and using few of the advantages that technology can bring to an enterprise. For example, Visa's financial management infrastructure was fragmented, complex, and costly to maintain. Often, data were not standardized, resulting in many different databases making disparate interpretations of business data. Even more surprisingly, Visa's corporate purchasing, accounts payable, and asset management functions were still being managed manually, resulting in time-consuming delays and discrepancies.

Fragmented internal systems are not unusual in a company that experiences rapid growth like Visa's double-digit growth for 11 consecutive years. After a careful review of available software solutions, Visa chose the Oracle E-Business Suite of business application software to remedy the problems that come with a complex and inefficient back office.

The results of conversion to the new software suite were spectacular. The modern financial applications in the Oracle product turned Visa's cumbersome, outdated desktop procedures into Web-based e-business solutions that met Visa's demands for all roles and processes. For example, Oracle Financials automated Visa's old organization and created a more agile system capable of accounting for the impact of financial activities on a global scale. Accounts payable was transformed from a cumbersome manual process into a streamlined system that automatically checks invoices against outgoing payments and requests review of any discrepancies via e-mail. And Oracle iProcurement helped automate Visa's requisitioning and purchasing system by streamlining the entire purchasing process and implementing a self-service model to increase processing efficiency [3, 9].

Software Suites and Integrated Packages

Let's begin our discussion of popular general-purpose application software by looking at **software suites**. That's because the most widely used productivity packages come bundled together as software suites such as Microsoft Office, Lotus SmartSuite, Corel WordPerfect Office, and Sun's StarOffice. Examining their components gives us an overview of the important software tools that you can use to increase your productivity.

Figure 4.4 compares the basic programs that make up the top four software suites. Notice that each suite integrates software packages for word processing, spreadsheet, presentation graphics, database management, and personal information management. Microsoft, Lotus, Corel, and Sun bundle several other programs in each suite, depending on the version you select. Examples include programs for Internet access, e-mail, Web publishing, desktop publishing, voice recognition, financial management, electronic encyclopedias, and so on.

A software suite costs a lot less than the total cost of buying its individual packages separately. Another advantage is that all programs use a similar *graphical user interface* (GUI) of icons, tool and status bars, menus, and so on, which gives them the same look and feel, and makes them easier to learn and use. Software suites also share common tools such as spell checkers and help wizards to increase their efficiency. Another big advantage of suites is that their programs are designed to work together seamlessly and import each other's files easily, no matter which program you are using at the time. These capabilities make them more efficient and easier to use than using a variety of individual package versions.

Of course, putting so many programs and features together in one super-size package does have some disadvantages. Industry critics argue that many software suite features are never used by most end users. The suites take up a lot of disk space (often upwards of 150 megabytes), depending on which version or functions you install. Because of their size, software suites are sometimes derisively called *bloatware* by their critics. The cost of suites can vary from as low as $100 for a competitive upgrade to over $700 for a full version of some editions of the suites.

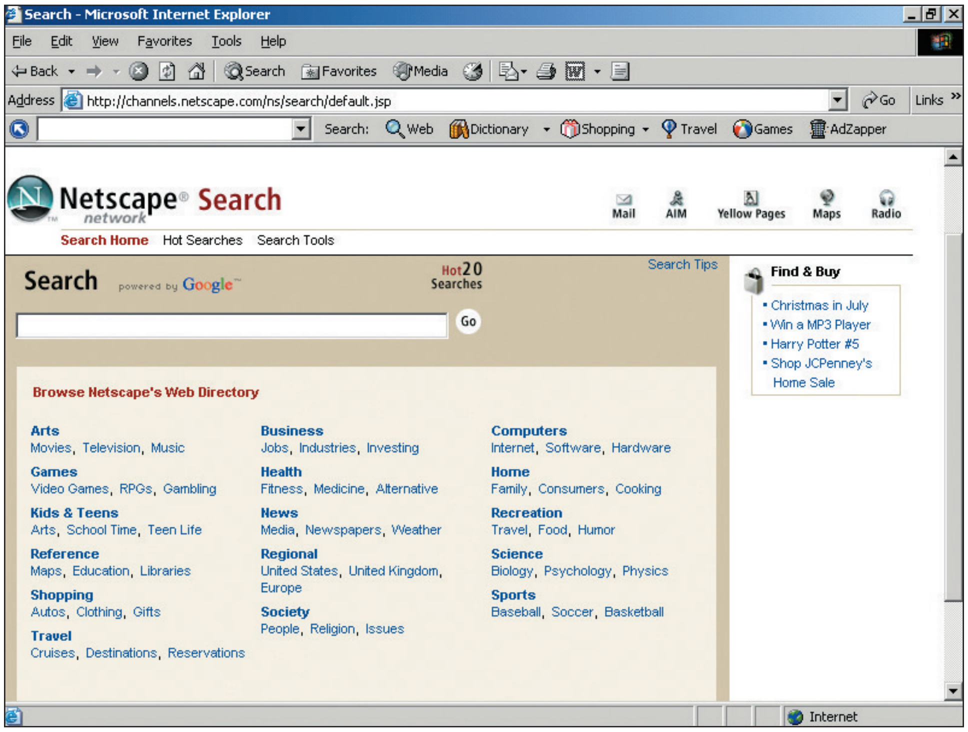
These drawbacks are one reason for the continued use of **integrated packages** like Microsoft Works, Lotus eSuite WorkPlace, AppleWorks, and so on. Integrated packages combine some of the functions of several programs—word processing, spreadsheets, presentation graphics, database management, and so on—into one software package.

Because integrated packages leave out many features and functions that are in individual packages and software suites, they are considered less powerful. Their limited functionality, however, requires a lot less disk space (less than 10 megabytes), costs less than a hundred dollars, and is frequently preinstalled on many low-end microcomputer systems. Integrated packages offer enough functions and features for many computer users while providing some of the advantages of software suites in a smaller package.

FIGURE 4.4
The basic program components of the top four software suites. Other programs may be included, depending on the suite edition selected.

Programs	Microsoft Office	Lotus SmartSuite	Corel WordPerfect Office	Sun StarOffice
Word Processor	Word	WordPro	WordPerfect	Writer
Spreadsheet	Excel	1–2–3	Quattro Pro	Calc
Presentation Graphics	PowerPoint	Freelance	Presentations	Impress
Database Manager	Access	Approach	Paradox	Base
Personal Information Manager	Outlook	Organizer	Corel Central	Schedule

FIGURE 4.5
Using the Microsoft Internet Explorer browser to access Google and other search engines on the Netscape.com website.


Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://channels.netscape.com/ns/search/default.jsp
Go Links
Search: Web Dictionary Shopping Travel Games AdZapper
Netscape Search
network
Mail AIM Yellow Pages Maps Radio
Search Home Hot Searches Search Tools
Search powered by Google
Hot 20 Searches
Search Tips
Go
Internet

Find & Buy
- Christmas in July
- Win a MP3 Player
- Harry Potter #5
- Shop JCPenney's Home Sale

Browse Netscape's Web Directory

Arts
Movies, Television, Music

Games
Video Games, RPGs, Gambling

Kids & Teens
Arts, School Time, Teen Life

Reference
Maps, Education, Libraries

Shopping
Autos, Clothing, Gifts

Travel
Cruises, Destinations, Reservations

Business
Jobs, Industries, Investing

Health
Fitness, Medicine, Alternative

News
Media, Newspapers, Weather

Regional
United States, United Kingdom, Europe

Society
People, Religion, Issues

Computers
Internet, Software, Hardware

Home
Family, Consumers, Cooking

Recreation
Travel, Food, Humor

Science
Biology, Psychology, Physics

Sports
Baseball, Soccer, Basketball

Source: Courtesy of Netscape.

Web Browsers and More

The most important software component for many computer users today is the once simple and limited, but now powerful and feature-rich, **Web browser**. Browsers such as Microsoft Explorer, Netscape Navigator, Firefox, Opera, or Mozilla are software applications designed to support navigation through the point-and-click hyperlinked resources of the World Wide Web and the rest of the Internet, as well as corporate intranets and extranets. Once limited to surfing the Web, browsers are becoming the universal software platform from which end users launch information searches, e-mail, multimedia file transfer, discussion groups, and many other Internet-based applications.

Figure 4.5 illustrates using the Microsoft Internet Explorer browser to access the search engines on the Netscape.com website. Netscape uses top-rated Google as its default search engine, but also provides links to other popular search tools including Ask Jeeves, Look Smart, Lycos, and Overture. Using search engines to find information has become an indispensable part of business and personal Internet, intranet, and extranet applications.

Industry experts predict the Web browser will be the model for how most people use networked computers in the future. Even today, whether you want to watch a video, make a phone call, download some software, hold a videoconference, check your e-mail, or work on a spreadsheet of your team's business plan, you can use your browser to launch and host such applications. That's why browsers are sometimes called the *universal client*, that is, the software component installed on all of the networked computing and communications devices of the clients (users) throughout an enterprise.

Electronic Mail, Instant Messaging, and Weblogs

The first thing many people do at work all over the world is check their electronic mail. **E-mail** has changed the way people work and communicate. Millions of end users now depend on e-mail software to communicate with each other by sending and receiving electronic messages and file attachments via the Internet or their organizations' intranets or extranets. E-mail is stored on networked mail servers until you are ready. Whenever you want to, you can read your e-mail by displaying it on your workstation.

So, with only a few minutes of effort (and a few microseconds of transmission time), a message to one or many individuals can be composed, sent, and received.

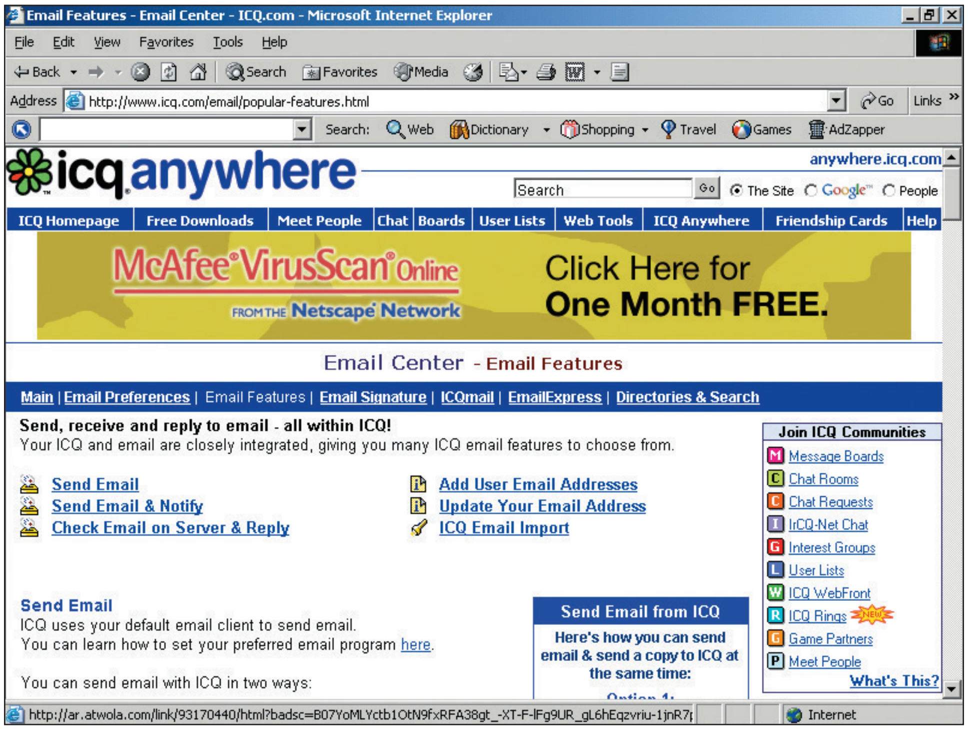
As we mentioned earlier, e-mail software is now a mainstay component of top software suites and Web browsers. Free e-mail packages like Microsoft HotMail and Netscape WebMail are available to Internet users from online services and Internet service providers. Most e-mail software like Microsoft Outlook Express or Netscape Messenger can route messages to multiple end users based on predefined mailing lists and provide password security, automatic message forwarding, and remote user access. They also allow you to store messages in folders and make it easy to add document and Web file attachments to e-mail messages. E-mail packages also enable you to edit and send graphics and multimedia files as well as text, and provide computer conferencing capabilities. Finally, your e-mail software may automatically filter and sort incoming messages (even news items from online services) and route them to appropriate user mailboxes and folders.

Instant messaging (IM) is an e-mail/computer-conferencing hybrid technology that has grown so rapidly that it has become a standard method of electronic messaging for millions of Internet users worldwide. By using instant messaging, groups of business professionals or friends and associates can send and receive electronic messages instantly, and thus communicate and collaborate in real time in a near-conversational mode. Messages pop up instantly in an IM window on the computer screens of everyone in your business workgroup or of your friends on your IM "buddy list" if they are online, no matter what other tasks they are working on at that moment. Instant messaging software can be downloaded and IM services implemented by subscribing to many popular IM systems, including AOL's Instant Messenger and ICQ, MSN Messenger, and Yahoo Messenger. See Figure 4.6.

A **weblog** (sometimes shortened to **blog** or written as "web log" or "Weblog") is a **website** of personal or noncommercial origin that uses a dated log format that is updated daily or very frequently with new information about a particular subject or range of subjects. The information can be written by the site owner, gleaned from other websites or other sources, or contributed by users via e-mail.

FIGURE 4.6

Using the e-mail features of the ICQ instant messaging system.


Email Features - Email Center - ICQ.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.icq.com/email/popular-features.html
Go Links
Search: Web Dictionary Shopping Travel Games AdZapper
icq anywhere
anywhere.icq.com
Search Go The Site Google People
ICQ Homepage | Free Downloads | Meet People | Chat | Boards | User Lists | Web Tools | ICQ Anywhere | Friendship Cards | Help
McAfee VirusScan Online FROM THE Netscape Network
Click Here for One Month FREE.

Email Center - Email Features

Main | Email Preferences | Email Features | Email Signature | ICQmail | EmailExpress | Directories & Search

Send, receive and reply to email - all within ICQ!
Your ICQ and email are closely integrated, giving you many ICQ email features to choose from.

Send Email
Send Email & Notify
Check Email on Server & Reply
Add User Email Addresses
Update Your Email Address
ICQ Email Import

Send Email
ICQ uses your default email client to send email.
You can learn how to set your preferred email program here.

You can send email with ICQ in two ways:

Send Email from ICQ
Here's how you can send email & send a copy to ICQ at the same time:

Join ICQ Communities
Message Boards
Chat Rooms
Chat Requests
IrCQ-Net Chat
Interest Groups
User Lists
ICQ WebFront
ICQ Rings New
Game Partners
Meet People
What's This?

http://ar.atwola.com/link/93170440/html?badsc=B07YoMLYctb1OtN9fxRFA38gt_-XT-F-lFg9UR_gL6hEqzvriu-1jnR7
Internet

Source: Courtesy of ICQ.com.

A weblog often has the quality of being a kind of "log of our times" from a particular point of view. Generally, weblogs are devoted to one or several subjects or themes, usually of topical interest. In general, weblogs can be thought of as developing commentaries, individual or collective, on their particular themes. A weblog may consist of the recorded ideas of an individual (a sort of diary) or be a complex collaboration open to anyone. Most of the latter are *moderated discussions*.

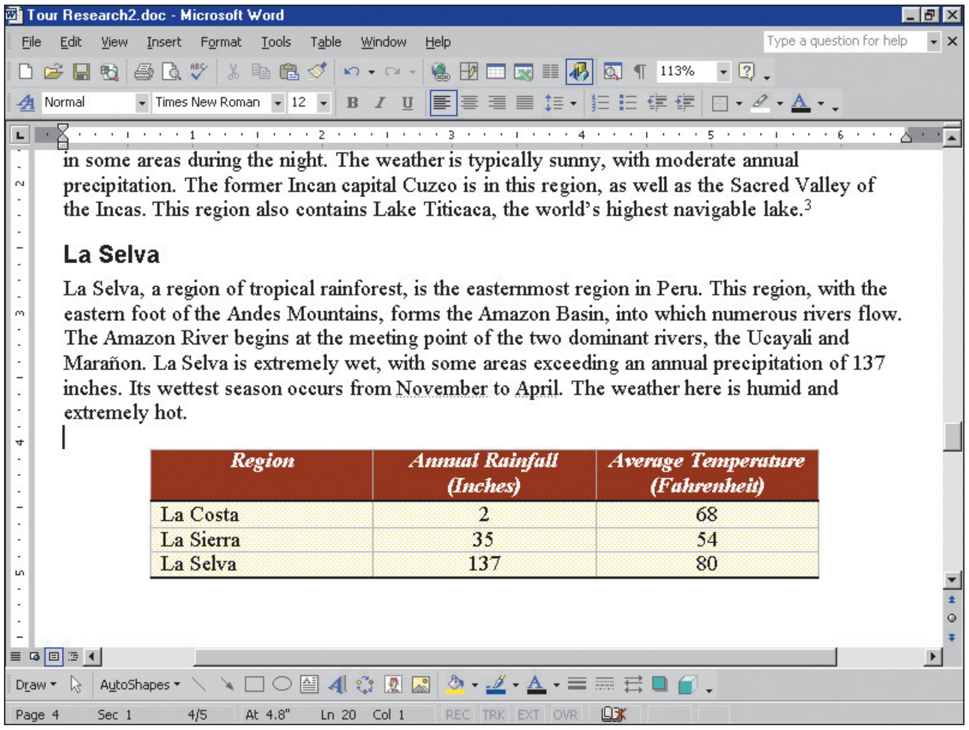
Since there are a number of variations on this idea and new variations can easily be invented, the meaning of this term is apt to gather additional connotations with time. As a format and content approach for a website, the weblog seems popular because the viewer knows that something changes every day, there is a personal (rather than bland commercial) point of view, and, on some sites, there is an opportunity to collaborate with or respond to the website and its participants.

Word Processing and Desktop Publishing

Software for **word processing** has transformed the process of writing just about anything. Word processing packages computerize the creation, editing, revision, and printing of *documents* (such as letters, memos, and reports) by electronically processing *text data* (words, phrases, sentences, and paragraphs). Top word processing packages like Microsoft Word, Lotus WordPro, and Corel WordPerfect can provide a wide variety of attractively printed documents with their desktop publishing capabilities. These packages can also convert documents to HTML format for publication as Web pages on corporate intranets or the World Wide Web.

Word processing packages also provide other helpful features. For example, a *spelling checker* capability can identify and correct spelling errors, and a *thesaurus* feature helps you find a better choice of words to express ideas. You can also identify and correct grammar and punctuation errors, as well as suggest possible improvements in your writing style, with grammar and style checker functions. Besides converting documents to HTML format, you can also use the top packages to design and create Web pages from scratch for an Internet or intranet website. See Figure 4.7.

FIGURE 4.7
Using the Microsoft Word word processing package. Note the insertion of a table in the document.


Tour Research2.doc - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Type a question for help
113%
Normal
Times New Roman
12
Draw
AutoShapes
Page 4 Sec 1 4/5 At 4.8" Ln 20 Col 1 REC TRK EXT OVR

in some areas during the night. The weather is typically sunny, with moderate annual precipitation. The former Incan capital Cuzco is in this region, as well as the Sacred Valley of the Incas. This region also contains Lake Titicaca, the world's highest navigable lake.[3]

La Selva

La Selva, a region of tropical rainforest, is the easternmost region in Peru. This region, with the eastern foot of the Andes Mountains, forms the Amazon Basin, into which numerous rivers flow. The Amazon River begins at the meeting point of the two dominant rivers, the Ucayali and Marañon. La Selva is extremely wet, with some areas exceeding an annual precipitation of 137 inches. Its wettest season occurs from November to April. The weather here is humid and extremely hot.

Region	*Annual Rainfall (Inches)*	*Average Temperature (Fahrenheit)*
La Costa	2	68
La Sierra	35	54
La Selva	137	80

Source: Courtesy of Microsoft Corp.

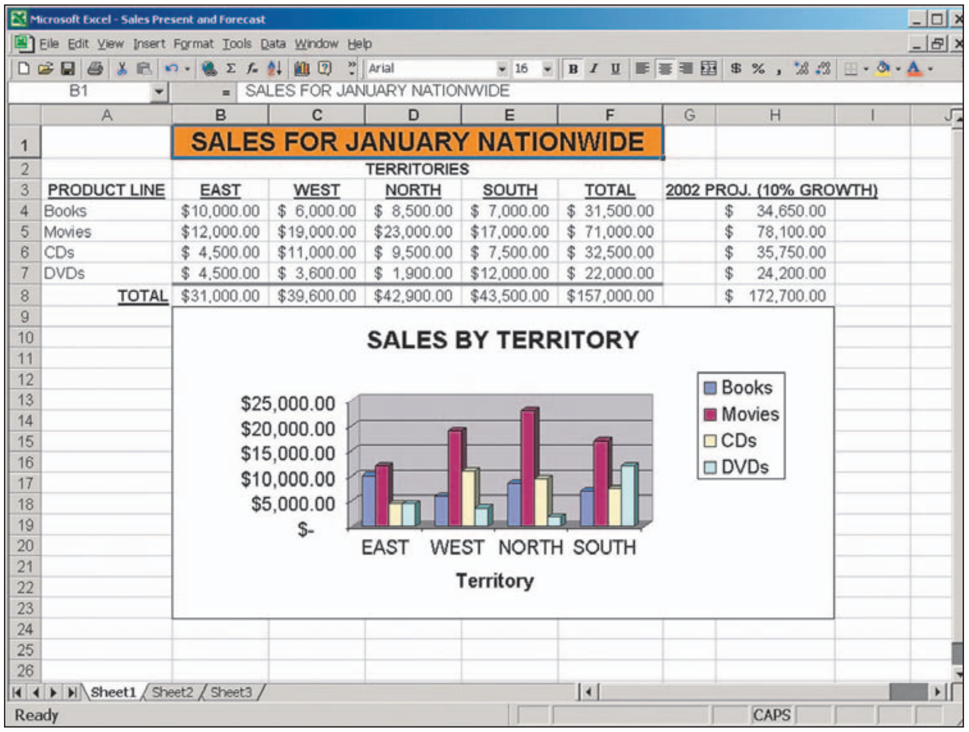
End users and organizations can use **desktop publishing** (DTP) software to produce their own printed materials that look professionally published. That is, they can design and print their own newsletters, brochures, manuals, and books with several type styles, graphics, photos, and colors on each page. Word processing packages and desktop publishing packages like Adobe PageMaker, Microsoft Publisher, and QuarkXPress are used to do desktop publishing. Typically, text material and graphics can be generated by word processing and graphics packages and imported as text and graphics files. Optical scanners may be used to input text and graphics from printed material. You can also use files of *clip art*, which are predrawn graphic illustrations provided by the software package or available from other sources.

Electronic Spreadsheets

Spreadsheet packages like Lotus 1-2-3, Microsoft Excel, and Corel QuattroPro are used by virtually every business for analysis, planning, and modeling. They help you develop an *electronic spreadsheet*, which is a worksheet of rows and columns that can be stored on your PC or on a network server, or converted to HTML format and stored as a Web page or *websheet* on the World Wide Web. Developing a spreadsheet involves designing its format and developing the relationships (formulas) that will be used in the worksheet. In response to your input, the computer performs necessary calculations based on the formulas you defined in the spreadsheet, and displays results immediately, whether at your workstation or website. Most packages also help you develop charts and graphic displays of spreadsheet results. See Figure 4.8.

For example, you could develop a spreadsheet to record and analyze past and present advertising performance for a business. You could also develop hyperlinks to a similar websheet at your marketing team's intranet website. Now you have a decision support tool to help you answer *what-if questions* you may have about advertising. For example, "What would happen to market share if advertising expense were to increase by 10 percent?" To answer this question, you would simply change the advertising expense formula on the advertising performance worksheet you developed. The computer

FIGURE 4.8
Using an electronic spreadsheet package, Microsoft Excel. Note the use of graphics.


Microsoft Excel - Sales Present and Forecast
File Edit View Insert Format Tools Data Window Help
Arial 16
B1 = SALES FOR JANUARY NATIONWIDE
Sheet1 Sheet2 Sheet3
Ready CAPS

SALES FOR JANUARY NATIONWIDE

	TERRITORIES					
PRODUCT LINE	EAST	WEST	NORTH	SOUTH	TOTAL	2002 PROJ. (10% GROWTH)
Books	$10,000.00	$ 6,000.00	$ 8,500.00	$ 7,000.00	$ 31,500.00	$ 34,650.00
Movies	$12,000.00	$19,000.00	$23,000.00	$17,000.00	$ 71,000.00	$ 78,100.00
CDs	$ 4,500.00	$11,000.00	$ 9,500.00	$ 7,500.00	$ 32,500.00	$ 35,750.00
DVDs	$ 4,500.00	$ 3,600.00	$ 1,900.00	$12,000.00	$ 22,000.00	$ 24,200.00
TOTAL	$31,000.00	$39,600.00	$42,900.00	$43,500.00	$157,000.00	$ 172,700.00

Source: Courtesy of Microsoft Corp.

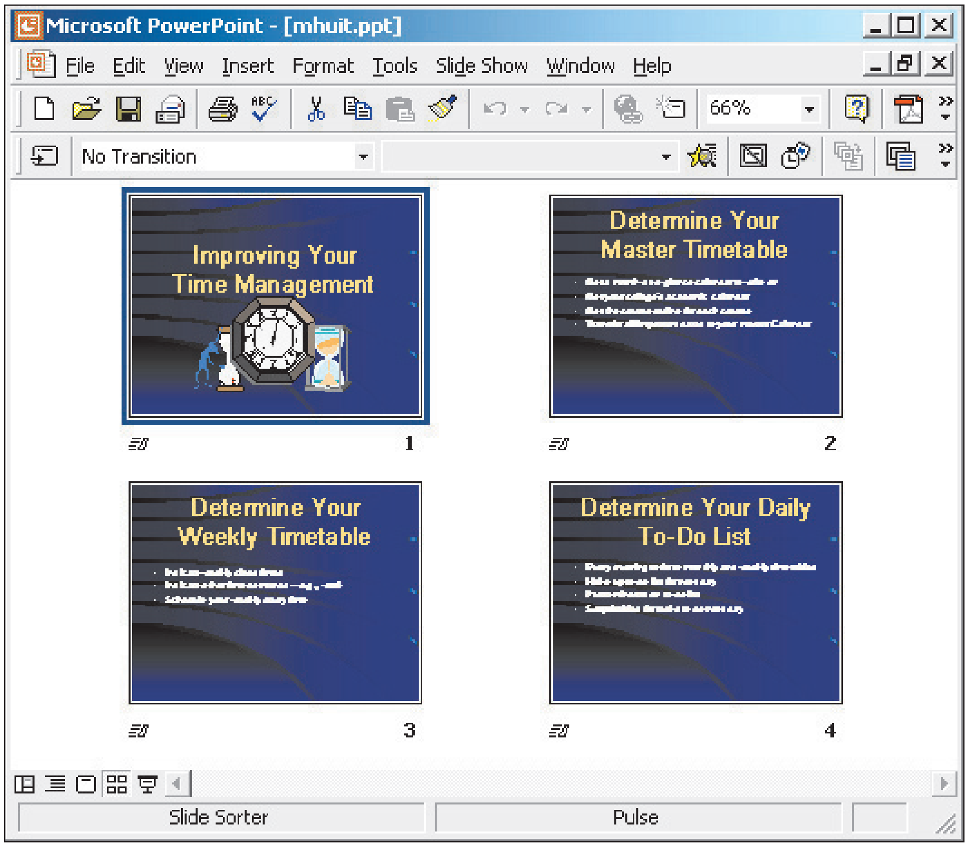
would recalculate the affected figures, producing new market share figures and graphics. You would then have a better insight into the effect of advertising decisions on market share. Then you could share this insight with a note on the websheet at your team's intranet website.

Presentation Graphics

Presentation graphics packages help you convert numeric data into graphics displays such as line charts, bar graphs, pie charts, and many other types of graphics. Most of the top packages also help you prepare multimedia presentations of graphics, photos, animation, and video clips, including publishing to the World Wide Web. Not only are graphics and multimedia displays easier to comprehend and communicate than numeric data, but multiple-color and multiple-media displays also can more easily emphasize key points, strategic differences, and important trends in the data. Presentation graphics have proved to be much more effective than tabular presentations of numeric data for reporting and communicating in advertising media, management reports, or other business presentations. See Figure 4.9.

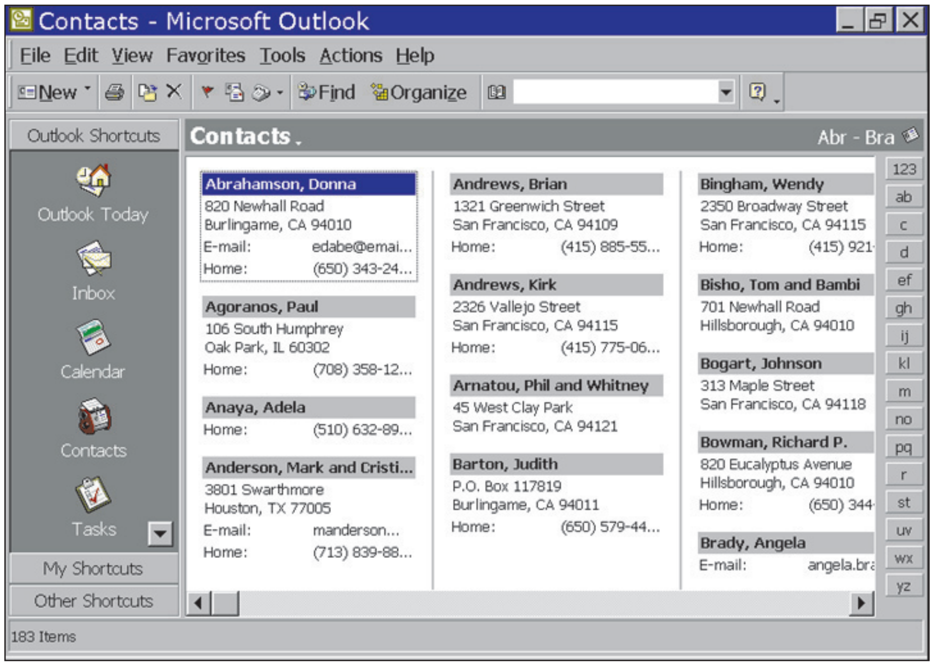
Presentation graphics software packages like Microsoft PowerPoint, Lotus Freelance, or Corel Presentations give you many easy-to-use capabilities that encourage the use of graphics presentations. For example, most packages help you design and manage computer-generated and orchestrated *slide shows* containing many integrated graphics and multimedia displays. Or you can select from a variety of predesigned *templates* of business presentations, prepare and edit the outline and notes for a presentation, and manage the use of multimedia files of graphics, photos, sounds, and video clips. And of course, the top packages help you tailor your graphics and multimedia presentation for transfer in HTML format to websites on corporate intranets or the World Wide Web.

FIGURE 4.9
Using the slide preview feature of a presentation graphics package, Microsoft PowerPoint.


Microsoft PowerPoint - [mhuit.ppt]
File Edit View Insert Format Tools Slide Show Window Help
66%
No Transition
Improving Your Time Management
1
Determine Your Master Timetable
2
Determine Your Weekly Timetable
3
Determine Your Daily To-Do List
4
Slide Sorter
Pulse

Source: Courtesy of Microsoft Corp.

FIGURE 4.10
Using a personal information manager (PIM): Microsoft Outlook.



Abrahamson, Donna
820 Newhall Road
Burlingame, CA 94010
E-mail: edabe@emai...
Home: (650) 343-24...

Agoranos, Paul
106 South Humphrey
Oak Park, IL 60302
Home: (708) 358-12...

Anaya, Adela
Home: (510) 632-89...

Anderson, Mark and Cristi...
3801 Swarthmore
Houston, TX 77005
E-mail: manderson...
Home: (713) 839-88...

Andrews, Brian
1321 Greenwich Street
San Francisco, CA 94109
Home: (415) 885-55...

Andrews, Kirk
2326 Vallejo Street
San Francisco, CA 94115
Home: (415) 775-06...

Arnatou, Phil and Whitney
45 West Clay Park
San Francisco, CA 94121

Barton, Judith
P.O. Box 117819
Burlingame, CA 94011
Home: (650) 579-44...

Bingham, Wendy
2350 Broadway Street
San Francisco, CA 94115
Home: (415) 921-

Bisho, Tom and Bambi
701 Newhall Road
Hillsborough, CA 94010

Bogart, Johnson
313 Maple Street
San Francisco, CA 94118

Bowman, Richard P.
820 Eucalyptus Avenue
Hillsborough, CA 94010
Home: (650) 344-

Brady, Angela
E-mail: angela.br

183 Items

Source: Courtesy of Microsoft Corp.

Personal Information Managers

The **personal information manager** (PIM) is a popular software package for end user productivity and collaboration as well as a popular application for personal digital assistant (PDA) hand-held devices. PIMs such as Lotus Organizer and Microsoft Outlook help end users store, organize, and retrieve information about customers, clients, and prospects, or schedule and manage appointments, meetings, and tasks. A PIM package will organize data you enter and will retrieve information in a variety of forms, depending on the style and structure of the PIM and the information you want. For example, information can be retrieved as an electronic calendar or list of appointments, meetings, or other things to do; as the timetable for a project; or as a display of key facts and financial data about customers, clients, or sales prospects. Most PIMs now include the ability to access the World Wide Web and provide e-mail capability. Also, some PIMs use Internet and e-mail features to support team collaboration by sharing information such as contact lists, task lists, and schedules with other networked PIM users. See Figure 4.10.

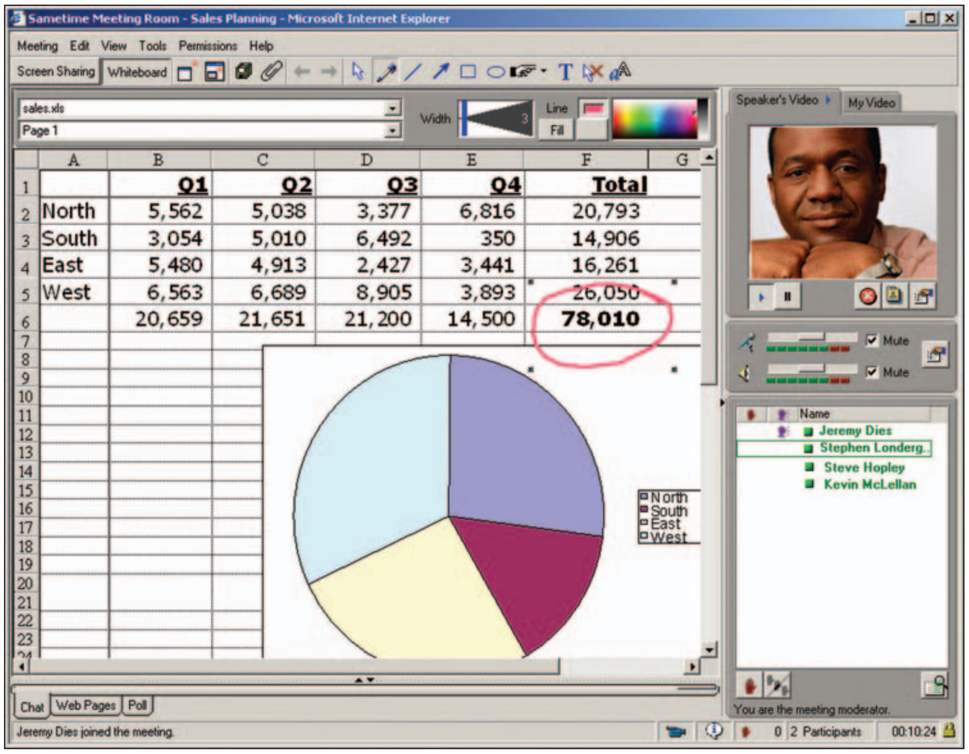
Groupware

Groupware is software that helps workgroups and teams collaborate to accomplish group assignments. Groupware is a category of general-purpose application software that combines a variety of software features and functions to facilitate collaboration. For example, groupware products like Lotus Notes, Novell GroupWise, and Microsoft Exchange support collaboration through electronic mail, discussion groups and databases, scheduling, task management, data, audio and videoconferencing, and so on.

Groupware products rely on the Internet and corporate intranets and extranets to make collaboration possible on a global scale by *virtual teams* located anywhere in the world. For example, team members might use the Internet for global e-mail, project discussion forums, and joint Web page development. Or they might use corporate intranets to publish project news and progress reports, and work jointly on documents stored on Web servers. See Figure 4.11.

Collaborative capabilities are also being added to other software to give it groupware-like features. For example, in the Microsoft Office software suite, Microsoft Word keeps track of who made revisions to each document, Excel tracks all changes

FIGURE 4.11
Lotus Sametime enables workgroups and project teams to share spreadsheets and other work documents in an interactive online collaboration process.



	A	B	C	D	E	F
1		Q1	Q2	Q3	Q4	Total
2	North	5,562	5,038	3,377	6,816	20,793
3	South	3,054	5,010	6,492	350	14,906
4	East	5,480	4,913	2,427	3,441	16,261
5	West	6,563	6,689	8,905	3,893	26,050
6		20,659	21,651	21,200	14,500	78,010

Source: Courtesy of IBM Lotus Software.

made to a spreadsheet, and Outlook lets you keep track of tasks you delegate to other team members. Recently, Microsoft Office suite has included functions that allow multiple people to work on and edit the same document at the same time. Using this feature, any changes made by one team member will become visible to all team members as they are being made.

Two recent additions to the collaborative software marketplace are Microsoft's Windows SharePoint Services and IBM's WebSphere. Both products allow teams to quickly create sophisticated websites for information sharing and document collaboration. Further, businesses can use these products as a platform for application development to facilitate the efficient creation of Web-based business portals and transaction processing applications. Websites built with collaborative development tools can integrate a wide variety of individual applications that can help increase both individual and team productivity.

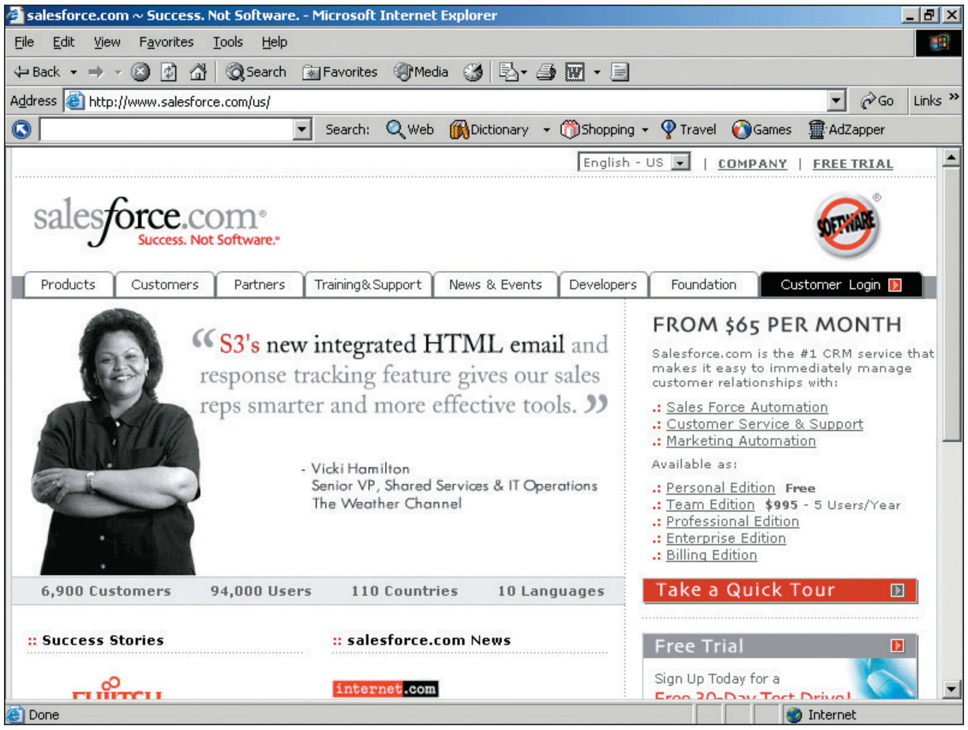
Software Alternatives

Many businesses are finding alternatives to acquiring, installing, and maintaining business application software purchased from software vendors or developing and maintaining their own software in-house with their own software developer employees. For example, as we will discuss further in Chapter 14, many large companies are *outsourcing* the development and maintenance of software they need to *contract programming* firms and other software development companies, including the use of *offshore* software developers in foreign countries, and using the Internet to communicate, collaborate, and manage their software development projects.

Application Service Providers

A large and fast-growing number of companies are turning to **application service providers** (ASPs), instead of developing or purchasing the application software they need to run their businesses. Application service providers are companies that own, operate, and maintain application software and the computer system resources (servers, system software, networks, and IT personnel) required to offer the use of the

FIGURE 4.12
Salesforce.com is a leading application service provider of Web-based sales management and customer relationship management services to both large and small businesses.


salesforce.com ~ Success. Not Software. - Microsoft Internet Explorer

Address http://www.salesforce.com/us/

English - US | COMPANY | FREE TRIAL

salesforce.com Success. Not Software.

Products | Customers | Partners | Training&Support | News & Events | Developers | Foundation | Customer Login

"S3's new integrated HTML email and response tracking feature gives our sales reps smarter and more effective tools."

- Vicki Hamilton
Senior VP, Shared Services & IT Operations
The Weather Channel

6,900 Customers 94,000 Users 110 Countries 10 Languages

FROM $65 PER MONTH

Salesforce.com is the #1 CRM service that makes it easy to immediately manage customer relationships with:

.: Sales Force Automation
.: Customer Service & Support
.: Marketing Automation

Available as:

.: Personal Edition Free
.: Team Edition $995 - 5 Users/Year
.: Professional Edition
.: Enterprise Edition
.: Billing Edition

Take a Quick Tour

:: Success Stories

:: salesforce.com News

Free Trial

Sign Up Today for a

Done Internet

Source: Courtesy of Salesforce.com.

application software for a fee as a service over the Internet. The ASP can bill their customers on a per-use basis, or on a monthly or annual fee basis.

Businesses are using an ASP instead of owning and maintaining their own software for many reasons. One of the biggest advantages is the low cost of initial investment, and in many cases, the short time needed to get the Web-based application set up and running. The ASP's pay-as-you-go fee structure is usually significantly less expensive than the cost of developing or purchasing, as well as running and maintaining, the application software. And using an ASP eliminates or drastically reduces the need for much of the IT infrastructure (servers, system software, and IT personnel) that would be needed to acquire and support the application software, including the continual challenges of distributing and managing companywide software patches and upgrades. Consequently, the use of ASPs by businesses and other organizations is expected to accelerate in the coming years [14]. See Figure 4.12.

Premiere Technologies: Vital Information and Big Savings through an ASP

In the late 1990s through 2002, Atlanta-based Premiere Technologies, a $500 million integrated personal communications services provider, acquired more than 100 small telecommunications companies throughout the world. Its goal was clear: to become a leading provider of value-added communications services, including conference calling, messaging, and Internet-based services.

Central to its aggressive growth strategy, Premiere Technologies implemented a PeopleSoft ERP system—a soup-to-nuts software suite that would provide information vital for running nearly all aspects of Premiere's business. The ERP system would integrate Premiere's business processes and far-flung offices. Employees would then be able to share information companywide, resulting in cost efficiencies, productivity growth, and intelligence gains.

ERP systems are notoriously difficult to implement and run. As a result, Premiere Technologies decided to outsource the day-to-day management of its ERP system to Atlanta-based TransChannel Inc., whose Internet Enabled ERP (iE2) outsourcing solution delivers comprehensive PeopleSoft support, maintenance, and deployment for a fixed monthly fee. Within two months, TransChannel brought the entire PeopleSoft ERP suite online (versus an estimated 18 months for internal deployment). Premiere Technologies also saved millions of dollars through capital cost avoidance by not having to purchase the platform infrastructure and upgrade all the client hardware necessary for rolling out the ERP system internally. The bottom-line result: enormous gains in efficiency, productivity, companywide communication, and worker satisfaction through increased application reach, reliability, and flexibility [4, 13].

Software Licensing

Regardless of whether a software application is purchased COTS or is accessed via an ASP, the software must be licensed for use. Software licensing is a complex topic that involves considerations of the special characteristics of software in the context of the underlying intellectual property rights, including copyright, trademark, and trade secrets, as well as traditional contract law, including the Uniform Commercial Code (UCC).

Contrary to what many may believe, when an individual or a company buys a software application, they have not purchased the rights of ownership. Rather, they have purchased a license to use the software under the terms of the software licensing agreement. Software is generally licensed in order to better protect the vendor's intellectual property rights. The license often prohibits reverse engineering, modifying, disclosing, or transferring the software. In most cases, the license also gives the purchaser permission to sell or dispose of the rights provided by the license but not to duplicate or resell multiple copies of the software.

The requirement for licensing does not disappear when use of the software is obtained through an ASP. In this case, the license to dispense use of the software is granted to the ASP by the various software vendors and, in return, the ASP agrees to pay the software vendor a royalty based on the number of user accounts the ASP resells the rights to.

Software vendors are working hard to provide easy licensing and access to their products while simultaneously preventing software piracy, which serves only to raise the ultimate cost of the product.

In the next section, we will learn about an entirely new approach to software licensing: open-source code.

SECTION II System Software: Computer System Management

System Software Overview

System software consists of programs that manage and support a computer system and its information processing activities. For example, operating systems and network management programs serve as a vital *software interface* between computer networks and hardware and the application programs of end users.

Read the Real World Case on webtop software. We can learn a lot about the business productivity enhancements of new software developments from this example. See Figure 4.13.

Overview

We can group system software into two major categories (see Figure 4.14):

- **System management programs.** Programs that manage the hardware, software, network, and data resources of computer systems during the execution of the various information processing jobs of users. Examples of important system management programs are operating systems, network management programs, database management systems, and system utilities.
- **System development programs.** Programs that help users develop information system programs and procedures and prepare user programs for computer processing. Major software development programs are programming language translators and editors, and a variety of CASE (computer-aided software engineering) and other programming tools. We will take a closer look at CASE tools later in this chapter.

Operating Systems

The most important system software package for any computer is its operating system. An **operating system** is an integrated system of programs that manages the operations of the CPU, controls the input/output and storage resources and activities of the computer system, and provides various support services as the computer executes the application programs of users.

The primary purpose of an operating system is to maximize the productivity of a computer system by operating it in the most efficient manner. An operating system minimizes the amount of human intervention required during processing. It helps your application programs perform common operations such as accessing a network, entering data, saving and retrieving files, and printing or displaying output. If you have any hands-on experience on a computer, you know that the operating system must be loaded and activated before you can accomplish other tasks. This emphasizes the fact that operating systems are the most indispensable components of the software interface between users and the hardware of their computer systems.

Operating System Functions

An operating system performs five basic functions in the operation of a computer system: providing a user interface, resource management, task management, file management, and utilities and support services. See Figure 4.15.

The User Interface. The **user interface** is the part of the operating system that allows you to communicate with it so you can load programs, access files, and accomplish other tasks. Three main types of user interfaces are the *command-driven*, *menu-driven*, and *graphical user interfaces*. The trend in user interfaces for operating systems and other software is moving away from the entry of brief end user commands, or even the selection of choices from menus of options. Instead, most software provides an easy-to-use graphical user interface (GUI) that uses icons, bars, buttons, boxes, and other images. GUIs rely on pointing devices like the electronic mouse or touchpad to make selections that help you get things done. Currently, the most common and widely recognized GUI is the Microsoft Windows desktop.

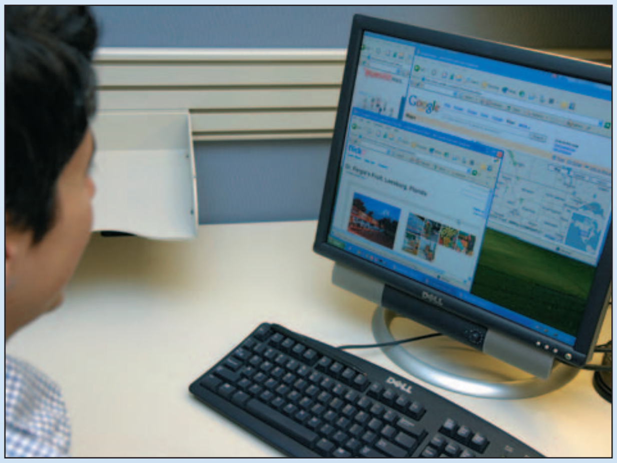
REAL WORLD CASE 2 Webtop Software: Transforming the Desktop with Personalized Web-Based Applications

It's been a long time—all the way back to the dawn of desktop computing in the early 1980s—since software developers have had as much fun as so many are having right now. Oh, building the latest version of yet another online gaming experience is challenging fun, we admit, but in productivity and business software development today, browser-based applications are where the action is. A killer webtop app no longer requires hundreds of drones slaving away on millions of lines of code. Three or four engineers and a steady supply of Red Bull are frequently all it takes to rapidly (maybe a few days or weeks) turn a midnight Web product brainstorm into a website so hot it melts the servers.

What has changed is the way today's Web-based apps can run almost as seamlessly as programs used on the desktop, with embedded audio, video, and drag-and-drop ease of use. Behind this Web-desktop fusion are technologies like Ajax (asynchronous JavaScript and XML), Macromedia's Flash, and Ruby on Rails. We'll spare you the technical details at this point; suffice it to say that these technologies are giving rise to a new webtop that may one day replace your present suite of desktop applications. Yes, that's what we said.

Hmm, we see you need convincing. Why not start with Writely, a free online word processor that anyone who knows how to use Microsoft Word will figure out in a few clicks. Then add Zimbra, which is taking a swipe at Microsoft Outlook with an online e-mail application that has all the latest neat Ajax tricks built in. Like what? You may ask. Okay, glide your mouse over an e-mail message that includes a date, and your calendar for that day pops up. Move it over a website address in the message and an image of the page appears. See what we mean?

FIGURE 4.13



The ease and productivity enhancement of personalized Web-based application software promises to make your webtop a powerful alternative to desktop software suites.

Source: John Flournoy/MHHEDIL.

For an online spreadsheet, try Tracker, the latest release from JotSpot (better known for its group-editing "wiki" software). Tracker becomes an interactive website open to viewing or changing websheets by the people you invite. Users also will soon be able to subscribe to a particular spreadsheet row (say, "Sales in China") via an RSS feed, and have it automatically fed to their webtops . . . or websites . . . or weblogs . . . get the picture?

All of these programs link to myriad open APIs (application programming interfaces) that serve as building blocks for new applications and data on the Web from Amazon, Google, and others. Thus can the information on your desktop be fused with the entire Web through a powerful and increasingly invisible bridge between the two. Now are you impressed?

Well, Google, Microsoft, and Yahoo are energetically trying to crash this party. When the big boys start elbowing themselves to the front of the online webtop software market, you know it's more than another gee-whiz technology passing fancy. They smell new sources of revenue from online software and webtop-presence market share gains, and they're not going to pass that up. Why even big and not too swift lately Microsoft recently launched beta versions of Windows Live, a personal online command center for e-mail, RSS feeds, and other content, and followed up quickly with beta versions of Office Live, a website-hosting and online project management service that taps into the existing Office desktop programs and is aimed at the massive but underserved small business market.

But let's not forget the smaller, cutting-edge software developers and webtop products in this new market, several of whom we have already mentioned. At the top spot is JotSpot of Palo Alto, whose products are wikis and online spreadsheets. JotSpot, a pioneer of Web collaboration apps, a.k.a. wikis, developed the new Tracker webtop application we described earlier, which provides a powerful, highly collaborative online spreadsheet.

We also mentioned the Writely online word processor, a product of Writely of Portola Valley, California. Its software enables online creation of documents, opens them to collaboration by anyone anywhere, and simplifies publishing the end result on a website as a blog entry. The Zimbra online e-mail product we mentioned earlier was developed by Zimbra of San Mateo, California. Besides bringing up your calendar for any date your mouse encounters in an e-mail, it can, among other things, launch Skype for any e-mail phone number, or retrieve a Google map for any e-mail address it bumps into.

An online calendar is the product of 30Boxes of San Francisco. This Web-based software allows families and groups to create private social networks, organize events, track schedules, and share photos; it will soon allow you to save phone numbers as hyperlinks and make calls by simply clicking on a link. Our final webtop software kudo goes to 37Signals of Chicago for its Basecamp online project management product. Its Basecamp app, elegant and inexpensive, enables the creation, sharing, and tracking of to-do lists, files, performance milestones, and other key project metrics. The company's related Backpack webtop application, recently released, is a powerful online organizer for individuals that can be easily added to anybody's webtop capabilities.

The business question for you at this point is: Why invest your time and money in a webtop product by one of these midget developers who may not survive the onslaught of giants like Google, Yahoo, and Microsoft in this market? After all, Monsieur Bill Gates made it quite plain during the November 2005 launches of Windows Live and Office Live that grabbing a dominant share of the webtop is a top strategic priority for Microsoft. Of course, Google, Yahoo, and other big players vow they won't let that happen.

But not to worry. If the history of Silicon Valley is any guide, a few of the midgets will become the next big thing, and the best remaining will be bought out by the giants, their products reworked and renamed, and you'll be two steps ahead of your mate in the next cubicle for having had the innovative spark to be one of thousands of pioneers in enjoying the ease and productivity of such software products on your personalized webtop.

Source: Adapted from Erik Schonfeld, Om Malik, and Michael V. Copeland, "The Next Net 25: The Webtop," *Business 2.0*, March 2006, and Ina Fried, "Windows Live Offers Microsoft a Quick Turnaround," *CNET News.com*, March 14, 2006.

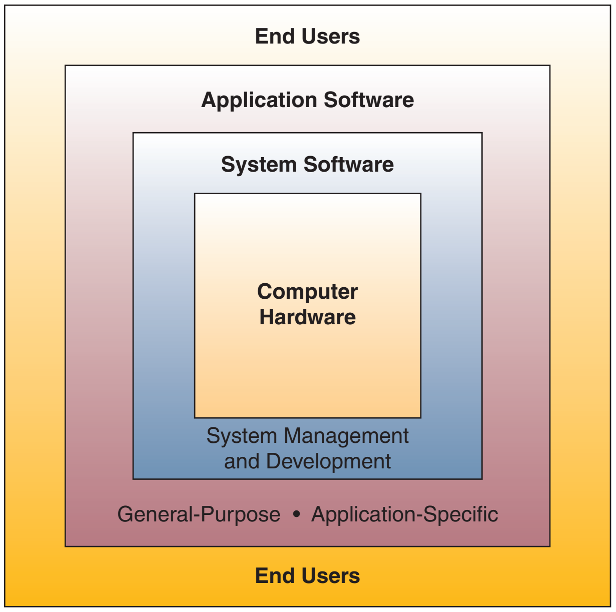
CASE STUDY QUESTIONS

1. Do you agree that webtop software will one day replace suites of desktop applications? Why or why not? Check out the features of a few of the webtop products mentioned in the case on the Internet to support your answer.
2. Will Microsoft succeed in dominating the webtop? Why or why not? Visit the websites of Windows Live and Office Live and review their products and services to support your answer.
3. Should you invest your time and money in acquiring and learning how to use some of the webtop applications mentioned in this case? Defend your answer based on your review of the webtop products from the small developers and Microsoft that you did earlier.

REAL WORLD ACTIVITIES

1. Research the websites of Google and Yahoo to find evidence of their entry into the webtop application software market. Evaluate several products you find, comparing them to the offerings of Microsoft and the independent software developers mentioned in the case.
2. Try out demo versions on the Internet of several of the webtop software applications mentioned in this case, including those that may be available from Google and Yahoo. Break into small groups with your classmates to discuss your reactions to the experience, and the features you would like to see changed or added that might lure you and others into acquiring one or more of these webtop products.

FIGURE 4.14
The system and application software interface between end users and computer hardware.

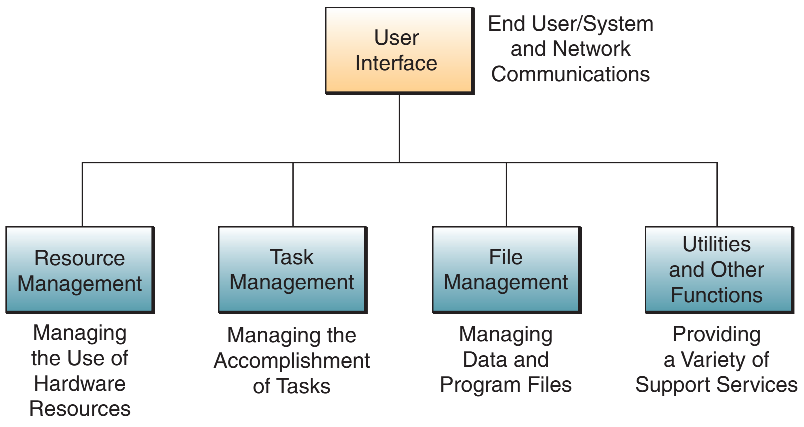

End Users
Application Software
System Software
Computer Hardware
System Management and Development
General-Purpose • Application-Specific
End Users

Resource Management. An operating system uses a variety of **resource management** programs to manage the hardware and networking resources of a computer system, including its CPU, memory, secondary storage devices, telecommunications processors, and input/output peripherals. For example, memory management programs keep track of where data and programs are stored. They may also subdivide memory into a number of sections and swap parts of programs and data between memory and magnetic disks or other secondary storage devices. This can provide a computer system with a **virtual memory** capability that is significantly larger than the real memory capacity of its primary storage circuits. So, a computer with a virtual memory capability can process large programs and greater amounts of data than the capacity of its memory chips would normally allow.

File Management. An operating system contains **file management** programs that control the creation, deletion, and access of files of data and programs. File management also involves keeping track of the physical location of files on magnetic disks and other secondary storage devices. So operating systems maintain directories of information about the location and characteristics of files stored on a computer system's secondary storage devices.

Task Management. The **task management** programs of an operating system manage the accomplishment of the computing tasks of the end users. The programs control

FIGURE 4.15
The basic functions of an operating system include a user interface, resource management, task management, file management, and utilities and other functions.


User Interface
End User/System and Network Communications
Resource Management
Managing the Use of Hardware Resources
Task Management
Managing the Accomplishment of Tasks
File Management
Managing Data and Program Files
Utilities and Other Functions
Providing a Variety of Support Services

which task gets access to the CPU and for how much time. The task management functions can allocate a specific slice of CPU time to a particular task and interrupt the CPU at any time to substitute a higher priority task. Several different approaches to task management may be taken, each with advantages in certain situations.

Multitasking (sometimes referred to as *multiprogramming* or *time-sharing*) is a task management approach that allows for several computing tasks to be performed in a seemingly simultaneous fashion. In reality, multitasking assigns only one task at a time to the CPU, but it switches from one program to another so quickly that it gives the appearance of executing all of the programs at the same time. There are two basic types of multitasking: *preemptive* and *cooperative*. In preemptive multitasking, the task management functions parcel out CPU *time slices* to each program. In contrast, cooperative multitasking allows each program to control the CPU for as long as it needs it. If a program is not using the CPU, however, it can allow another program to use it temporarily. Most Windows- and Unix-based operating systems use the preemptive approach, while most Macintosh-style platforms use cooperative multitasking. Although the terms *multitasking* and *multiprocessing* are often used interchangeably, they are actually different concepts based on the number of CPUs being used. In multiprocessing, more than one CPU is being accessed, but in multitasking, only one CPU is in operation.

Most computers make use of some sort of multitasking. On modern microcomputers, multitasking is made possible by the development of powerful processors and their ability to directly address much larger memory capacities. This allows primary storage to be subdivided into several large partitions, each of which is being used by a different software application.

In effect, a single computer can act as if it were several computers, or *virtual machines*, since each application program is running independently at the same time. The number of programs that can be run concurrently depends on the amount of memory that is available and the amount of processing each job demands. That's because a microprocessor (or CPU) can become overloaded with too many jobs and provide unacceptably slow response times. However, if memory and processing capacities are adequate, multitasking allows end users to easily switch from one application to another, share data files among applications, and process some applications in a *background* mode. Typically, background tasks include large printing jobs, extensive mathematical computation, or unattended telecommunications sessions.

Microsoft Windows

For many years, MS-DOS (Microsoft Disk Operating System) was the most widely used microcomputer operating system. It is a single-user, single-tasking operating system, but was given a graphical user interface and limited multitasking capabilities by combining it with Microsoft **Windows.** Microsoft began replacing its DOS/Windows combination in 1995 with the Windows 95 operating system, featuring a graphical user interface, true multitasking, networking, multimedia, and many other capabilities. Microsoft introduced an enhanced Windows 98 version during 1998, and a Windows Me (Millennium Edition) consumer PC system in 2000.

Microsoft introduced its **Windows NT** (New Technology) operating system in 1995. Windows NT is a powerful, multitasking, multiuser operating system that was installed on many network servers to manage PCs with high-performance computing requirements. New Server and Workstation versions were introduced in 1997. Microsoft substantially enhanced its Windows NT products with the **Windows 2000** operating system during the year 2000.

Late in 2001, Microsoft introduced **Windows XP** Home Edition and Professional versions, and thus formally merged its two Windows operating system lines for consumer and business users, uniting them around the Windows NT and Windows 2000 code base. With Windows XP, consumers and home users finally received an enhanced Windows operating system with the performance and stability features that business users had in Windows 2000 and continue to have in Windows XP Professional. Microsoft also introduced four new **Windows Server 2003** versions in 2003, which are summarized and compared in Figure 4.16 [8].

FIGURE 4.16 Comparing the purposes of the four versions of the Microsoft Windows Server 2003 operating system.

Microsoft Windows Server 2003 Comparisons

- **Windows Server 2003, Standard Edition**
  For smaller server applications, including file and print sharing, Internet and intranet connectivity, and centralized desktop application deployment.
- **Windows Server 2003, Enterprise Edition**
  For larger business applications, XML Web services, enterprise collaboration, and enterprise network support.
- **Windows Server 2003, Datacenter Edition**
  For business-critical and mission-critical applications demanding the highest levels of scalability and availability.
- **Windows Server 2003, Web Edition**
  For Web serving and hosting, providing a platform for developing and deploying Web services and applications.

UNIX

Originally developed by AT&T, **UNIX** now is also offered by other vendors, including Solaris by Sun Microsystems and AIX by IBM. UNIX is a multitasking, multiuser, network-managing operating system whose portability allows it to run on mainframes, midrange computers, and microcomputers. UNIX is still a popular choice for Web and other network servers.

Linux

Linux is a low-cost, powerful, and reliable UNIX-like operating system that is rapidly gaining market share from UNIX and Windows servers as a high-performance operating system for network servers and Web servers in both small and large networks. Linux was developed as free or low-cost *shareware* or *open-source software* over the Internet in the 1990s by Linus Torvald of Finland and millions of programmers around the world. Linux is still being enhanced in this way, but is sold with extra features and support services by software vendors such as Red Hat, Caldera, and SUSE Linux. PC versions, which support office software suites, Web browsers, and other application software, are also available.

The concept of **open-source software** (OSS), (discussed further in the Real World Case at the end of this chapter),, is growing far beyond the Linux operating system. The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is accustomed to the slow pace of conventional software development, seems astonishing. The open-source community of software developers has learned that this rapid evolutionary process produces better software than the traditional commercial (closed) model, in which only a very few programmers can see the source. The concept of open source is, admittedly, counter to the highly commercial (and proprietary) world of traditional software development. Nonetheless, an increasingly large number of developers have embraced the open-source concept and have come to realize that the proprietary approach to software development has hidden costs that can often outweigh its benefits.

Since 1998, the OSS movement has become a revolution in software development. This revolution, however, can actually trace its roots back more than 30 years. Typically, in the PC era, computer software had been sold only as a finished product, otherwise called a *precompiled binary*, which is installed on a user's computer by copying files to appropriate directories or folders. Moving to a new computer platform (Windows to Macintosh, for example) usually required the purchase of a new license. If the company went out of business or discontinued support of a product, users of that product had no recourse. Bug fixes were completely dependent on the organization that sold the software. By contrast, OSS is software that is licensed to guarantee free access to the programming behind the precompiled binary, otherwise called the *source code*. This allows the user to install the software on a new platform without an additional purchase and to get support (or create a support consortium with other like-minded users) for a product whose creator no longer supports it.

Those who are technically inclined can fix bugs themselves rather than waiting for someone else to do so. Generally, there is a central distribution mechanism that allows one to obtain the source code as well as precompiled binaries in some cases. There are also mechanisms for which one may pay a fee to obtain the software as well, such as on a CD-ROM or DVD, which may also include some technical support. A variety of licenses are used to ensure that the source code will remain available, wherever the code is actually used.

To be clear, there are several things open source is not—it is not shareware, public-domain software, freeware, or software viewers and readers that are made freely available without access to source code. Shareware, whether or not one registers it and pays the registration fee, typically allows no access to the underlying source code. Unlike freeware and public-domain software, OSS is copyrighted and distributed with license terms designed to ensure that the source code will always be available. While a fee may be charged for the software's packaging, distribution, or support, the complete package needed to create files is included, not simply a portion needed to view files created elsewhere.

The philosophy of open source is based on a variety of models which sometimes conflict; indeed it often seems there are as many philosophies and models for developing and managing OSS as there are major products. In 1998, a small group of open-source enthusiasts decided it was time to formalize some things about open source. The newly formed group registered themselves on the Internet as www.opensource.org and began the process of defining exactly what is, and what is not, open-source software. As it stands today, open-source licensing is defined by the following characteristics:

- The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources.
- The program must include source code and must allow distribution in source code as well as compiled form.
- The license must allow modifications and derived works and must allow them to be distributed under the same terms as the license of the original software.
- The license may restrict source code from being distributed in modified form only if the license allows the distribution of patch files with the source code for the purpose of modifying the program at build time.
- The license must not discriminate against any person or group of persons.
- The license must not restrict anyone from making use of the program in a specific field of endeavor.
- The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
- The license must not be specific to a product.
- The license must not contaminate other software by placing restrictions on any software distributed along with the licensed software.

This radical approach to software development and distribution is not without its detractors—most notably Microsoft. Nonetheless, the open-source movement is flourishing and stands to continue to revolutionize the way we think about software development.

Mac OS X

Actually based on a form of UNIX, the **Mac OS X** is the latest operating system from Apple for the iMac and other Macintosh microcomputers. The Mac OS X version 10.2 Jaguar has an advanced graphical user interface and multitasking and multimedia capabilities, along with an integrated Web browser, e-mail, instant messaging, search engine, digital media player, and many other features.

Orbitz and E*Trade: Switching to Linux

Chicago-based Orbitz Inc. (www.orbitz.com) is clear on the cost savings, enhanced processing power, and speed afforded by Linux. The online travel reservation company is using Linux on its 50 Sun Microsystems Java application servers running the Solaris UNIX operating system. These heavy-lifting systems feed the company's 700 Web servers—also running Linux—which dish up the screens customers interact with when they make airline, hotel, and vacation reservations online. Orbitz, founded in 2000 by five major U.S. airlines, currently tracks about 2 billion flight and fare options from more than 455 airlines in addition to 45,000 lodging properties and 23 rental car companies.

Orbitz benchmarked several vendors' latest operating systems including Linux on Intel servers, and the results were compelling. While maintaining the same capacity in terms of the number of users on its site, Orbitz was able to move from the UNIX servers to the Linux systems for about one-tenth the cost. As for the Web servers, Orbitz really sees the value of Linux's ease of maintenance. All 700 Web servers require only one administrator.

Orbitz isn't the only company enjoying the gains from converting to Linux. Take E*Trade Financial (www.etrade.com). In 1999, it paid $12 million for 60 Sun machines to run its online trading website. In 2002, E*Trade replaced those machines with 80 Intel-based servers running Linux for a mere $320,000. That has enabled E*Trade to bring its tech budget down 30 percent, from $330 million in 2000 to $200 million in 2002—a big reason the company has stayed alive despite the ups and downs of the stock market and the brokerage business. On top of all that, website response time has improved by 30 percent [10, 6, 14].

Other System Management Programs

There are many other types of important system management software besides operating systems. These include *database management systems*, which we will cover in Chapter 5, and *network management programs*, which we will cover in Chapter 6. Figure 4.17 compares several types of system software offered by IBM and its competitors.

FIGURE 4.17 Comparing system software offered by IBM and its main competitors.

Software Category	What It Does	IBM Product	Customers	Main Competitor	Customers
Network management	Monitors networks to keep them up and running.	**Tivoli**	T. Rowe Price uses it to safeguard customer records.	**HP OpenView**	Amazon.com uses it to monitor its servers.
Application server	Shuttles data between business apps and the Web.	**WebSphere**	REI uses it to serve up its website and distribute data.	**BEA WebLogic**	Washingtonpost.com builds news pages with it.
Database manager	Provides digital storehouses for business data.	**DB2**	Mikasa uses it to help customers find its products online.	**Oracle 9i**	It runs Southwest Airlines' frequent-flyer program.
Collaboration tools	Powers everything from e-mail to electronic calendars.	**Lotus**	Retailer Sephora uses it to coordinate store maintenance.	**Microsoft Exchange**	Time Inc. uses it to provide e-mail to its employees.
Development tools	Allows programmers to craft software code quickly.	**Rational**	Merrill Lynch used it to build code for online trading.	**Microsoft Visual Studio .NET**	Used to develop Allstate's policy management system.

Source: Adapted from Susan Orenstein, Erik Schonfeld, and Scott Herhold, "The Toughest Guy in Software," *Business 2.0*, April 2003, p. 82.

Several other types of system management software are marketed as separate programs or are included as part of an operating system. Utility programs, or **utilities**, are an important example. Programs like Norton Utilities perform miscellaneous housekeeping and file conversion functions. Examples include data backup, data recovery, virus protection, data compression, and file defragmentation. Most operating systems also provide many utilities that perform a variety of helpful chores for computer users.

Other examples of system support programs include performance monitors and security monitors. **Performance monitors** are programs that monitor and adjust the performance and usage of one or more computer systems to keep them running efficiently. **Security monitors** are packages that monitor and control the use of computer systems and provide warning messages and record evidence of unauthorized use of computer resources. A recent trend is to merge both types of programs into operating systems like Microsoft's Windows 2003 Datacenter Server, or into system management software like Computer Associates's CA-Unicenter, which can manage both mainframe systems and servers in a data center.

Another important software trend is the use of system software known as **application servers,** which provide a *middleware* interface between an operating system and the application programs of users. **Middleware** is software that helps diverse software applications and networked computer systems exchange data and work together more efficiently. Examples include application servers, Web servers, and enterprise application integration (EAI) software. Thus, for example, application servers like BEA's WebLogic and IBM's WebSphere help Web-based e-business and e-commerce applications run much faster and more efficiently on computers using Windows, UNIX, and other operating systems.

Programming Languages

To understand computer software, you need a basic knowledge of the role that programming languages play in the development of computer programs. A **programming language** allows a programmer to develop the sets of instructions that constitute a computer program. Many different programming languages have been developed, each with its own unique vocabulary, grammar, and uses.

Machine Languages

Machine languages (or *first-generation languages*) are the most basic level of programming languages. In the early stages of computer development, all program instructions had to be written using binary codes unique to each computer. This type of programming involves the difficult task of writing instructions in the form of strings of binary digits (ones and zeros) or other number systems. Programmers must have a detailed knowledge of the internal operations of the specific type of CPU they are using. They must write long series of detailed instructions to accomplish even simple processing tasks. Programming in machine language requires specifying the storage locations for every instruction and item of data used. Instructions must be included for every switch and indicator used by the program. These requirements make machine language programming a difficult and error-prone task. A machine language program to add two numbers together in the CPU of a specific computer and store the result might take the form shown in Figure 4.18.

Assembler Languages

Assembler languages (or *second-generation languages*) are the next level of programming languages. They were developed to reduce the difficulties in writing machine language programs. The use of assembler languages requires language translator programs called *assemblers* that allow a computer to convert the instructions of such language into machine instructions. Assembler languages are frequently called symbolic languages because symbols are used to represent operation codes and storage locations. Convenient alphabetic abbreviations called *mnemonics* (memory aids) and other symbols represent operation codes, storage locations, and data elements. For example, the computation $X = Y + Z$ in an assembler language might take the form shown in Figure 4.18.

FIGURE 4.18
Examples of four levels of programming languages. These programming language instructions might be used to compute the sum of two numbers as expressed by the formula X = Y + Z.

Four Levels of Programming Languages	
• **Machine Languages:** Use binary coded instructions 1010 11001 1011 11010 1100 11011	• **High-Level Languages:** Use brief statements or arithmetic notations BASIC: X = Y + Z COBOL: COMPUTE X = Y + Z
• **Assembler Languages:** Use symbolic coded instructions LOD Y ADD Z STR X	• **Fourth-Generation Languages:** Use natural and nonprocedural statements SUM THE FOLLOWING NUMBERS

Assembler languages are still used as a method of programming a computer in a machine-oriented language. Most computer manufacturers provide an assembler language that reflects the unique machine language instruction set of a particular line of computers. This feature is particularly desirable to *system programmers*, who program system software (as opposed to application programmers, who program application software), since it provides them with greater control and flexibility in designing a program for a particular computer. They can then produce more efficient software, that is, programs that require a minimum of instructions, storage, and CPU time to perform a specific processing assignment.

High-Level Languages

High-level languages (or *third-generation languages*) use instructions, which are called *statements*, that use brief statements or arithmetic expressions. Individual high-level language statements are actually *macroinstructions;* that is, each individual statement generates several machine instructions when translated into machine language by high-level language translator programs called *compilers* or *interpreters*. High-level language statements resemble the phrases or mathematical expressions required to express the problem or procedure being programmed. The *syntax* (vocabulary, punctuation, and grammatical rules) and the *semantics* (meanings) of such statements do not reflect the internal code of any particular computer. For example, the computation X = Y + Z would be programmed in the high-level languages of BASIC and COBOL as shown in Figure 4.18.

High-level languages like BASIC, COBOL, and FORTRAN are easier to learn and program than an assembler language, since they have less rigid rules, forms, and syntaxes. However, high-level language programs are usually less efficient than assembler language programs and require a greater amount of computer time for translation into machine instructions. Since most high-level languages are machine-independent, programs written in a high-level language do not have to be reprogrammed when a new computer is installed, and programmers do not have to learn a different language for each type of computer.

Fourth-Generation Languages

The term **fourth-generation language** describes a variety of programming languages that are more nonprocedural and conversational than prior languages. These languages are called fourth-generation languages (4GLs) to differentiate them from machine languages (first generation), assembler languages (second generation), and high-level languages (third generation).

Most fourth-generation languages are *nonprocedural languages* that encourage users and programmers to specify the results they want, while the computer determines the sequence of instructions that will accomplish those results. Thus, fourth-generation languages have helped simplify the programming process. **Natural languages** are sometimes considered to be *fifth-generation* languages (5GLs) and are very close to English or other human languages. Research and development activity in artificial intelligence (AI) is developing programming languages that are as easy to use as ordinary conversation in one's native tongue. For example, INTELLECT, a natural language, would use a statement like "What are the average exam scores in MIS 200?" to program a simple average exam score task.

In the early days of 4GLs, results suggested that high-volume transaction processing environments were not in the range of a 4GL's capabilities. While 4GLs were characterized by their ease of use, they were also viewed as less flexible than their predecessors, primarily due to their increased storage and processing speed requirements. In today's large data volume environment, 4GLs are widely used and no longer viewed as a trade-off between ease of use and flexibility.

Object-Oriented Languages

Object-oriented languages like Visual Basic, C++, and Java are also considered to be fifth-generation languages and have become major tools of software development. Briefly, while most other programming languages separate data elements from the procedures or actions that will be performed upon them, object-oriented languages tie them together into **objects.** Thus, an object consists of data and the actions that can be performed on the data. For example, an object could be a set of data about a bank customer's savings account and the operations (such as interest calculations) that might be performed upon the data. Or an object could be data in graphic form such as a video display window plus the display actions that might be used upon it. See Figure 4.19.

In procedural languages, a program consists of procedures to perform actions on each data element. However, in object-oriented systems, objects tell other objects to perform actions on themselves. For example, to open a window on a computer video display, a beginning menu object could send a window object a message to open, and a window would appear on the screen. That's because the window object contains the program code for opening itself.

Object-oriented languages are easier to use and more efficient for programming the graphics-oriented user interfaces required by many applications. Therefore, they are the most widely used programming languages for software development today. Also, once objects are programmed, they are reusable. Therefore, reusability of objects is a major benefit of object-oriented programming. For example, programmers can construct a user interface for a new program by assembling standard

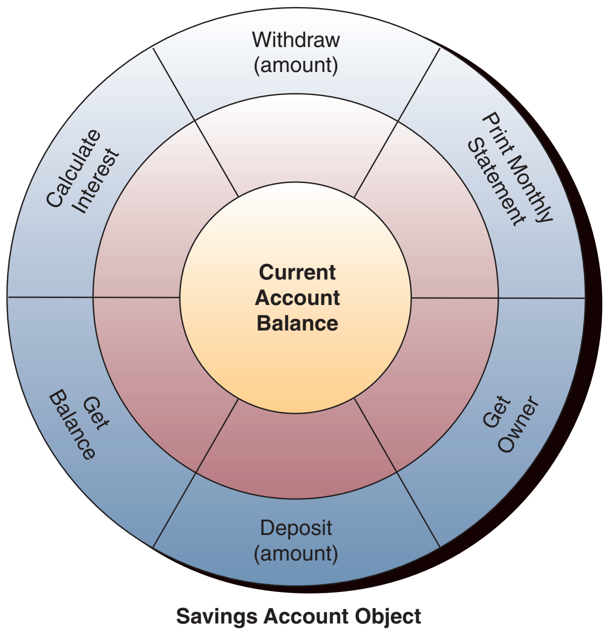
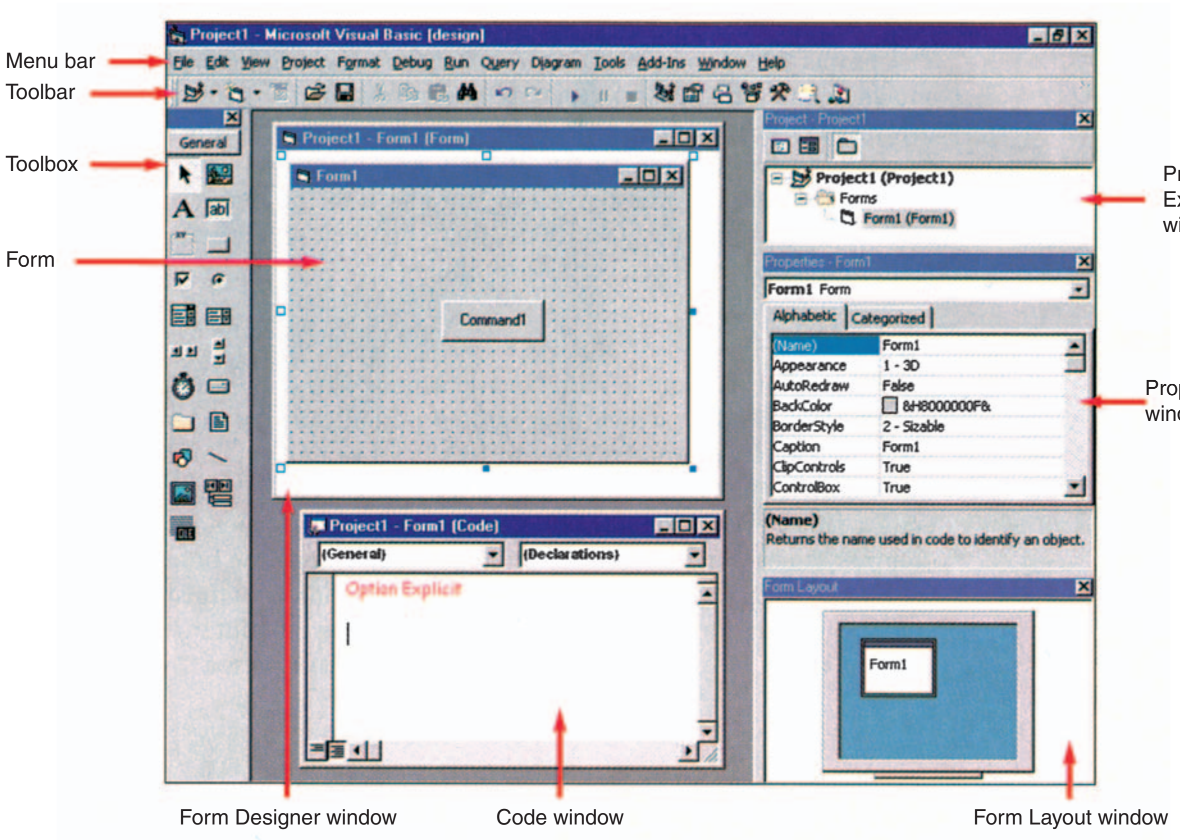


FIGURE 4.19
An example of a bank savings account object. This object consists of data about a customer's account balance and the basic operations that can be performed on those data.

FIGURE 4.20 The Visual Basic object-oriented programming environment.



(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H8000000F&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True

(Name)
Returns the name used in code to identify an object.

```
Option Explicit
```

Source: Courtesy of Microsoft Corp.

objects such as windows, bars, boxes, buttons, and icons. Therefore, most object-oriented programming packages provide a GUI that supports a point-and-click, drag-and-drop visual assembly of objects known as *visual programming*. Figure 4.20 shows a display of the Visual Basic object-oriented programming environment. Object-oriented technology is discussed further in the coverage of object-oriented databases in Chapter 5.

Web Languages and Services

HTML, XML, and Java are three programming languages that are important tools for building multimedia Web pages, websites, and Web-based applications. In addition, XML and Java have become strategic components of the software technologies that are supporting many Web services initiatives in business.

HTML

HTML (Hypertext Markup Language) is a page description language that creates hypertext or hypermedia documents. HTML inserts control codes within a document at points you can specify that create links *(hyperlinks)* to other parts of the document or to other documents anywhere on the World Wide Web. HTML embeds control codes in the ASCII text of a document that designate titles, headings, graphics, and multimedia components, as well as hyperlinks within the document.

As we mentioned earlier, several of the programs in the top software suites will automatically convert documents into HTML formats. These include Web browsers, word processing and spreadsheet programs, database managers, and presentation graphics packages. These and other specialized *Web publishing* programs like Microsoft FrontPage, Lotus FastSite, and Macromedia's DreamWeaver provide a range of features to help you design and create multimedia Web pages without formal HTML programming.

XML

XML (eXtensible Markup Language) is not a Web page format description language like HTML. Instead, XML describes the contents of Web pages (including business documents designed for use on the Web) by applying identifying tags or *contextual labels* to the data in Web documents. For example, a travel agency Web page with airline names and flight times would use hidden XML tags like "airline name" and "flight time" to categorize each of the airline flight times on that page. Or product inventory data available at a website could be labeled with tags like "brand," "price," and "size." By classifying data in this way, XML makes website information much more searchable, easier to sort, and much easier to analyze.

For example, XML-enabled search software could easily find the exact product you specify if the product data at a website had been labeled with identifying XML tags. And a website that used XML could more easily determine what Web page features its customers used and what products they investigated. Thus, XML promises to make electronic business and commerce processes a lot easier and more efficient by supporting the automatic electronic exchange of business data between companies and their customers, suppliers, and other business partners.

Java and .NET

Java is an object-oriented programming language created by Sun Microsystems that is revolutionizing the programming of applications for the World Wide Web and corporate intranets and extranets. Java is related to the C11 and Objective C programming languages, but is much simpler and more secure, and is computing-platform independent. Java is also specifically designed for real-time, interactive, Web-based network applications. Java applications consisting of small application programs, called *applets*, can be executed by any computer and any operating system anywhere in a network.

The ease of creating Java applets and distributing them from network servers to client PCs and network computers is one of the major reasons for Java's popularity. Applets can be small special-purpose application programs or small modules of larger Java application programs. Java programs are platform-independent, too—they can run on Windows, UNIX, and Macintosh systems without modification.

Microsoft's **.NET** is a collection of programming support for what are known as Web services, the ability to use the Web rather than your own computer for various services (see below). .NET is intended to provide individual and business users with a seamlessly interoperable and Web-enabled interface for applications and computing devices and to make computing activities increasingly Web browser–oriented. The .NET platform includes servers, building-block services such as Web-based data storage, and device software. It also includes Passport, Microsoft's fill-in-the-form-only-once identity verification service.

The .NET platform is expected to enable the entire range of computing devices to work together and to have user information automatically updated and synchronized on all of them. In addition, it will provide a premium online subscription service. The service will feature customized access to and delivery of products and services from a central starting point for the management of various applications (such as e-mail) or software (such as Office .NET). For developers, .NET offers the ability to create reusable modules, which should increase productivity and reduce the number of programming errors.

The full release of .NET is expected to take several years to complete, with intermittent releases of products such as a personal security service and new versions of Windows and Office that implement the .NET strategy coming on the market separately. Visual Studio .NET is a development environment that is now available, and Windows XP supports certain .NET capabilities.

The latest version of Java is Java2 Enterprise Edition (J2EE), which has become the primary alternative to Microsoft's .NET software development platform for many organizations intent on capitalizing on the business potential of Web-based applications

FIGURE 4.21 The benefits and limitations of the Java2 Enterprise Edition (J2EE) and Microsoft .NET software development platforms.

J2EE		.NET	
PROS	**CONS**	**PROS**	**CONS**
• Runs on any operating system and application server (may need adjustments). • Handles complex, high-volume, high-transaction applications. • Has more enterprise features for session management, fail-over, load balancing, and application integration. • Is favored by experienced enterprise vendors such as IBM, BEA, SAP, and Oracle. • Offers a wide range of vendor choices for tools and application servers. • Has a proven track record.	• Has a complex application development environment. • Tools can be difficult to use. • Java Swing environment's ability to build graphical user interfaces has limitations. • May cost more to build, deploy, and manage applications. • Lacks built-in support for Web services standards. • Is difficult to use for quick-turnaround, low-cost, and mass-market projects.	• Easy-to-use tools may increase programmer productivity. • Has a strong framework for building rich graphical user interfaces. • Gives developers choice of working in more than 20 programming languages. • Is tightly integrated with Microsoft's operating system and enterprise server software. • May cost less, due in part to built-in application server in Windows, unified management, less expensive tools. • Has built-in support for Web service standards.	• Framework runs only on Windows, restricting vendor choice. • Users of prior Microsoft tools and technology face a potentially steep learning curve. • New runtime infrastructure lacks maturity. • Questions persist about the scalability and transaction capability of the Windows platform. • Choice of integrated development environments is limited. • Getting older applications to run in new .NET environment may require effort.

Source: Carol Silwa, ".Net vs. Java," *Computerworld*, May 20, 2002, p. 31.

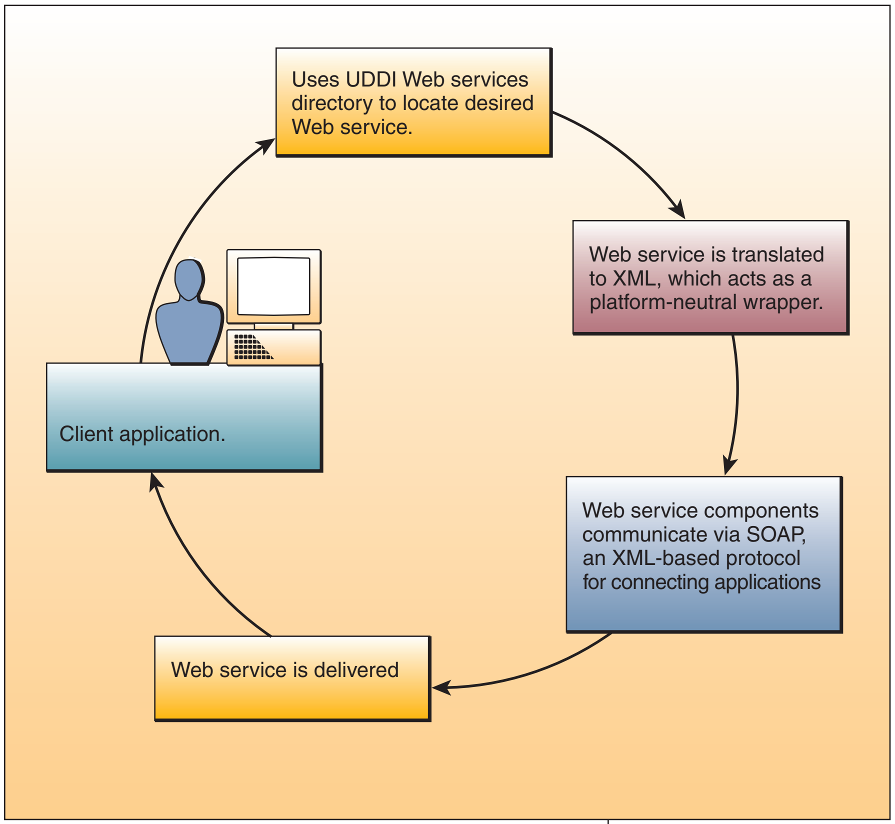
and Web services. Figure 4.21 compares the pros and cons of using J2EE and .NET for software development.

Web Services

Web services are software components that are based on a framework of Web and object-oriented standards and technologies for using the Web to electronically link the applications of different users and different computing platforms [4]. Thus, Web services can link key business functions for the exchange of data in real time within the Web-based applications a business might share with its customers, suppliers, and other business partners. For example, Web services would enable the purchasing application of a business to use the Web to quickly check the inventory of a supplier before placing a large order, while the sales application of the supplier could use Web services to automatically check the credit rating of the business with a credit-reporting agency before approving the purchase. Therefore, among both business and IT professionals, the term *Web services* is commonly used to describe the Web-based business and computing functions or services accomplished by Web services software technologies and standards.

Figure 4.22 illustrates how Web services work and identifies some of the key technologies and standards that are involved. The XML language is one of the key technologies that enable Web services to make applications work between different computing platforms. Also important are **UDDI** (Universal Description and Discovery Integration), the "yellow pages" directory of all Web services and how to locate and use them, and **SOAP** (Simple Object Access Protocol), an XML-based protocol of specifications for connecting applications to the data that they need [5].

FIGURE 4.22
The basic steps in accomplishing a Web services application.


Uses UDDI Web services directory to locate desired Web service.
Web service is translated to XML, which acts as a platform-neutral wrapper.
Web service components communicate via SOAP, an XML-based protocol for connecting applications
Web service is delivered
Client application.

Source: Adapted from Bala Iyer, Jim Freedman, Mark Gaynor, and George Wyner, "Web Services: Enabling Dynamic Business Networks," *Communications of the Association for Information Systems*, Vol. 11, 2003, p. 543.

Web services promise to be the key software technology for automating the access to data and application functions between a business and its trading partners. As companies increasingly move to doing business over the Web, Web services will become essential for the development of the easy and efficient e-business and e-commerce applications that will be required. The flexibility and interoperability of Web services will also be essential for coping with the fast-changing relationships between a company and its business partners that are commonplace in today's dynamic global business environment.

Wells Fargo & Co.: Developing Web Services

The term *Web services* is used to describe a collection of technologies—an alphabet soup of Web-based technical standards and communication protocols—such as XML, Universal Description Discovery and Integration (UDDI), and Simple Object Access Protocol (SOAP)—that link applications running on different computer platforms. Unlike present application integration approaches that require custom coding or expensive middleware to link individual applications, Web services aim to expose and link key functions within applications (such as the ability to see the balance in your checking account or place an order from a factory) to other applications that need them to complete business processes.

An increasing number of businesses have begun using Web services technologies developed by IBM, Microsoft, BEA Systems, and many others. Once you see one in action, you immediately see what the buzz is all about.

Wells Fargo & Co., a leading financial services provider, is using Web services to help streamline the process of initiating electronic transactions with wholesale banking customers. Their new online system, which replaced an internally developed system called Payment Manager, allows scores of customers to send data and instructions regarding wire transfers and automated clearinghouse transactions, among many other functions. While Payment Manager worked well, it required custom development to tie Wells Fargo systems to every customer who wanted to transact business electronically.

Using Web services, Wells Fargo doesn't have to custom-code at all. Now they can reuse code already built, thus making their online system easier to maintain and making it easier to add new features.

Key Web services protocols in use at Wells Fargo—including SOAP and UDDI—present standard interfaces that applications can be coded to, facilitating data exchanges between disparate applications within the bank as well as with customers and partners. SOAP uses XML syntax to send commands between applications across the Internet. UDDI defines a universal registry or catalog of Web services. It lets software automatically discover and integrate with services on the Web when needed.

With Web services, Wells Fargo is able to accept roughly 50 different formats of files from customers; popular file types include SAP and J.D. Edwards ERP file formats. Those files are typically sent using file transfer protocol (FTP). Some of the largest customers have a direct leased line into Wells Fargo, and they send information directly into the Payment Manager transaction hub. If the file format and instructions call for a wire transfer, the information is sent to the Wells Fargo wire transfer system. Alternatively, the files could need to be sent into Wells Fargo's automated clearinghouse systems. If the file calls for cutting a check, Wells Fargo can convert the request into a written check and send it out through a third-party check processor.

Among the possibilities for future Web services are an event-notification application whereby Wells Fargo could, for example, notify customers of the arrival of information they have been awaiting. That's an improvement over the current method of customers having to log into a Wells Fargo website and find the information they're looking for [8, 12].

Programming Software

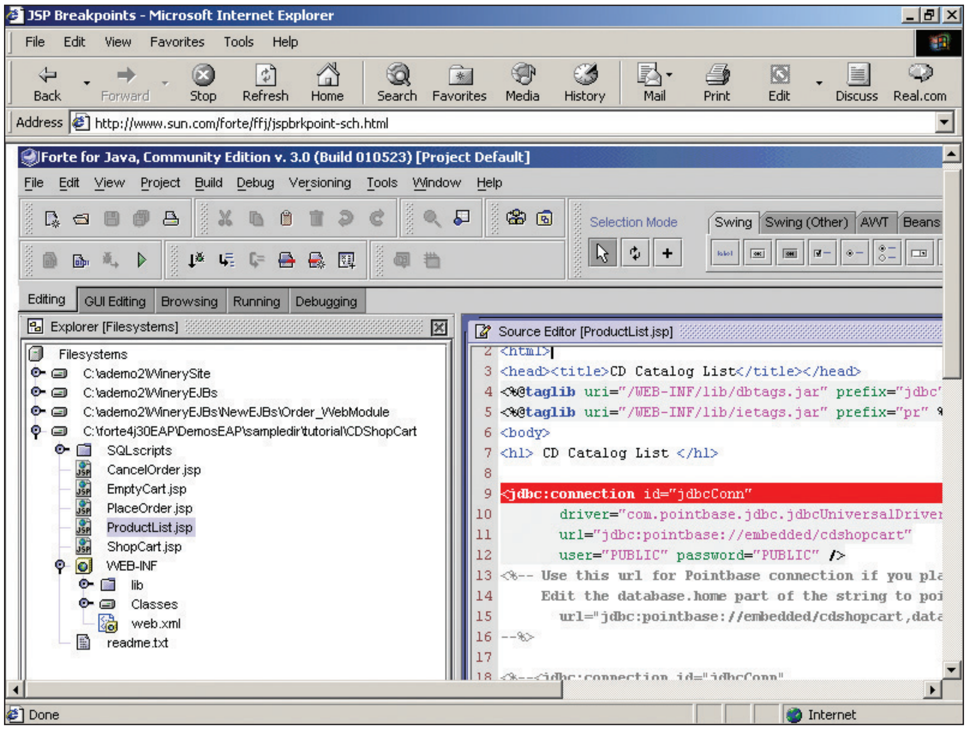
Various software packages are available to help programmers develop computer programs. For example, *programming language translators* are programs that translate other programs into machine language instruction codes that computers can execute. Other software packages, such as programming language editors, are called *programming tools* because they help programmers write programs by providing a variety of program creation and editing capabilities. See Figure 4.23.

Language Translator Programs

Computer programs consist of sets of instructions written in programming languages that must be translated by a **language translator** into the computer's own machine language before they can be processed, or executed, by the CPU. Programming language translator programs (or *language processors*) are known by a variety of names. An **assembler** translates the symbolic instruction codes of programs written in an assembler language into machine language instructions, while a **compiler** translates high-level language statements.

An **interpreter** is a special type of compiler that translates and executes each statement in a program one at a time, instead of first producing a complete machine language program, as compilers and assemblers do. Java is an example of an interpreted

FIGURE 4.23
Using the graphical programming interface of a Java programming tool, Forte for Java, by Sun Microsystems.


JSP Breakpoints - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit Discuss Real.com
Address http://www.sun.com/forte/ffj/jspbrkpoint-sch.html
Forte for Java, Community Edition v. 3.0 (Build 010523) [Project Default]
File Edit View Project Build Debug Versioning Tools Window Help
Selection Mode
Swing Swing (Other) AWT Beans
Editing GUI Editing Browsing Running Debugging
Explorer [Filesystems]
Source Editor [ProductList.jsp]
Done
Internet

```
Filesystems
  C:\ademo2\WinerySite
  C:\ademo2\WineryEJBs
  C:\ademo2\WineryEJBs\NewEJBs\Order_WebModule
  C:\forte4j30EAP\DemosEAP\sampledir\tutorial\CDShopCart
    SQLscripts
    CancelOrder.jsp
    EmptyCart.jsp
    PlaceOrder.jsp
    ProductList.jsp
    ShopCart.jsp
    WEB-INF
      lib
      Classes
      web.xml
    readme.txt
```

```
<html>
<head><title>CD Catalog List</title></head>
<%@taglib uri="/WEB-INF/lib/dbtags.jar" prefix="jdbc"
<%@taglib uri="/WEB-INF/lib/ietags.jar" prefix="pr" %
<body>
<hl> CD Catalog List </hl>

<jdbc:connection id="jdbcConn"
        driver="com.pointbase.jdbc.jdbcUniversalDriver
        url="jdbc:pointbase://embedded/cdshopcart"
        user="PUBLIC" password="PUBLIC" />
<%-- Use this url for Pointbase connection if you pla
      Edit the database.home part of the string to poi
        url="jdbc:pointbase://embedded/cdshopcart,data
--%>

<%--<jdbc:connection id="jdbcConn"
```

Source: Courtesy of Sun Microsystems.

language. Thus, the program instructions in Java applets are interpreted and executed *on the fly* as the applet is being executed by a client PC.

Programming Tools

Software development and the computer programming process have been enhanced by adding *graphical programming interfaces* and a variety of built-in development capabilities. Language translators have always provided some editing and diagnostic capabilities to identify programming errors or *bugs*. However, most software development programs now include powerful graphics-oriented *programming editors* and *debuggers*. These **programming tools** help programmers identify and minimize errors while they are programming. Such programming tools provide a computer-aided programming environment. This decreases the drudgery of programming while increasing the efficiency and productivity of software developers. Other programming tools include diagramming packages, code generators, libraries of reusable objects and program code, and prototyping tools. All of these programming tools are an essential part of widely used programming languages like Visual Basic, C11, and Java.

CASE Tools

Since the early days of programming, software developers have needed automated tools. Initially the concentration was on program support tools such as translators, compilers, assemblers, macro processors, and linkers and loaders. However, as computers became more powerful and the software that ran on them grew larger and more complex, the range of support tools began to expand. In particular, the use of interactive time-sharing systems for software development encouraged the development of program editors, debuggers, and code analyzers.

As the range of support tools expanded, manufacturers began integrating them into a single application using a common interface. Such tools were referred to as **CASE tools** (computer-aided software engineering).

CASE tools can take a number of forms and can be applied at different stages of the software development process. Those CASE tools that support activities early in the life cycle of a software project (such as requirements and design support tools) are sometimes called *front-end* or *upper* CASE tools. Those that are used later in the life cycle (such as compilers and test support tools) are called *back-end* or *lower* CASE tools.

Exploring the details of CASE tools is beyond the scope of this text, and you will encounter them again when you study systems analysis and design. For now, CASE is an important part of resolving the problems of complex application development and maintenance of software applications.

Summary

- **Software.** Computer software consists of two major types of programs: (1) application software that directs the performance of a particular use, or application, of computers to meet the information processing needs of users, and (2) system software that controls and supports the operations of a computer system as it performs various information processing tasks. Refer to Figure 4.2 for an overview of the major types of software.
- **Application Software.** Application software includes a variety of programs that can be segregated into general-purpose and application-specific categories. General-purpose application programs perform common information processing jobs for end users. Examples are word processing, electronic spreadsheet, and presentation graphics programs. Application-specific programs accomplish information processing tasks that support specific business functions or processes, scientific or engineering applications, and other computer applications in society.
- **System Software.** System software can be subdivided into system management programs and system development programs. System management programs manage the hardware, software, network, and data resources of a computer system during its execution of information processing jobs. Examples of system management programs are operating systems, network management programs, database management systems, system utilities, application servers, and performance and security monitors. Network management programs support and manage telecommunications activities and network performance telecommunications networks. Database management systems control the development, integration, and maintenance of databases. Utilities are programs that perform routine computing functions, such as backing up data or copying files, as part of an operating system or as a separate package. System development programs like language translators and programming editors help IS specialists develop computer programs to support business processes.
- **Operating Systems.** An operating system is an integrated system of programs that supervises the operation of the CPU, controls the input/output storage functions of the computer system, and provides various support services. An operating system performs five basic functions: (1) a user interface for system and network communications with users, (2) resource management for managing the hardware resources of a computer system, (3) file management for managing files of data and programs, (4) task management for managing the tasks a computer must accomplish, and (5) utilities and other functions that provide miscellaneous support services.
- **Programming Languages.** Programming languages are a major category of system software. They require the use of a variety of programming packages to help programmers develop computer programs, and language translator programs to convert programming language instructions into machine language instruction codes. The five major levels of programming languages are machine languages, assembler languages, high-level languages, fourth-generation languages, and object-oriented languages. Object-oriented languages like Java and special-purpose languages like HTML and XML are being widely used for Web-based business applications and services.

Key Terms and Concepts

These are the key terms and concepts of this chapter. The page number of their first explanation is given in parentheses.

1. Application service provider (130)
2. Application software (118)
3. Assembler language (141)
4. CASE tools (149)
5. Custom software (118)
6. COTS software (118)
7. Desktop publishing (127)
8. E-mail (124)
9. Fourth-generation language (142)

10. Function-specific application programs (121)
11. General-purpose application programs (118)
12. Groupware (129)
13. High-level language (142)
14. HTML (144)
15. Instant messaging (125)
16. Integrated package (123)
17. Java (145)
18. Language translator (148)
19. Machine language (141)
20. Middleware (141)
21. Multitasking (137)
22. Natural language (142)
23. Object-oriented language (143)
24. Operating system (133)
25. Personal information manager (129)
26. Presentation graphics software (128)
27. Programming language (141)
28. Software suites (123)
29. Spreadsheet package (127)
30. System software (133)
31. User interface (133)
32. Utility programs (141)
33. Virtual memory (136)
34. Web browser (124)
35. Web services (146)
36. Word processing software (126)
37. XML (145)

Review Quiz

Match one of the previous key terms and concepts with one of the brief examples or definitions that follow. Try to find the best fit for answers that seem to fit more than one term or concept. Defend your choices.

____ 1. Programs that manage and support the operations of computers.

____ 2. Programs that direct the performance of a specific use of computers.

____ 3. A system of programs that manages the operations of a computer system.

____ 4. Companies that own, operate, and maintain application software and the computer system resources for a fee as a service over the Internet.

____ 5. Integrated software tool that supports the development of software applications.

____ 6. Software designed in-house for use by a specific organization or set of users.

____ 7. The function that provides a means of communication between end users and an operating system.

____ 8. Acronym meaning commercial off-the-shelf.

____ 9. Provides a greater memory capability than a computer's actual memory capacity.

____ 10. The ability to do several computing tasks concurrently.

____ 11. Still the most common programming language used on the Web.

____ 12. Converts numeric data into graphic displays.

____ 13. Translates high-level instructions into machine language instructions.

____ 14. Performs housekeeping chores for a computer system.

____ 15. A category of application software that performs common information processing tasks for end users.

____ 16. Software available for the specific applications of end users in business, science, and other fields.

____ 17. Helps you surf the Web.

____ 18. Use your networked computer to send and receive messages.

____ 19. Creates and displays a worksheet for analysis.

____ 20. Allows you to create and edit documents.

____ 21. You can produce your own brochures and newsletters.

____ 22. Helps you keep track of appointments and tasks.

____ 23. A program that performs several general-purpose applications.

____ 24. A combination of individual general-purpose application packages that work easily together.

____ 25. Software to support the collaboration of teams and workgroups.

____ 26. Uses instructions in the form of coded strings of ones and zeros.

____ 27. Uses instructions consisting of symbols representing operation codes and storage locations.

____ 28. Uses instructions in the form of brief statements or the standard notation of mathematics.

____ 29. Might take the form of query languages and report generators.

____ 30. Languages that tie together data and the actions that will be performed upon the data.

____ 31. As easy to use as one's native tongue.

____ 32. Includes programming editors, debuggers, and code generators.

____ 33. Produces hyperlinked multimedia documents for the Web.

____ 34. A Web document content description language.

____ 35. A popular object-oriented language for Web-based applications.

____ 36. Windows is an example of a common one of these used on most PCs.

____ 37. Software that helps diverse applications work together.

____ 38. Enables you to communicate and collaborate in real time with the online associates in your workgroup.

____ 39. Links business functions within applications for the exchange of data between companies via the Web.

Discussion Questions

1. What major trends are occurring in software? What capabilities do you expect to see in future software packages?
2. How do the different roles of system software and application software affect you as a business end user? How do you see this changing in the future?
3. Refer to the Real World Case on Microsoft vertical market software in the chapter. Does Microsoft's entry into industry-specific applications signal the end for smaller industry-specific software developers? What changes in strategy by such developers are necessary to compete with Microsoft?
4. Why is an operating system necessary? That is, why can't an end user just load an application program into a computer and start computing?
5. Should a Web browser be integrated into an operating system? Why or why not?
6. Refer to the Real World Case on webtop software in the chapter. What are the business incentives that lure people to start a small company to develop and market webtop software applications? Would you invest in such a company today, or any other variation in that market space? Why or why not?
7. Are software suites, Web browsers, and groupware merging together? What are the implications for a business and its end users?
8. How are HTML, XML, and Java affecting business applications on the Web?
9. Do you think Linux will surpass, in adoption and use, other operating systems for network and Web servers? Why or why not?
10. Which application software packages are the most important for a business end user to know how to use? Explain the reasons for your choices.

Analysis Exercises

Complete the following exercises as individual or group projects that apply chapter concepts to real world business situations.

1. Desktop Application Recognition

Tool Selection

ABC Department Stores would like to acquire software to do the following tasks. Identify what software packages they need.

a. Surf the Web and their intranets and extranets.
b. Send messages to each other's computer workstations.
c. Help employees work together in teams.
d. Use a group of productivity packages that work together easily.
e. Help sales reps keep track of meetings and sales calls.
f. Type correspondence and reports.
g. Analyze rows and columns of sales figures.
h. Develop a variety of graphical presentations.

2. Y2K Revisited

The End of Time

Decades ago, programmers trying to conserve valuable storage space shortened year values to two digits. This created what became known as the "Y2K" problem or "millennium bug" at the turn of the century. Programmers needed to review billions of lines of code to ensure important programs would continue to operate correctly. The Y2K problem merged with the Dot-com boom and created a tremendous demand for information technology employees. Information system users spent billions of dollars fixing or replacing old software. The IT industry is only now beginning to recover from the postboom slump. Could such hysteria happen again? It can, and it likely will.

Today, most programs use several different schemes to record dates. One scheme, POSIX time, widely employed on UNIX-based systems, requires a signed 32-bit integer to store a number representing the number of seconds since January 1, 1970. "0" represents midnight on January 1, "10" represents 10 seconds after midnight, and "–10" represents 10 seconds *before* midnight. A simple program then converts this into any number of international date formats for display. This scheme works well because it allows programmers to subtract one date/time from another date/time and directly determine the interval between them. It also requires only 4 bytes of storage space. But 32 bits still calculates to a finite number, whereas time is infinite. As a business manager, you will need to be aware of this new threat and steer your organization away from repeating history. The following questions will help you evaluate the situation and learn from history.

a. Since 1 represents 1 second and 2 represents 2 seconds, how many seconds can be represented in a binary number 32 bits long? Use a spreadsheet to show your calculations.

b. Given that POSIX time starts at midnight, January 1, 1970, in what year will time "run out"? Remember that half the available numbers represent dates before 1970. Use a spreadsheet to show your calculations.

c. As a business manager, what can you do to minimize this problem for your organization?

3. Tracking Project Work

Queries and Reports

You are responsible for managing information systems development projects at AAA Systems. To better track progress in completing projects, you have decided to maintain a simple database table to track the time your employees spend on various tasks and the projects with which they are associated. It will also allow you to keep track of employees' billable hours each week. The table below provides a sample data set.

a. Build a database table to store the data shown and enter the records as a set of sample data.

b. Create a query that will list the hours worked for all workers who worked more than 40 hours during production week 20.

c. Create a report grouped by project that will show the number of hours devoted to each task on the project and the subtotal number of hours devoted to each project as well as a grand total of all hours worked.

d. Create a report grouped by employee that will show each employee's hours worked on each task and total hours worked. The user should be able to select a production week and have data for just that week presented.

4. Matching Training to Software Use

3-D Graphing

You have responsibility for managing software training for Sales, Accounting, and Operations Department workers in your organization. You have surveyed the workers to get a feel for the amounts of time spent using various packages, and the results are shown below. The values shown are the total number of workers in each department and the total weekly hours the department's workers spend using each software package. You have been asked to prepare a spreadsheet summarizing these data and comparing the use of the various packages across departments.

Department	Employees	Spreadsheet	Database	Presentations
Sales	225	410	1,100	650
Operations	75	710	520	405
Accounting	30	310	405	50

a. Create a spreadsheet illustrating each application's average use per department. To do this, you will first enter the data shown above. Then compute the average weekly spreadsheet use by dividing spreadsheet hours by the number of Sales workers. Do this for each department. Repeat these three calculations for both database and presentation use. Round results to the nearest 1/100th.

b. Create a three-dimensional bar graph illustrating the averages by department and software package.

c. A committee has been formed to plan software training classes at your company. Prepare a slide presentation with four slides illustrating your findings. The first slide should serve as an introduction to the data. The second slide should contain a copy of the original data table (without the averages). The third slide should contain a copy of the three-dimensional bar graph from the previous answer. The fourth slide should contain your conclusions regarding key applications per department. Use professional labels, formatting, and backgrounds.

Project_Name	Task_Name	Employee_ID	Production_Week	Hours_Worked
Fin-Goods-Inv	App. Devel.	456		
Fin-Goods-Inv	DB Design	345		20
Fin-Goods-Inv	UI Design	234		16
HR	Analysis	234		24
HR	Analysis	456		48
HR	UI Design	123		8
HR	UI Design	123		40
HR	UI Design	234		32
Shipmt-Tracking	DB Design	345		24
Shipmt-Tracking	DB Design	345		16
Shipmt-Tracking	DB Development	345		20
Shipmt-Tracking	UI Design	123		32
Shipmt-Tracking	UI Design	234		24

REAL WORLD CASE 3

Amazon and eBay: The New Face of Web Services in Business

Amazon.com Inc. remains as protective as ever of the technology that powers its website. "We don't go into detail about what our underlying infrastructure looks like," says Chief Technology Officer Al Vermeulen. At the same time, though, Amazon is throwing open its site to outside programmers, providing access to databases and features that have taken years and something approaching $1 billion to develop over nearly a decade. "We're going to go full bore in exposing our entire platform," Vermeulen says enthusiastically. Why secretive one minute but open the next? Amazon has figured out that all that bottled-up intellectual property becomes even more valuable once outsiders get their hands on it. In mid-2003, Amazon took its first step to create a "programmable website" when it launched Amazon Web Services 1.0, a set of programming protocols that provide third-party programmers and fellow retailers with access to some of its data and basic website functionality.

The idea proved such a hit that more than 50,000 programmers have signed up. In late 2004, Amazon introduced Amazon Web Services 4.0, which opened its data fields even more.

Amazon and eBay Inc. have established themselves as hubs of commerce when it comes to consumer online shopping. Now they're emerging as hubs of software development activity, where openness breeds innovation and innovation generates sales. "We think Web services feeds directly into making that flywheel spin faster," Vermeulen says.

In 2003, eBay took a small, invitation-only developer program that had been operating since 2000 and opened it to the public, making the code to its e-commerce software available for download and supplying a software development kit for it that works with popular developer tools from Borland, Microsoft, and companies that build tools based on the Java programming language.

Today around 8,000 companies or individuals have become members of the development program, and about 600 applications built by independent developers use eBay's servers. One billion times a month—more than 30 million times a day—an application makes a hit on eBay's database.

Amazon and eBay, in turning themselves into software development hubs, are once again expanding the possibilities—and increasing the pressure—for any company that wants to be a center of e-commerce. As Amazon and eBay popularize the use of programmable websites, other e-businesses might find they, too, want to open their websites to a community of developers—be they independents, or programmers from customers or business partners who want to add their own innovations to a site. "There's no reason we can't have thousands of developer communities for thousands of different websites, even on a small scale," says Jeff Barr, Amazon's technical program manager. Other businesses will find the hard part isn't exposing the inner workings of a site as Web services but establishing a business model that works with it. "There has to be real sound financial return for doing these things over the long term," he says.

Google Inc., the search-engine powerhouse that's enjoying the benefits of a highly successful initial public stock offering, publishes on its website programming protocol that lets users write applications that incorporate Google's search engine. The company doesn't have formal developer programs like eBay and Amazon do, but it's a start.

Oddcast Inc. is typical of the kind of company that's helping to make eBay a hub of development. The five-year-old software company develops interactive characters that act as website guides for clients such as Coca-Cola, Intel, and McDonald's. Last month, it started selling an online capability called "publish to eBay," so someone can license Oddcast's VHost SitePal software and design an online character that's published directly to that person's eBay auction site. "Never in a million years would this have been developed by eBay for its customers," Oddcast CTO Gil Sideman says.

Here's a measure of how important this open approach has become to eBay: About 40 percent of the items listed for sale on eBay's U.S. site come in through its programming interface. That means two of five products are loaded onto the site software-to-software, rather than manually posted using the more common browser-based approach. Major retailers are taking advantage of these tools, and software companies are hustling to make their tools fit the model.

To make this hub-of-development concept work, Amazon and eBay needed to learn how to inspire clever programmers to work on their platforms. They're succeeding in part because they have the kind of user numbers that interest programmers. But they're also presenting programmers with a new challenge in the world of Web services: tools and technology for integrating Web platforms. The companies are fast movers when it comes to exposing the capabilities of their platforms at a time when many companies still are cautious about Web services technology. They're showing that opening up some of their technology vaults can spur the creation of other software apps that expand their market reach.

Working together, Amazon and Microsoft created a downloadable plug-in that works with Microsoft's Office 2003 desktop suite. Called the Office 2003 Research and Reference Pane, the code makes it possible for users to launch product searches from within, say, Microsoft Word, and to cut and paste the information they get into documents. It wasn't the only time Amazon and Microsoft collaborated on Web services—last fall, Amazon CTO Vermeulen demonstrated similar integration using Microsoft's next-generation Vista operating system.

So just what gets dished up by one of Amazon's Web services? The output includes product details, search capabilities, customer reviews, sales rankings, wish lists, and registries. Amazon gives programmers the option of choosing "lite" or heavy versions of those categories, depending on their needs. Its ground rules: Programmers must link to the Amazon site, pricing data can only be stored for an hour, data can't be resold, and applications must be written so that they don't make more than one call per second to the Amazon site.

Clearly, the developers writing software to integrate with the e-commerce engines of the biggest websites are more than just fanatics doing it for fun. For one thing, there's a careful vetting process of developers who can write commercial apps that use the companies' program code. Oddcast's Sideman, for example, says eBay's detailed questionnaire took him half a day to fill out, including listing technical and business information about the company, citing examples of its products in use, and describing what technology would touch eBay's IT environment. A technical reviewer phoned two days later with more questions.

Does all this have any effect on sales? And is there real money to be made here? Oddcast's eBay feature costs $3 a month, on top of a $10 monthly charge to use SitePal, which has 1,500 customers. "We're still struggling with analyzing the potential models," Sideman says. eBay spokesman Jeff McManus also sidesteps the question of whether the company's investment in its developer program produces a tangible return. "We watch that, but it's fuzzy," he says. "There's a little crystal-ball factor in that. What's often more important than accelerating business is saving time." Amazon's Vermeulen, however, is unequivocal: "There's clearly been positive ROI," he says, pointing to increases in both the number of products sold on the site and shoppers.

Amazon and eBay know there's hard work to do to get the model and business rules right. But they show no sign of easing up on plans to become as much a destination for developers as they are a destination for shoppers.

Source: Adapted from Aaron Ricadela and John Foley, "New Face of E-Commerce," *InformationWeek*, July 26, 2004. Copyright © 2004 CMP Media LLC.

CASE STUDY QUESTIONS

1. What are the purpose and business value of Web services?
2. What are the benefits of Web services to Amazon, eBay, and their developer partners?
3. What are the business challenges of Web services? Visit the Web services websites of IBM (www.ibm.com/solutions/webservices) and Microsoft (www.microsoft.com/webservices) to help with your answer.

REAL WORLD ACTIVITIES

1. The concept of Web services and the opportunities they provide are growing everyday. Using the Internet, see if you can find ways in which companies are using Web services beyond those listed in the case.
2. Being able to integrate one organization's website with another's poses some interesting questions of privacy, intellectual property protection, and technical challenges. Break into small groups with your classmates and discuss these issues. Do you think there is any risk associated with this type of cooperation?

REAL WORLD CASE 4

Linux and the Growth of Open-Source Software: IT Managers Evaluate the Value and the Risks

A senior IT executive at a major West Coast bank scoffs at the notion of using open-source software. She recites a litany of fears: It's unsafe and unreliable; it won't fly with government regulators; and it lacks a support infrastructure. However, her fears are unfounded—and the odds are actually quite high that her bank already deploys open-source software as part of its Internet infrastructure. She just isn't aware of it yet.

The growth of Linux has thrown the spotlight on open-source software. Linux, the most talked-about open-source product today, is a totally reengineered version of the UNIX operating system.

Linux server shipments grew more than 24 percent in 2001, according to International Data Corp. Some analysts predict Linux will account for almost half of all business servers by 2004. The widespread adoption of the Apache Web server, Sendmail, and Perl scripting language—all open-source products—further illustrates the inroads open-source software has made in large companies.

Apache represents 60 percent of the Web servers on the Internet, while Sendmail accounts for 40 percent of the e-mail servers on the Internet, says Jean-Christophe Cimetriere, CEO of TechMetrix Research, a technology consulting firm. It would be difficult to find a company active on the Internet that doesn't have some open-source products in place.

Although a growing number of brick-and-mortar companies such as Kinko's, Plantronics, and the Raychem division of Tyco are adopting open-source products, many business IT executives appear to be unaware of the growing acceptance of such products—or they're just in denial.

Open source understandably raises concerns about support, accountability, and viability. It upsets commonly held ideas about licensing and turns the conventional software-value proposition on its head. Who, IT managers may wonder, do you hold accountable for a problem with Linux?

Open-source software, also referred to as free software, is a confusing concept that, indeed, flies in the face of everything business IT executives have learned about software development, licensing, value, and quality. At the simplest level, open source refers to software that's delivered with unrestricted access to its source code. Even before anyone adopts a product, people can look at the source code and are free to modify it.

Open source typically is provided under a license. The GNU Public License (GPL) is the most common, but there are other open-source licenses that offer slightly different terms, such as BSD (Berkeley Software Distribution) and Artistic license, which covers Perl.

The open-source license generally grants the right to run the program, own a copy of the program's source code, modify the program's source code, and distribute copies of the programs you build using the open-source code. The only thing you usually can't do is fold an open-source program into a program you're licensing under a proprietary license.

Whatever a person builds using open-source software, he or she must provide the same capabilities to anyone else under the same open-source license terms. The differences between the various open-source licenses typically have to do with scope and constraints of software usage.

Open-source software is free in the sense of freedom to access the source code and do with it whatever you want, within the constraints of the particular open-source license. While you don't pay for the open-source code itself—the code must be available for the asking—you'll likely pay for the packaging of the code, support, documentation, training, and a host of other items that accompany the source code.

"We charge what's called a software fee," explains Bonnie Crater, president of Zelerate Inc. The San Mateo, California, company offers an open-source E-commerce package under the GNU Public License. As such, Zelerate can't charge a license fee for the source code. Instead, its software fee covers things such as support and warranty.

Although open source has been around in various guises for two decades or more, the open-source phenomenon first began attracting attention with the success of Linux. Eric Raymond, an early contributor to GNU and developer of Fetchmail, an open-source product, characterized Linux development as "release early and often, delegate everything you can, and be open to the point of promiscuity," in what amounts to a seemingly unfettered bazaar.

The Linux phenomenon provided a stark contrast to the classic IT approach to software, which Raymond describes as reverent, highly controlled cathedral-building in his 1997 paper, "The Cathedral and the Bazaar."

The paper drew large amounts of attention online, generated heated debate, and essentially split the software world into two camps: the denizens of the freewheeling bazaar who embraced open source, and the IT priests who opted for the cathedral.

Although nobody at that time seemed ready to bet their core enterprise resource planning system on open source (SAP, for example, now runs on Linux), IT managers were, often unknowingly, trusting their Web and directory servers and all manner of Internet infrastructure to open-source systems. Even now, misconceptions about open-source systems continue to persist among IT executives. For instance, people mistakenly think that open-source software is the work of high school and college students. In truth, professional system administrators and developers from major global companies who are trying to solve real problems do much of the open-source work.

Another misconception revolves around security. Open-source contributors are perceived as hackers intent on creating security holes, making open source inherently insecure. On the contrary, the nature of open-source development, by which everyone sees every line of code, goes a long way toward removing security threats. "If somebody sees a security problem, it gets fixed fast by the open-source community, often within hours," Crater says. By comparison, proprietary software products suffer from security holes that the vendor may be reluctant to disclose and slow to fix—Microsoft is an obvious case.

The same can be said about open-source reliability. The open-source process subjects code to the most rigorous code review imaginable—thousands of developers, for instance, review every line of Linux. No proprietary software vendor subjects its code to this level of scrutiny.

If anything, open source is more reliable than any proprietary software on the market, says Jim Johnson, chairman of the Standish Group, a research and consulting firm in West Yarmouth, Massachusetts. Standish Group research shows Linux servers have about 14 hours of downtime per year, amounting to 99.6 percent uptime during average peak operational periods.

By comparison, Standish found the average Microsoft Enterprise Cluster to have just over 99 percent uptime or about 30 hours of downtime during the average peak operational period. In terms of actual hours, the Microsoft server cluster will be down more than twice as long as a Linux server.

With IBM, Sun Microsystems, and other major computer companies offering Linux support, IT managers will find contract support options that rival anything a proprietary software vendor can provide. Similarly, companies will find a growing array of open-source apps, from Lutris Technologies' Enhydra application server and Zelerate's E-commerce suite to Sun's StarOffice desktop suite.

However, corporate IT departments have some legitimate concerns about open source in general and Linux in particular. "Linux is another UNIX, and large enterprises already have a lot of UNIX. It will cost them money and effort to add another UNIX platform," Johnson says.

If IT managers can get over their hang-ups about open source, they can experience the benefits the Standish Group found: highly secure, reliable, flexible software at a fraction of the cost of conventional offerings. With ready access to the source code, they no longer have to wait until some vendor decides to make an enhancement that they consider important.

Johnson is convinced this represents the future for an increasingly large proportion of company systems. The only exceptions are those must-not-fail systems too critical to be trusted to Microsoft, Linux, or any general platform, proprietary or open. These, instead, require one of the very high-end, fault-tolerant platforms. "We now feel more comfortable with Linux than with Microsoft," Johnson says. "But until you've tried open source, you can't know."

Source: Adapted from Alan Radding, "Open-Source Software: IT Managers Are Becoming More 'Open'-Minded," *InformationWeek*, April 16, 2001. Copyright © 2005 CMP Media LLC.

CASE STUDY QUESTIONS

1. What are the business benefits of adopting open-source software?
2. What are the risks associated with open-source software? How can these risks be addressed?
3. Do you see open-source software eventually replacing the current proprietary software model? Explain your answer.

REAL WORLD ACTIVITIES

1. A wide variety of organizations have been formed to advance the open-source initiative. Using the Internet, see if you can find information on these open-source advocate organizations. A good place to start is www.opensource.org.
2. Supporters as well as detractors of open-source operating systems such as Linux are quite passionate about their feelings. Break into small groups with your classmates, and discuss the advantages and disadvantages of open-source applications. If any of your classmates have experience with systems such as Linux, ask them to explain their feelings and experiences.