

BUILDING E-COMMERCE APPLICATIONS AND INFRASTRUCTURE

Content

Helping Customers Navigate the Web Site and Increase Sales at Campmor

- 19.1 Major E-Commerce Applications and Their Functionalities
- 19.2 A Five-Step Approach to Developing an E-Commerce Landscape
- 19.3 Development Options for E-Commerce Applications
- 19.4 Criteria for Selecting a Development Approach
- 19.5 E-Commerce Software Packages and Suites
- 19.6 Connecting to Databases and Other Enterprise Systems
- 19.7 Vendor and Software Selection
- 19.8 Usage Analysis and Site Management

Managerial Issues

Real-World Case: Molding a New Vision for E-Commerce at D-M-E

Learning Objectives

Upon completion of this chapter, you will be able to:

1. Discuss the major steps in developing an EC application.
2. Describe the major EC applications and list their major functionalities.
3. List the major EC application development options along with their benefits and limitations.
4. Discuss various EC application outsourcing options, including application service providers (ASPs), software as a service (SaaS), and utility computing.
5. Discuss the major EC software packages and EC application suites.
6. Describe various methods for connecting an EC application to back-end systems and databases.
7. Discuss the value and technical foundation of Web Services and their evolution into second-generation tools in EC applications.
8. Understand service-oriented architecture (SOA) and virtualization and their relationship to EC application development.
9. Describe the criteria used in selecting an outsourcing vendor and package.
10. Understand the value and uses of EC application log files.
11. Discuss the importance of usage analysis and site management.

HELPING CUSTOMERS NAVIGATE THE WEB SITE AND INCREASE SALES AT CAMPMOR

The Problem

Nature enthusiasts no longer have to make a major hike to take advantage of the vast array of camping, fishing, rock-climbing, and canoeing supplies housed within Campmor's walls. Since discovering—and then tapping—the possibilities of a well-designed e-commerce site, the Paramus, New Jersey, retailer has expanded to the four corners of the earth. But doing so was no walk in the park. Having struck out on its own, Campmor eventually realized it needed a guide or two who were familiar with the daunting wilderness that the Internet can be. Although an early adopter of e-commerce, Campmor realized that its Web site could be much more effective. The company turned to its long-time systems integrator, Tachyon Solutions, to use site analytics to redesign the site.

Tachyon tracked a variety of factors associated with the site, such as who visited, how long they stayed, and which visitors were converted to customers. Tachyon determined that visitors frequently abandoned their carts prior to making a purchase—a clear indicator they were having trouble finding their way on the site. What was needed, Tachyon determined, was enhanced search capabilities that could be tied closely with the IBM solutions already in use.

"[Campmor] had invested a lot in their data catalog, but the embedded search in WebSphere didn't really fully take advantage of it," said Tony Frazier, program director of content discovery marketing at IBM, which was the vendor of choice for the site. The Campmor site uses IBM's DB2, eServer iSeries, and WebSphere Commerce products. "Information about size, color, etc. was there but not well seen," noted Frazier.

The Solution

After some consultation, Tachyon and Campmor decided that enhanced search capabilities would help customers find things more quickly and reduce shopping cart abandonment rates. An exhaustive evaluation of a number of search companies brought Tachyon and Campmor to iPhrase Technologies and its iPhrase Onestep solution, which was renamed WebSphere Content Discovery Server when iPhrase was acquired by IBM in November 2005. According to Marian Lewis, CEO of Tachyon, the search technology was selected for its robustness and its ability to be readily linked to IBM's WebSphere.

To optimize the search technology for the specific needs of an outdoor gear retailer, iPhrase and Tachyon worked

together closely. Tachyon and iPhrase readily split the search-defining job to enable each partner to focus on its particular area of expertise.

WebSphere Content Discovery Server also enables Campmor to cross-sell related items and to customize the shopping experience. A customer searching for tents, for example, can be directed to information on products such as sleeping bags, lanterns, or other camping-related equipment.

The solution also includes apparel ontology, with synonyms and acronyms for various terms a customer might enter, as well as likely misspellings, which are automatically corrected. Because site visitors can conduct detailed searches in less than a second, sales have gone up.

The Results

Customers can find products quickly using the new search-and-discovery tools, which led to a 35 percent increase in online sales in 2005. Today, the Web site generates 70 percent of the company's revenue.

Campmor found that the number of orders placed from searches increased 10 percent and the average size of a

search-based order increased 15 percent within a year. In 2006, 50 percent of the company's orders went directly from the Web to the warehouse without human interaction; only about 3 percent of orders need to be entered by hand.

The initial engagement between Tachyon and iPhrase has blossomed into a variety of new opportunities.

Sources: Compiled from McKeefry (2006) and Tachyon Solutions (2007).

WHAT WE CAN LEARN . . .

This case demonstrates that there is more to building an EC application than choosing a development technique and creating content. Once Campmor initiated its EC application project, it selected an outsourcer to build a custom-made application. Critical to the project was the need to make the site easily navigable by customers so they can easily purchase products. To achieve this, it was necessary for the company—an early adopter of e-commerce—to obtain the continued cooperation from the external developer of the EC application. This case also demonstrates that developing a good EC application often requires an ongoing relationship with a development partner that offers a diverse set of services. All these issues and a few related ones are the subject of this chapter.

19.1 MAJOR E-COMMERCE APPLICATIONS AND THEIR FUNCTIONALITIES

Once it has been determined that a business can benefit from an online presence, the business type, the product line, the business's organization, and the budget dictate what functionality the Web site should have and how the Web site should be developed. Companies can choose from a number of different types of Web sites, including B2C, B2B, exchanges, and the like. Sites of a particular type (e.g., retailer, provider of business services, manufacturer, distributor/wholesaler, media, travel/entertainment) usually use the same underlying applications and provide similar sorts of functionality. Although this simplifies the task of creating the underlying application architecture, the site requirements must still be considered carefully. Before discussing the best approach to developing the site, it is useful to consider the major characteristics, functionalities, and requirements of an EC system. The following discussion focuses on these considerations for some of the more common EC applications.

B2C STOREFRONTS

An electronic storefront must support the same tasks that a physical store supports. In particular, an *electronic storefront* (a seller's Web site where purchases can be made) needs to offer certain capabilities to buyers and to the merchant. These are shown in Exhibit 19.1.

In order to provide these capabilities, an electronic storefront should have the following functions (Nickerson 2002):

- ▶ A *product presentation function* provides the customer with information about the product through the user interface (browser). The information presented can include product advertisements, detailed product specifications, product views, and sample product

EXHIBIT 19.1 Capabilities Needed by Users of Electronic Storefronts

Buyers need the ability to:

- Discover, search for, evaluate, and compare products for purchase using e-catalogs.
- Select products to purchase and negotiate or determine their total price.
- Place an order for desired products using a shopping cart.
- Pay for the ordered products, usually through some form of credit.
- Confirm an order, ensuring that the desired product is available.
- Track orders once they are shipped.

Sellers need the ability to:

- Provide access to a current catalog of product offerings, allowing prospective buyers to analyze and evaluate the offerings.
- Provide an electronic shopping cart in which buyers can assemble their purchases.
- Verify a customer's credit and approve the customer's purchase.
- Process orders (back-end services).
- Arrange for product delivery.
- Track shipments to ensure that they are delivered.
- Provide the means for buyers and visitors to register at the site, to make comments, or to request additional information.
- Answer customers' questions or pass queries and requests to a Web-based call center.
- Analyze purchases in order to customize buyers' experiences.
- Provide Web-based post-sale support.
- Create the capability for cross-sell and up-sell.
- Provide language translation if needed.
- Measure and analyze the traffic at the site to modify and maintain the various applications.

presentations. This function can include additional features such as language selection, product search, and customization for customer preferences.

- ▶ An *order entry function* allows a customer to place an order for selected products. Information about each product ordered is added to the electronic shopping cart, which is a database of orders in process. This function is linked to the enterprise's inventory system in order to check product availability. It also requires access to the enterprise's customer database to update and use customer data.
- ▶ An *electronic payment function* enables the customer to pay for the order and, thus, complete the transaction. Payment options may include credit card, debit card, COD, check (before delivery), and invoice (after delivery). Security is very important in the electronic payment function. The function should provide the necessary security through Secure Socket Layer (SSL), Secure Electronic Transactions (SET), or some other protocol, and customers should be apprised of the security provisions.
- ▶ An *order fulfillment function* provides for the delivery of the product to the customer. The delivery can be digital for products such as music, software, and information. This function is linked to the enterprise's inventory system so that the inventory database can be updated when the order is fulfilled.
- ▶ A *customer service function* provides assistance to customers who have problems or questions related to the purchasing process. Options for providing customer service include FAQs, toll-free telephone numbers, e-mail, and chat rooms.
- ▶ The *product support function* provides assistance to the customer after the product has been received. This support may include initial setup and installation, regular operation, troubleshooting, return policy, ongoing maintenance, and warranty or nonwarranty repair or replacement.

For a comparison of various software packages that support the creation of a B2C storefront, see ecommerce-guide.com/solutions. For more on storefront construction, tools, and vendors, see Chapter 16 and Section 19.5.

SUPPLIER SELL-SIDE B2B SITES

A sell-side B2B site is similar to a B2C storefront, enabling one business to purchase goods and services from another. However, a B2B site also has additional features (see Chapter 5), including:

- ▶ Personalized catalogs and Web pages for all major buyers
- ▶ A B2B payment gate
- ▶ Electronic contract negotiation features
- ▶ Product configuration by customers (e.g., Cisco or Dell)
- ▶ Affiliate program capabilities
- ▶ Business alerts (e.g., to special sales, to news)

E-PROCUREMENT

An e-procurement site is an online intermediary that offers businesses access to hundreds of parts and services provided by suppliers. E-procurement systems come in several variations, each with its own specialized capabilities. See more discussion about e-procurement in Chapter 5.

Aggregating Catalogs

In large organizations, multiple buyers are involved in making purchases from a large number of suppliers. One way to reduce costs and other inefficiencies in the purchase process is to aggregate the items from approved suppliers into a single online catalog (see Chapter 5). Some of the specialized requirements for this type of site include:

- ▶ Search engine for locating items with particular characteristics
- ▶ Comparison engine for alternative vendors
- ▶ Ordering mechanism
- ▶ Budget and authorization feature

- ▶ Usage comparison (among various departments)
- ▶ Payment mechanism (e.g., use of a purchasing card)

Aggregated catalogs are used by many large companies and government agencies.

Reverse Auctions and Tendering Systems

In a reverse auction, buyers list the items they wish to purchase, and sellers bid to provide those items at the lowest price (see Chapter 5). Sites of this sort provide the following capabilities:

- ▶ Catalog of items to be tendered and their content management
- ▶ Search engine (if the site has many items)
- ▶ Personalized pages for potential large bidders
- ▶ Reverse auction mechanisms, sometimes in real time
- ▶ Facility to help prepare, issue, manage, and respond to a buyer's request for quotes (RFQs)
- ▶ Ability to bid dynamically
- ▶ Automatic vendor approval and workflow (e.g., SmartMatch's supplier identification technology)
- ▶ Electronic collaboration with trading partners
- ▶ Standardization of RFQ writing
- ▶ Site map
- ▶ Mechanism for selecting suppliers
- ▶ Automatic matching of suppliers with RFQs
- ▶ Automatic business process workflow
- ▶ Ability for bidders to use m-commerce for bidding
- ▶ Automated language translation

Forward Auctions

Forward auctions enable selling companies to post items they want to sell, and buying companies to compete for the best prices acceptable to the selling companies for those items. In forward auctions, winning bidders (buyers) are obligated to buy items (see Chapters 5 and 10 and docs.sun.com/source/816-5981-10/auctions/auc_deffwrdaucs.htm). A typical forward auction consists of the following steps:

1. Both the seller and the buyer complete the online registration process, including providing shipping points and regulatory and banking information.
2. The seller starts an auction by listing the product, the asking price, and the quantity on a form.
3. The buyer chooses a bid product and indicates a bid price and quantity. The buyer may also set the maximum price and bid increments.
4. Sophisticated software determines the auction winner, based on price, volume, and timing.
5. A funds transfer from the bidder takes place immediately.
6. Upon successful funds transfer, a freight company is dispatched to the seller's location; the product is loaded and delivered to buyer's location.
7. The buyer inspects and accepts the product.
8. Funds are released to the seller.

The capabilities of forward auctions can be best viewed at ebay.com.

Exchanges

An exchange is an e-marketplace that connects many buyers with many suppliers (see Chapter 6). In addition to combining the functionalities of buy-side, e-procurement, and auction sites, they also have a number of other capabilities:

- ▶ Collaboration services (including multichannel services)
- ▶ Community services

- ▶ Web-automated workflow
- ▶ Integrated business process solutions
- ▶ Central coordination of global logistics for members, including warehousing and shipping services
- ▶ Integration services (systems/process integration into e-marketplace, trading partners, and service providers)
- ▶ Data mining, customized analysis and reporting, real-time transactions, trend and customer behavior tracking
- ▶ Transaction-flow managers
- ▶ Negotiation mechanisms
- ▶ Language translation
- ▶ Comprehensive links to related resources

These lists of major characteristics and functionalities can be used by application developers as outlines or checklists from which to develop plans for specific EC applications. For a listing of software packages that support the various capabilities of B2B sites, see *Business.com* (2007).

Portals

A portal is a single Web interface that provides personalized access to information, applications, business processes, and much more. With portal technology, an organization can lower development and deployment costs and significantly increase productivity. Using a portal, information can be aggregated and integrated within a particular working environment, application, or service, or a single interface can be used to target an individual user's needs and interests. Portals help to harmonize content, commerce, and collaboration with business goals. A list of different types of portals and their capabilities follows:

- ▶ *Line-of-business portals* provide easy access to applications that serve a specific area, such as procurement or human resources.
- ▶ A *corporate intranet portal* often acts as a gateway to other portals and Web sites operated by an organization.
- ▶ *Extranet portals* act as an interface between companies, customers, and suppliers, revealing subsets of information to specific audiences.
- ▶ *Customer service and self-service portals* are often seen as subsets of a corporate extranet.
- ▶ *Team or divisional portals* are used by groups or communities that want to share specific content or business functions.
- ▶ A *personal portal* is geared to assist individuals who access information and resources.
- ▶ An *enterprise portal* is the central portal for an entire organization. It comprises all other portals deployed.

See *Microsoft.com* (2007) for more features and capabilities of portals.

Other EC Systems

Several EC systems exist, each with its own set of required capabilities. For example, e-learning sites may have a student part and an instructor part, each with its own set of requirements. Collaboration EC sites require several collaboration capabilities and tools.

Section 19.1 ▶ REVIEW QUESTIONS

1. Examine 15 different Web sites and choose your 5 favorites. Are these the kinds of Web sites that you would choose to visit or use as a template if you had a business? Explain why or why not.
2. List the major functions of an electronic storefront.
3. Describe some of the major functions required by an aggregating catalog.
4. Describe some of the major functions needed to build a reverse auction.
5. Describe the basic steps of a forward auction.

6. List some of the functional requirements of an online exchange.
7. Name the different types of portals and describe their basic features.

19.2 A FIVE-STEP APPROACH TO DEVELOPING AN E-COMMERCE LANDSCAPE

A well-developed Web site not only adds to the value of the product or service being offered; it also enhances the worth of the company. Therefore, it is important that a firm choose the correct development strategy in order to obtain the greatest return on its investment. The diversity of e-business models and applications, which vary in size from small stores to global exchanges, requires a variety of development methodologies and approaches.

For example, small storefronts with a few key components can be developed with HTML, Java, or another programming language. They also can be implemented with commercial packages, leased from an *application service provider (ASP)*, or purchased from a *site builder*. *Larger or special EC applications* can be developed in-house or outsourced (see the opening case). Building medium to large applications requires extensive integration with existing information systems, such as corporate databases, intranets, enterprise resource planning (ERP), and other application programs. Therefore, although the process of building EC systems can vary, in many cases, it tends to follow a fairly standard format.

The traditional systems development life cycle (SDLC) systematically leads developers through six analysis and design stages: problem identification, analysis, logical design, physical design, implementation, and maintenance. The SDLC is the basis for development of the majority of *traditional business systems* (see Whitten and Bentley 2007 for more details on this approach). However, innovative new software and hardware are enabling a move to a more streamlined approach to *e-commerce development*, as discussed in Case 19.1.

Exhibit 19.2 (p. 9) shows the five major steps needed to develop a typical e-commerce application.

STEP 1: IDENTIFYING, JUSTIFYING, AND PLANNING EC SYSTEMS

EC applications, like all other information systems, are usually built to enable one or more business processes. Consequently, their planning must be aligned with that of the organization's overall business plan and the specific processes involved. Always remember that existing processes may need to be restructured to take full advantage of the benefits of the supporting IT. Furthermore, each application must be carefully analyzed, using different methods, such as the methodology discussed in the opening case, to ensure that it will have the needed functionality to meet the requirements of the business processes and the users and that its benefits will justify its cost (see Chapter 5). Both of these activities may be complex, but they are necessary, especially for systems that require high investment to acquire, operate, and maintain. The output of this step is a decision to go with a specific application, with a timetable, budget, and assigned responsibility. This first step is typically performed in-house (with consultants if needed). All other steps can be completed either in-house or outsourced.

STEP 2: CREATING AN EC ARCHITECTURE

An **EC architecture** is a plan for organizing the underlying infrastructure and applications of a site. The plan specifies the following:

- ▶ Information and data required to fulfill the business goals and vision
- ▶ Application modules that will deliver and manage the information and data
- ▶ Specific hardware and software on which the application modules will run
- ▶ Necessary security, scalability, and reliability required by the applications
- ▶ Human resources and procedures for implementing the architecture

Various IT tools and methodologies can be used to support the creation of an application architecture (e.g., see Kendall and Kendall 2005). Because the creation of an architecture is an iterative process, collaborative methodologies, such as joint application development (JAD), are especially useful in identifying and modifying system requirements.

EC architecture

A plan for organizing the underlying infrastructure and applications of a site.

CASE 19.1

EC Application

TD BANKNORTH—ONLINE CUSTOMER SERVICE RIGHTNOW

As a community bank that uses service as a selling point, TD Banknorth Inc., a leading banking and financial services company headquartered in Portland, Maine, and a majority-owned subsidiary of TD Bank Financial Group with banking divisions in eight Northeastern states, was not satisfied with a 90 percent response rate within 24 hours to its customer e-mails. So, it decided to overhaul its online customer service system to improve its record.

With the help of RightNow's on-demand solutions and professional services, TD Banknorth increased its response rate to an impressive 97 percent for nonconfidential e-mails within 24 hours and more than 50 percent within just .6 hours after receipt. The company also found that making information easier to find on its Web site lowered e-mail volume by 55 percent, to about 35 messages a day. Mark Ellis, Senior Vice President of eCommerce at TD Banknorth, credits the decline to the new customer resource center, which provides Web site visitors information more efficiently by integrating e-mail and Internet capabilities. This lower volume has freed service representatives to handle other duties, such as answering phones and handling secure e-mails.

In 2004, when the company, then called Banknorth Group Inc., set out to upgrade its online customer service, its aim was to provide information fast enough that customers would not have to send e-mails or make phone calls. The company saw an opportunity to improve e-mail service levels and provide customers with easily accessible answers to their banking questions. TD Banknorth also sought to improve the efficiency of its service operations across the board, maintain cost control, and accomplish all this while its technology resources supported an aggressive acquisition strategy. To optimize the online self-service facilities it planned to provide to its customers, TD Banknorth decided on a true knowledge base system that recognizes natural language rather than the hodgepodge of quotation marks and connectors that Web portals such as Google and Yahoo! expect. For example, it wanted its customers to be able to simply type, "How can I reset my online banking password?" or enter a few keywords from that phrase so that the answer appears courtesy of the FAQ mechanism. If TD Banknorth customers failed to find satisfactory answers, they could alternatively e-mail the bank. They also wanted the knowledge-base tool to allow contact-center representatives to tweak the technology. For example, drawing on their interactions with customers, the reps could propose question-and-answer pairs for the knowledge base.

After an in-depth search process for the best solution provider, TD Banknorth whittled the list of candidates down to three finalists. At this point TD Banknorth began looking more closely at specific features and functionality and applying a weighted scorecard to appropriately compare the respective solutions. TD Banknorth evaluated everything from incident management tools and knowledge base search functions to ease of customization and management. At the end of the day, RightNow was the clear choice. It turned out to be the right decision, and TD Banknorth succeeded in addressing its customer service concerns on all counts with the help of RightNow's on demand solutions and professional services.

Part way through the project, the bank consolidated its six regional holdings into a single brand, which required a parallel consolidation of the corporate Web site. This pushed the target implementation date out but created another opportunity. A soft launch was introduced into the plan, which allowed TD Banknorth to comprehensively test and refine the knowledge base content prior to the go-live date. Despite the scope and complexity of the system TD Banknorth required, RightNow was able to go live in just 90 days. In 2006, when TD Banknorth bought Hudson United Bancorp of Mahwah, New Jersey, for \$1.9 billion. Hudson United's customers also started using the new site. Having RightNow host the knowledge base allowed for a quick 90-day rollout.

Although responsiveness generally is not a make-or-break issue, unlike onerous fees or failure to post a deposit, sometimes a bad experience—for example, an unanswered e-mail—can cost a bank a customer. "RightNow has improved our response times and enabled us to support the bank's growth without driving up our overhead," declares Ellis. "Our return-on-investment is definitely going to be well beyond our initial projections."

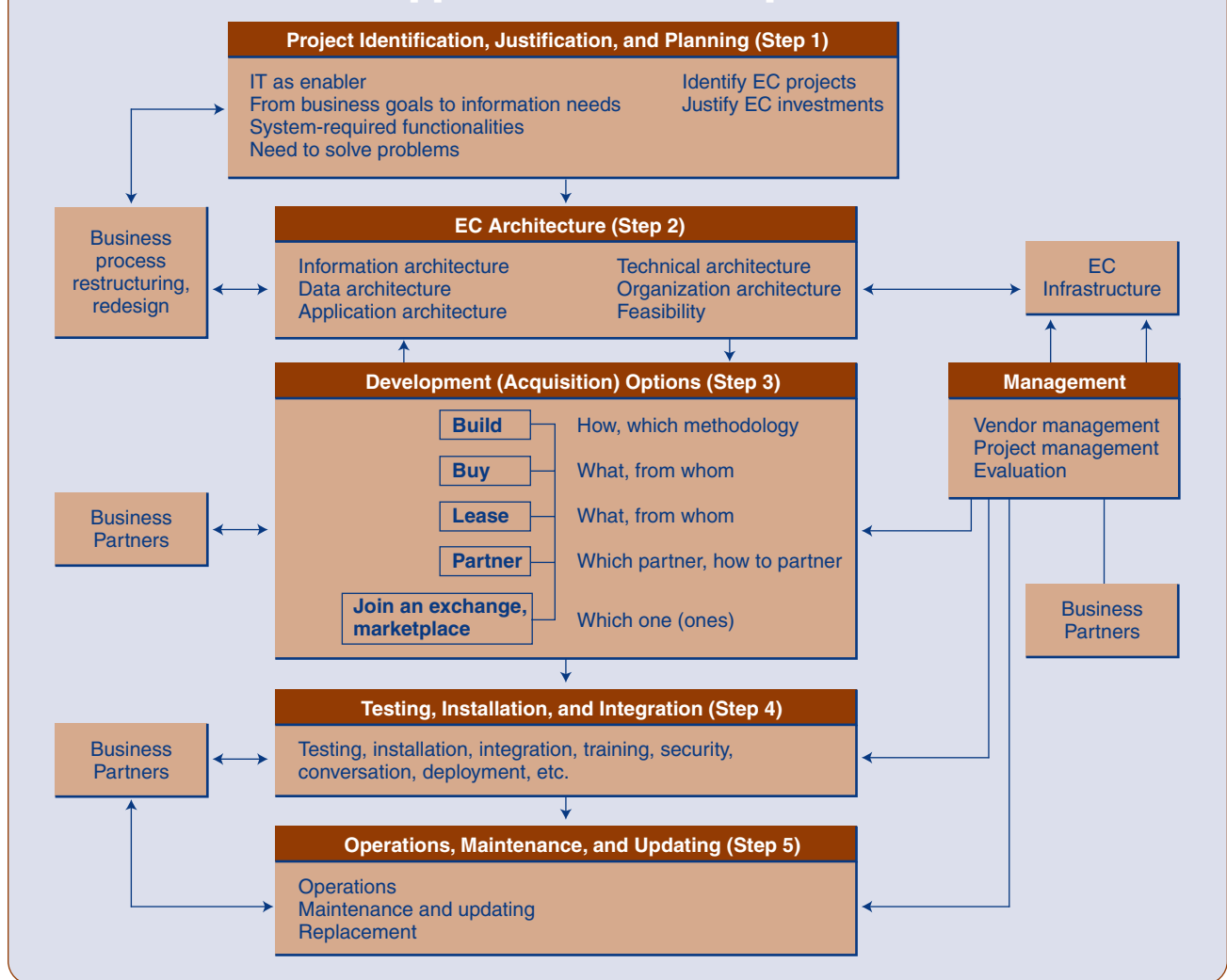
Sources: Compiled from Duvall (2006) and RightNow Technologies Inc. (2007).

Questions

1. Do you agree that RightNow was the best choice for TD Banknorth? Why or Why not?
2. Why is the bank's customer service record so important to TD Banknorth?
3. What roles does Web Services play at TD Banknorth?

The results obtained from step 2 are routed to the strategic planning level (e.g., to a steering committee). Based on the results of step 2, the application portfolio may be changed. For example, the steering committee may discourage or scale down the specific project because it is too risky. Once the architecture is determined and the project gets final approval, a decision about *how* to develop the specific EC application must be made and a development option chosen.

EXHIBIT 19.2 The EC Applications Development Process



STEP 3: SELECTING A DEVELOPMENT OPTION

EC applications can be developed through several alternative approaches that will be discussed in detail in Section 19.3. The major options are:

- Build the system in-house.
- Have a vendor build a customized system.
- Buy an existing application and install it, with or without modifications, by yourself or through a vendor.
- Lease standard software from an application service provider (ASP), lease as a service (SaaS), or lease via utility computing.
- Enter into a partnership or alliance that will enable the company to use someone else's application.
- Join a third-party e-marketplace, such as an auction site, a bidding (reverse auction) site, or an exchange, that provides needed capabilities to participants (e.g., Yahoo! Store).
- Use a combination of approaches.

The criteria for selecting from among the various options are presented in Section 19.4. Once an option is chosen, the system can be developed. At the end of this step, an application is ready to be installed and made available. No matter what option is chosen, there is a strong possibility that the firm will work with vendor(s) and/or software provider(s). In this case, the firm will need to manage its vendor relationships (see Section 19.8).

STEP 4: INSTALLING, TESTING, INTEGRATION, AND DEPLOYING EC APPLICATIONS

Once a system has been developed, the next step involves getting the application up and running in the selected hardware and network environment. One of the steps in installing an application is connecting it to back-end databases, to other applications, and often to other Web sites. For example, if a prospective customer orders a product from a site, it would be helpful if the site could determine if the product is in stock. To do this, the ordering system would need to be connected to the inventory system. Details of the connection process are supplied in Section 19.6. This step can be done in-house or outsourced.

At this point, the modules that have been installed need to be tested. Sommerville (2004) recommends a series of different tests:

unit testing

Testing application software modules one at a time.

integration testing

Testing the combination of application modules acting in concert.

usability testing

Testing the quality of the user's experience when interacting with a Web site.

acceptance testing

Determining whether a Web site meets the original business objectives and vision.

- ▶ **Unit testing.** Test each module one at a time.
- ▶ **Integration testing.** Test the combination of modules acting in concert.
- ▶ **Usability testing.** Test the quality of the user's experience when interacting with the site.
- ▶ **Acceptance testing.** Determine whether the site meets the firm's original business objectives and vision.

Once all the Web site applications pass all of the tests, they can be made available to the end users. At this stage, issues such as conversion strategies, training, and resistance to change may need to be addressed.

STEP 5: OPERATIONS, MAINTENANCE, AND UPDATING

It usually takes as much time, effort, and money to operate and maintain a site as it does to build and install it in the first place. To enjoy continual usage, a site needs to be updated continually. For example, at a B2C site new products need to be added to the catalog, prices need to be changed, and new promotions need to be run. These changes and updates need to undergo the same testing procedures used during the installation process. Additionally, usage patterns and performance need to be studied to determine which parts of the underlying applications should be modified or eliminated from the site. See Reynolds (2004) for more about the operation and maintenance of an EC site.

MANAGING THE DEVELOPMENT PROCESS

The development process can be fairly complex and must be managed properly (Xia and Lee 2004). For medium-to-large applications, a project team is usually created to manage the process and the vendors. Collaboration with business partners also is critical. As shown in various chapters of this book, some e-business failures are the result of a lack of cooperation by business partners. For example, a firm can install a superb e-procurement system, but if their vendors will not use it properly the system will collapse. Projects can be managed with project management software (see examples of various project management software at office.microsoft.com/project and primavera.com). Best practice management also includes periodic evaluations of system performance. Standard project management techniques and tools are useful for this task. For a review of project management techniques, see Schwalbe (2006). Finally, do not rule out the possibility that implementing an EC project may require restructuring one or more business processes. See Kanter and Walsh (2004) for further discussion of this topic.

Section 19.2 ▶ REVIEW QUESTIONS

1. Go to the Web site of the developers of each of your five favorite Web sites (chosen in answer to Section 19.1, Review Question #1). What expertise do they profess to have? What projects have they completed? Would you feel comfortable hiring their services?
2. List the major steps in developing an EC application.
3. Define the various types of testing used during the EC development process.

19.3 DEVELOPMENT OPTIONS FOR E-COMMERCE APPLICATIONS

If the desired Web site is relatively simple, a firm may decide to build the Web site itself. However, the firm must ask a few questions: Is the firm capable of developing the site? Does the firm have access to the proper tools to create the pages? If the firm does not have these capabilities, it is usually best to turn over the task to a professional developer. The ideal developer is one who can design a site with the correct look and feel, who has an in-depth knowledge of search engine optimization, and who is able to correctly handle any complex coding that may be required. Resources on building a Web site are available at Internet Marketing Singapore (internetmarketingsingapore.com), Sell IT! (sellitontheweb.com/ezone/help.shtml), and the Microsoft Small Business Center (microsoft.com/smallbusiness/resources/technology/ecommerce/5_common_e_commerce_site_mistakes.aspx). A useful site for finding an experienced Web site designer is Supplier-Match.com (supplier-match.com/categories/e-commerce_services.jsp).

Regardless of the complexity of the site, three basic options for developing an EC Web site are available: (1) develop the site in-house, either from scratch or with off-the-shelf components; (2) buy a packaged application designed for a particular type of EC site; or (3) lease the application from a third party. Each of these approaches has its benefits and limitations.

IN-HOUSE DEVELOPMENT: INSOURCING

The first generation of EC development was accomplished largely through proprietary programming and in-house development (see Zhao et al. 1998). Using this approach, the Internet browser serves as the development platform. The programmers write EC systems using a combination of HTML and script languages such as HTX, CGI, IDC, and JavaScript. Databases developed on top of a database management system (DBMS) usually serve as the information repository to store EC data. Although this first generation of EC development has built up valuable experience and achieved industrial momentum, the lack of **reusability** (i.e., the likelihood a segment of source code can be used again to add new functionalities with slight or no modification) in current EC applications and the lack of **interoperability** (i.e., the ability to connect people, data and diverse systems, standards) created a great barrier to widespread application of EC.

Although in-house development—**insourcing**—can be time consuming and costly, it may lead to EC applications that better fit an organization's strategy and vision and differentiate it from the competition. Companies that have the resources to develop their e-business application in-house may follow this approach in order to differentiate themselves from the competition, which may be using standard applications that can be bought or leased. The in-house development of EC applications, however, is a challenging task, because most applications are novel, have users from outside the organization, and involve multiple organizations. Shurville and Williams (2005) demonstrate how a combination of hard and soft project and change management methodologies guided successful in-house development of a campuswide information system.

Development Options

Developers have three major options for developing an application:

- ▶ **Build from scratch.** This option is used rarely. It should be considered only for specialized applications for which components are not available. It is expensive and slow, but it may provide the best fit.
- ▶ **Build from components.** The required applications are often constructed from standard components (e.g., Web servers such as Apache or Microsoft's IIS) using Web scripting languages, such as PHP, Microsoft's Active Server Pages (ASP), JavaServer Pages (JSP), or ColdFusion. These scripting languages make it easier to integrate application functionality with back-end databases and other back-office systems (e.g., order entry). For a methodology of evaluating component-based systems, see Dahanayake et al. (2003).

reusability

The likelihood a segment of source code can be used again to add new functionalities with slight or no modification.

interoperability

Connecting people, data, and diverse systems. The term can be defined in a technical way or in a broad way, taking into account social, political, and organizational factors.

insourcing

In-house development of applications.

- Enterprise application integration.** The enterprise application integration (EAI) option is similar to the build from components option, but instead of using components, an entire application is employed. This is an especially attractive option when applications from several business partners need to be integrated.

Insourcing is a challenging task that requires specialized IT resources. For this reason, most organizations usually rely on packaged applications or completely outsource the development and maintenance of their EC sites.

BUY THE APPLICATIONS

A number of commercial packages provide standard features required by EC applications. These packages are ready to turn on and operate. This option is also known as a **turnkey approach**; the package is ready to use without further assembly or testing.

The turnkey approach involves buying a commercial package, installing it as is, and starting it up. Buying a commercial package requires much less time and money than in-house development. When selecting a particular package, the package should not only satisfy current needs, it must also be flexible enough to handle future ones; otherwise the package may quickly become obsolete. Additionally, because one package can rarely meet all of an organization's requirements, it is sometimes necessary to acquire multiple packages. In this case, the packages need to be integrated with each other and with other software and data.

This option has several major advantages:

- Many different types of off-the-shelf software packages are available.
- It saves time and money (compared to in-house development).
- The company need not hire programmers specifically dedicated to an EC project.
- The company knows what it is getting before it invests in the product.
- The company is neither the first nor the only user.
- The price is usually much lower than the in-house option.
- The vendor updates the software frequently.

This option also has some major disadvantages:

- Software may not exactly meet the company's needs.
- Software may be difficult or impossible to modify, or it may require huge process changes.
- The company may experience loss of control over improvements and new versions.
- Off-the-shelf applications can be difficult to integrate with existing systems.
- Vendors may drop a product or go out of business.

See softwaresearch.us/search.aspx?keywords=E+commerce+turnkey for a directory of vendors of EC turnkey systems. The buy option is especially attractive if the software vendor allows for modifications. However, the option may not be as attractive in cases of high obsolescence rates or high software cost. In such cases, leasing may be a more appealing option.

OUTSOURCING/LEASING EC APPLICATIONS

The use of outside contractors or external organizations (often software vendors) to acquire EC applications is called **outsourcing**. It is a method of transferring the management and/or day-to-day execution of an entire business function to a third-party service provider. Outsourcing is a valuable option that more and more companies are using. In many cases, systems need to be built quickly, and the special expertise of outside contractors and software vendors is necessary.

Large companies may choose outsourcing when they want to experiment with new EC technologies without a great deal of up-front investment. Outsourcing also allows large firms to protect their internal networks and to gain expert advice. Small firms with limited IT expertise and tight budgets also find outsourcing advantageous.

Outsourcers can perform any or all tasks in EC applications development. For example, they can plan, program, and build applications and integrate, operate, and maintain them. It is useful for firms to develop good relationships with outsourcers (see Kishore et al. 2003).

turnkey approach

Ready to use without further assembly or testing; supplied in a state that is ready to turn on and operate.

outsourcing

A method of transferring the management and/or day-to-day execution of an entire business function to a third-party service provider.

Several types of vendors offer services for creating and operating EC applications:

- ▶ **Software houses.** Many software companies, from IBM to Oracle, offer a range of outsourcing services for developing, operating, and maintaining EC applications.
- ▶ **Outsourcers and others.** IT outsourcers, such as EDS, offer a variety of services. Also, the large CPA companies and management consultants (e.g., Accenture) offer some outsourcing services.
- ▶ **Telecommunications companies.** Increasingly, the large telecommunications companies are expanding their hosting services to include the full range of IT and EC solutions. MCI, for example, offers Web Commerce services for a monthly fee.

Although the trend to outsource is rising, so is the trend to conduct outsourcing offshore—mainly in India and China. This approach is not without risks. For example, although outsourcing offshore may lead to substantial dollar savings, offshore labor skills may be inferior to those found onshore, and the resultant quality of the Web site development may be unacceptable. For more discussion about the drivers, effects, and risks of outsourcing, see Dutta and Roy (2005), Aron et al. (2005), and Overby (2003).

To accommodate the increasingly popular move to EC development by vendors, a less risky outsourcing option known as *software on demand* has emerged as a development method. Initially, this leasing option was provided through *utility computing*, then through *ASPs*, and more recently, through firms that provide *software as a service* (SaaS). Large and small firms alike will often choose the lease option to experiment with new software before making a large up-front investment or as an ongoing method of acquiring and supporting EC software. Another benefit is that it enables firms to protect their internal networks. Through leasing, new entrants into e-business (usually smaller firms) are able to establish a market presence in a much shorter period of time. Each of these alternative sources of software acquisition is discussed in the following sections

UTILITY COMPUTING

Utility computing is a business model whereby computer resources are provided on an on-demand and pay-per-use basis. This contrasts sharply with the traditional SDLC model of purchasing physical systems, configuring them, and devoting them to one application for their useful life. With utility computing, customers do not own the expensive computer resources but are billed only for their actual use of the resources. Because the utility computing provider can spread customers' variance in resource needs, resource utilization can be optimized. Because the utility computing service is based on usage, computing resources are metered and the user charged on that basis. This is comparable to the use of electricity, gas, and most other utilities; hence the name *utility computing*. Utility computing is sometimes also called *on demand computing*.

Utility computing has a long history. It was first described by John McCarthy in 1961 at the MIT Centennial (en.wikipedia.org/wiki/Utility_computing):

If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility. . . . The computer utility could become the basis of a new and important industry.

HP was the original leader of the *utility computing* concept, recognizing that computing power and resources can be delivered as a service, flowing on demand as and where needed. Subsequently, IBM offered this type of flexible delivery of computing power and database storage to big banks from its worldwide data centers. In 2000, Sun offered utility computing to consumers through its Sun Grid service. HP introduced the Utility Data Center in 2001. Since 2000, many companies have entered the utility computing market. Some of these organizations use utility computing to help offset hardware costs, others use it to share the cost of resources within organizations. In December 2005, Alexa launched Alexa Web Search Platform, a Web search building tool, for which the underlying power is utility computing; Alexa charges users for storage, utilization, and so on. SoftLayer Technologies is continuing to develop utility services aimed at meeting the needs of the emerging Web 2.0 market. Lance Crosby, President and Chief Executive Officer at SoftLayer said, "As a company, we recognize the importance of IT scalability. With the needs of businesses changing as rapidly

utility (on-demand) computing

Unlimited computing power and storage capacity that can be used and reallocated for any application—and billed on a pay-per-use basis.

as the technology that powers them, we understand that the ability to expand on demand is paramount” (Utilitycomputing.com 2006).

As shown in Exhibit 19.3, the utility-computing value proposition consists of three layers of tools and two types of value-added services. Each tool must be seamlessly integrated to create a comprehensive solution but will usually be implemented separately. These three tools are:

policy-based service-level-management tools

Coordinate, monitor, and report on the ways in which multiple infrastructure components come together to deliver a business service.

policy-based resource-management tools

Automate and standardize all types of IT management best practices, from initial configuration to ongoing fault management and asset tracking.

application service provider (ASP)

A company that provides business applications to users for a small monthly fee.

- ▶ **Policy-based service-level-management tools** coordinate, monitor, and report on the ways in which multiple infrastructure components come together to deliver a business service.
- ▶ **Policy-based resource-management tools** coordinate, monitor, and report on the ways in which multiple infrastructure components come together to deliver a business service. They automate and standardize all types of IT management best practices, from initial configuration to ongoing fault management and asset tracking.
- ▶ **Virtualization tools** enable server, storage, and network resources to be deployed and managed as giant pools and seamlessly changed as needs change.

These tools share multisourcing delivery and financing services (left side of Exhibit 19.3) and provide for customer access and management services (right side of Exhibit 19.3).

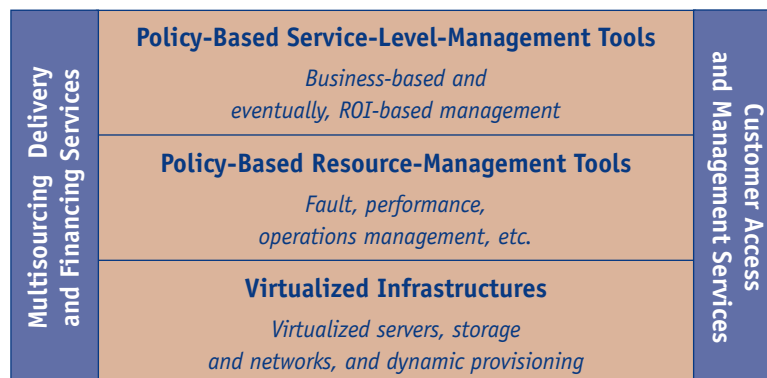
APPLICATION SERVICE PROVIDERS (ASP)

An **application service provider (ASP)** manages application servers in a centrally controlled location rather than on a customer’s site. Applications are then accessed via the Internet or VANs through a standard Web browser interface. Such an arrangement provides a full range of services for the company using the ASP: Applications can be scaled, upgrades and maintenance can be centralized, physical security over the applications and servers can be guaranteed; and the necessary critical mass of human resources can be efficiently utilized. The determinants of ASP adoption as an innovation are discussed by Daylami et al. (2005).

The end user businesses pay a licensing fee. Monthly fees are separate and are paid to the maker of the software and to the ASP “host” of the software. In general, these fees include payment for the application software, hardware, service and support, maintenance, and upgrades. The fee can be fixed or may be based on utilization.

Leasing from an ASP is a particularly desirable option for SMEs, for which in-house development and operation of IT applications can be time consuming and expensive. Leasing from ASPs saves various expenses (e.g., labor costs) in the initial development stage. It also helps reduce software maintenance, upgrading, and user training costs in the long run. A company can select other software products from the same ASP to meet its changing needs and does not have to invest further in upgrading the existing one. Thus, overall business competitiveness can be strengthened through reducing time-to-market and enhancing the firm’s ability to adapt to changing market conditions. ASPs are particularly effective for IT applications for which timing, flexibility, and agility are crucial.

EXHIBIT 19.3 The Five Elements of a Successful Utility-Computing Value Proposition



Source: Kucharvy, T. *The Five Rules for Jump-Starting the Utility-Computing Market*. Boston, MA: Summit Strategies, Inc., January 2003.

Leasing from ASPs does have its disadvantages. ASP applications are typically hosted by third parties, who ordinarily do not have application expertise but merely manage the servers; they serve up traditional client-server applications with front-ends added as an afterthought. ASPs generally do not build the application themselves but instead take an off-the-shelf application (such as a messaging platform, an enterprise requirements planning tool, or a salesforce automation package) and run it for customers. For example, an ASP might make the latest version of Microsoft Office available across the Web to customers who pay a fee per month for access to the software

In addition, many companies are concerned with the adequacy of protection offered by ASPs against hackers, theft of confidential information, and virus attacks. Also, leased software often does not provide the perfect fit for the desired application. It is also important to ensure that the speed of the Internet connection is compatible with that of the application to avoid distortions in its performance. For example, it is not advisable to run heavy-duty applications on a modem link below a T1 line or high-speed DSL.

From the ASP vendor's point of view, the benefits presented by the ASP model are many. For one, in the long-distance carrier and Internet service providers (ISP) markets, revenues are squeezed due to heavy competition. These companies are looking to generate revenues from sources other than connectivity and transport, and ASP services offer a new outlet. The ASP Industry Consortium, has emerged to support the ASP concept; its founding members include AT&T, Cisco, Citrix Systems, Ernst & Young, Verizon, IBM, Marimba, Sharp Electronic, Sun Microsystems, UUNET, and Verio.

A detailed list of the benefits and risks associated with ASPs is provided in Exhibit 19.4. Information about the general state of the ASP marketplace can be obtained from the Computing Technology Industry Association's (CompTIA) Software Services (comp.tia.org/certification/a/default.aspx). Major ASPs for enterprise EC systems include SAP, Oracle, and IBM. A comprehensive list of ASPs for EC can be found at the DMOZ Open Directory Project (softwaresearch.us/search.aspx?keywords=E+commerce+turnkey).

Because most applications typically provided by ASPs are not written as Internet-native applications, performance can be poor and application updates are often no better than self-managed applications. Consequently, a need to supplement ASP offerings emerged. The latest innovation in on-demand computing is *software as a service (SaaS)*.

SOFTWARE AS A SERVICE (SAAS)

Software as a Service (SaaS) is a new model of software delivery whereby the software is specifically designed for delivery in an online environment. It is essentially "leased" from a software company that provides maintenance, daily technical operation, and support for the software provided to the client. "Leading the charge" into this software delivery paradigm

software as a service (SaaS)

A model of software delivery where the software company provides maintenance, daily technical operation, and support for the software provided to their client. SaaS is a model of software delivery rather than a market segment.

EXHIBIT 19.4 Benefits and Risks of Using an ASP

Type	Benefits	Potential Risks
Business	<ul style="list-style-type: none"> Reduces the need to attract and retain skilled IT professionals Enables companies to concentrate on lack of strategic use of IT Enables SMEs to use tier-1 applications (e.g., ERP, SCM, and CRM) Application scalability enables rapid growth of companies 	<ul style="list-style-type: none"> Loss of control and high level of dependence on ASP Inability of ASP to deliver quality of service; skills and experience
Technical	<ul style="list-style-type: none"> Fast and easy application deployment Higher degree of application standardization Access to wide range of applications Application maintenance simplified and performed by ASP Simplified user support and training 	<ul style="list-style-type: none"> Level of customization and legacy application integration offered by ASP is insufficient Low reliability and speed of delivery due to bandwidth limitations Low capability of ASP to deal with security and confidentiality issues
Economic	<ul style="list-style-type: none"> Low total cost of ownership Low up-front investments in hardware and software Improved cost control as result of predictable subscription costs 	<ul style="list-style-type: none"> Pricing changes by ASP unpredictable for application updates and services

Source: Kern, T. and J. Kreijger. "An Exploration of the ASP Outsourcing Option" Proceedings, 34th HICSS, Hawaii, January 2001. ©2001 IEEE.

EXHIBIT 19.5 List of SaaS Vendors and Applications

- Business Objects (Crystal Reports) (business intelligence)
- CollabNet (collaboration)
- Google (Google Maps, Google Docs, Google Spreadsheets, JotSpot)
- NetSuite (for small firms)
- Oracle (PeopleSoft on Demand; E-Business Suite on Demand)
- Replicon (Timesheets)
- RightNow Technologies (CRM and Customer Experience)
- Salesforce.com (CRM)
- SPS Commerce (EDI for retailers, distributors, manufacturers)
- WebEx (Web conferencing, video conferencing)
- YouSendIt (Web-based file hosting; large e-mail attachments)

shift is Google (Beer 2006). Exhibit 19.5 lists software developers who are now offering some of their products as SaaS and the diverse applications they address.

To engage in SaaS, the user pays the software provider for the user's actual usage or a fixed monthly/quarterly/annual fee instead of the one-time large fee plus support, as with the traditional software licensing model. SaaS is particularly advantageous when a company wants to experiment with a package before making a heavy up-front investment. This solution also enables a firm to protect its internal networks and quickly establish a presence in the market. With SaaS, software can be delivered to any market segment, including home office users, small businesses, medium and large businesses. It can result in substantial cost and time savings. Although SaaS is useful to any size company, it is particularly appealing for SMEs, who often have limited IT resources and tight budgets.

SaaS allows organizations to access business functionality from a central location over the Web at a cost typically less than that for licensed applications because SaaS pricing is based on a monthly fee. Also, because the software is hosted remotely, users do not need to invest in additional hardware or software support. SaaS removes the need for organizations to handle installation, set up, and daily upkeep and maintenance. SaaS may also be referred to as *hosted applications*.

An important factor in the shift to SaaS is the need to integrate or combine software components. According to some (Jakovljevic 2006), SaaS at its most sophisticated must be delivered using a *service-oriented architecture (SOA)* with *Web Services*.

RISE OF WEB SERVICES AND SERVICE-ORIENTED ARCHITECTURE

Interrelated technologies that can greatly facilitate the development of complex EC applications are Web Services and service-oriented architecture (SOA).

Except in the simplest of cases, EC sites require the integration of software applications written in different programming languages and residing on different computer hardware distributed across the Internet. For example, on many B2C sites order entry is handled by one software application or module, payment authorization by another application or module, and shipping by yet another application or module. In these cases, there is a good chance that the order entry, payment authorization, and shipping software modules all reside on separate application servers linked through a Web server. Even when packaged applications are used, a substantial amount of the implementation effort revolves around the task of tying together these disparate applications or modules in such a way that the underlying connections are transparent to the end users.

Existing technologies make integration a difficult task for a number of reasons (Tabor 2002; Erl 2004):

- **Platform-specific objects.** Existing EC software applications consist of a series of software *objects*. Software objects have *properties* (attributes) and *methods* (actions that can be performed on or by the object). For example, an order-entry application might have an "order" object that has a property specifying the "quantity" being ordered and a method called "set" that allows the quantity to be updated. In a distributed application, such as an

EC storefront, the application objects residing on different computers must have a way to communicate with one another across the network.

Two main technologies are available for accomplishing this task: Microsoft's Distributed Component Object Model (DCOM) for Microsoft's Windows operating systems and the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) for Unix-based systems. The problem is that there is limited interoperability between these two technologies. If one component or application is based on DCOM and another component or application is based on CORBA, then they cannot communicate easily with one another. Special software called a DCOM/CORBA bridge must be used to accomplish the task.

- ▶ **Dynamic environment.** In today's rapidly changing business environment, business partners come and go, and so do software vendors and their applications. If a software component or application is no longer available because the vendor is no longer in business or has dropped a particular product line, then an existing EC application has to be flexible enough to substitute a new component or application for the old one. If a new business partner requires additional functionality, then an existing EC application has to be flexible enough to incorporate new features, functions, or applications. Again, existing EC application architectures make it difficult to accommodate these types of changes.
- ▶ **Security barriers.** Companies use firewalls to protect their networks against security risks. Firewalls are designed to limit the types of communications and requests that can be made from one computer to another. In most cases, only the simplest sorts of Web requests using standard communication protocols (such as HTTP) are allowed. This makes it very difficult for one component or application residing on one computer to communicate with another component or application residing on another computer.

For these reasons, there is a need for universal standards, and this is where Web Services, XML, and service-oriented architecture enter the picture.

THE ROLES OF XML AND WEB SERVICES

What is required to address these problems is a technology that can be integrated across different hardware and operating systems, that can interface with both new and legacy systems, and that minimizes network security risks. This is where Web Services come into play. According to the World Wide Web Consortium (W3C), a **Web Service** is a software system identified by a URI (uniform resource indicator) whose public interfaces and bindings can be defined and described using XML.

As the definition indicates, Web Services are based on XML (see Online Appendix B). The operations (or methods) that a Web Service can perform are "defined and described" using XML. Likewise, when another program or application wants to invoke the operations or methods of a Web Service, the request is sent as an XML message. An XML document or message is a text file with a set of tags and content (or values). The tags within an XML document (denoted by "<") describe the content. For instance, the following XML document might be used to represent an order placed by a customer for a digital camera:

```
<ORDER>
<ORDER_ID>123</ORDER_ID>
<ORDER_ITEM>Digital Camera ABC</ORDER_ITEM>
<ORDER_QUANTITY>1</ORDER_QUANTITY>
</ORDER>
```

Because XML messages are text based, they can be sent over the Web using standard Web communication protocols (e.g., HTTP). This makes it easy for programs or applications written in different program languages and running on different hardware to interoperate. It also means that the messages sent from one program or application to another can pass easily through firewalls.

WEB SERVICES

Web Services are self-contained, self-describing business and consumer modular applications, delivered over the Internet that users can select and combine through almost any device, ranging from personal computers to mobile phones. By using a set of shared protocols

Web Service

A software system identified by a URI (uniform resource indicator), whose public interfaces and bindings are defined and described using XML.

and standards, these applications permit disparate systems to “talk” with one another—that is, to share data and services—without requiring people to translate the conversation. The result promises to provide on-the-fly and in-real-time links among the online processes of different systems and companies. These links could shrink corporate IT departments, foster new interactions among businesses, and create a more user-friendly Web for consumers. Web Services provide for inexpensive and rapid solutions for application integration, access to information, and application development. Web Services can be extremely useful for EC, especially B2B structures.

Key Technologies in Web Services

In addition to XML, three other technologies are also instrumental in providing Web Services (for examples, see Iverson 2004 and Birman 2005). These include:

Simple Object Access Protocol (SOAP)

Protocol or message framework for exchanging XML data across the Internet.

- Simple Object Access Protocol (SOAP).** SOAP is the most frequently used protocol or message framework for exchanging XML data across the Internet. A SOAP message, which is written as XML, consists of three parts: an envelope, an optional header, and a body. The envelope encapsulates the message; the header provides optional information about the message; and the body is the XML data being exchanged. For example, the following SOAP message might be used to request the number of items in inventory available for purchase:

```
<SOAP-ENV:Envelope>
<SOAP-ENV:Body>
<s:getInventoryQuantity>
<item>Digital Camera ABC</item>
</s:getInventoryQuantity>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

When a program wants to invoke a process or method performed by a specific Web Service (e.g., `getInventoryQuantity` in the example), it simply sends a SOAP message to the service over the Web. In turn, the Web Service sends a SOAP message in response.

Web Services Description Language (WSDL)

An XML document that defines the programmatic interface—operations, methods, and parameters—for Web Services.

- Web Services Description Language (WSDL).** WSDL is an XML document that defines the programmatic interface for a Web Service. The document specifies the operations or methods that the Web Service can perform, along with the parameters that the service needs to carry out the operations and the values that the service will return in response to a particular request.

Universal Description, Discovery, and Integration (UDDI)

An XML framework for businesses to publish and find Web Services online.

- Universal Description, Discovery, and Integration (UDDI).** UDDI is a general business registry that originally was used as a way for the participants in a B2B exchange to share information about their business and business processes (Deitel et al. 2003). More recently, UDDI has been used as an XML framework for businesses to publish and find Web Services online. The Web Service entries in a UDDI typically point to the Web address (URL) of the WSDL file associated with the Web Service.

- Security protocols.** A **security protocol** is a communication protocol that encrypts and decrypts a message for online transmission; security protocols generally provide authentication. Several security standards are in development, including *Security Assertion Markup Language (SAML)*, which is a standard for authentication and authorization. Other security standards are XML signature, XML encryption, XKMS, and XACML.

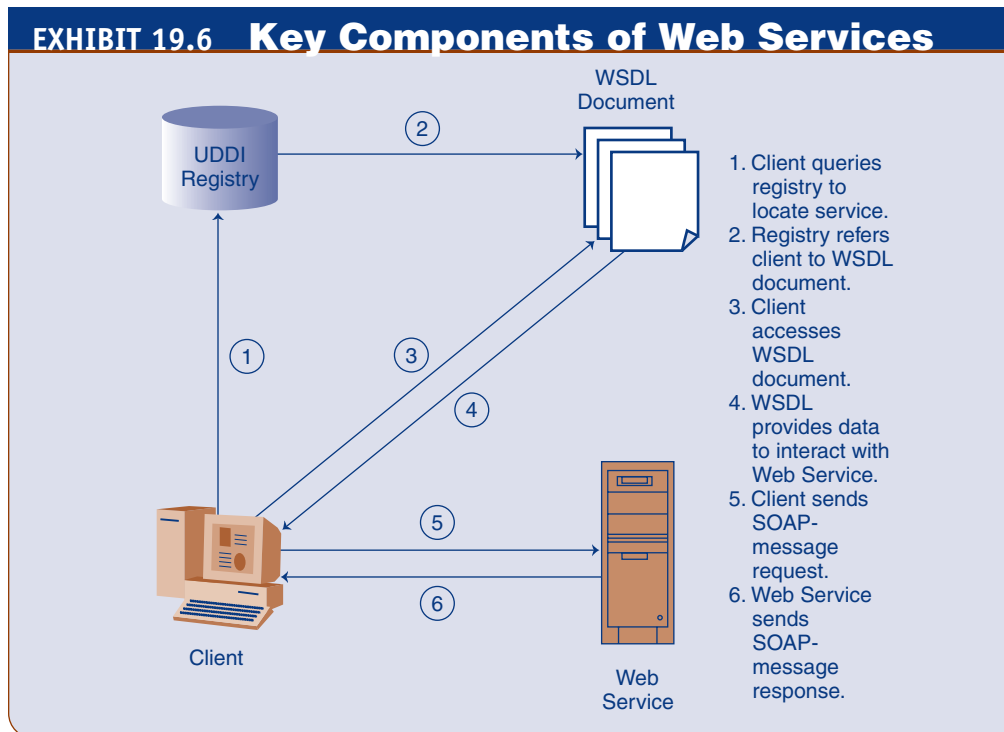
Exhibit 19.6 describes the interaction of the key components in a Web Service. Web Services are based on XML, as detailed in Online Appendix B.

security protocol

A communication protocol that encrypts and decrypts a message for online transmission; security protocols generally provide authentication.

Web Services Platforms

A number of the major hardware and software vendors have created software development environments that help programmers create and deploy Web Services. The development environments provided by three of the leaders in this arena—Microsoft, IBM, and Sun—are described briefly here. For a more extensive list of Web Services platforms, see Newcomer and Lomow (2005).



Source: Web Services: A Technical Introduction by Dietel, H., et al. ©2003 Pearson Education. Reprinted by permission of Pearson Education.

- Microsoft .NET.** Microsoft has been a leader in the Web Services marketplace (Rammer and Szpuszta 2005). Microsoft's .NET framework provides the foundation for Web Services that can be created and deployed on Windows 2000 and XP. The development environment for the .NET framework is Visual Studio .NET. Visual Studio .NET enables software developers to design, develop, and deploy Web Services with the major Windows programming languages—C++, C# (C Sharp), and Visual Basic .NET.
- IBM WebSphere.** The foundation of IBM's Web-based applications, including their EC offerings, is the WebSphere Application Server. Over the past couple of years, IBM has integrated various Web Services technologies (e.g., SOAP and WSDL) into its application server. To assist software and application developers with the design, development, and deployment of Web Services on the WebSphere platform, IBM has enhanced its existing development environment—the WebSphere Studio Application Developer—to support Web Services and has created a new development environment called the Emerging Technologies Toolkit (see Rodriguez et al. 2004). Section 19.5 provides a comprehensive discussion of the IBM WebSphere EC suite.
- J2EE Architecture.** With J2EE Architecture (java.sun.com/j2ee), developers have access to a complete development platform specifically designed to meet the needs of enterprise application development. J2EE makes all Java enterprise functionality available and accessible in a well-integrated fashion, simplifying complex problems in the development, deployment, and management of multitier server-centric enterprise solutions.

The Notion of Web Services as Components

People generally view information systems, including the Web, as relating to information (data) processing. Web Services enable the Web to become a platform for applying business services as components in IT applications. For example, user authentication, currency conversion, and shipping arrangement are components of broad business processes or applications, such as e-commerce ordering or e-procurement systems. (For further discussion, see Stal 2002.)

The idea of taking elementary services and joining them together to create new applications is not new. As described earlier, this is the approach of component-based development. The problem is that earlier approaches were cumbersome and expensive. According to Tabor (2002), early component-integration technologies exhibited problems with data format, data transmission, interoperability, flexibility (they are platform specific), and security. Web Services offer a fresh approach to integration. Furthermore, business processes that are composed of Web Services are much easier to adapt to changing customer needs and business climates than are “home-grown” or purchased applications.

A Web Services Example

As a simple example of how Web Services operate, consider an airline Web site that provides consumers the opportunity to purchase tickets online. The airline recognizes that customers also might want to rent a car and reserve a hotel as part of their travel plans. The consumer would like the convenience of logging onto only one system rather than three, saving time and effort. Also, the same consumer would like to input personal information only once.

The airline does not have car rental or hotel reservation systems in place. Instead, the airline relies on car rental and hotel partners to provide Web Service access to their systems. The specific services the partners provide are defined by a series of WSDL documents. When a customer makes a reservation for a car or hotel on the airline’s Web site, SOAP messages are sent back and forth in the background between the airline’s and the partners’ servers. In setting up their systems, there is no need for the partners to worry about the hardware or operating systems each is running. Web Services overcome the barriers imposed by these differences. An additional advantage for the hotel and car reservation systems is that their Web Services can be published in a UDDI so that other businesses can take advantage of their services.

Web Services Entering the Mainstream

Popular sites on the Web also are turning to Web Services as a way of providing access to their internal systems. In this way, the content and operations supported by these systems can be transparently integrated with applications developed by other companies or individuals; Google’s and Amazon.com’s services have received the most publicity.

Google’s Web Services API (application programming interface) enables programmers and application developers to issue SOAP-based search requests to Google’s index of more than 3 billion Web pages and to receive results as XML data. Google’s Web Services provide a range of possibilities, including issuing regularly scheduled search requests to monitor the Web for new information on a subject; performing market research by analyzing differences in the amount of information available on different subjects over time; and searching via non-HTML interfaces, such as the command line, pagers, or visualization applications.

Amazon.com also offers an extensive set of Web Services that can be used by its “Associates” and other product sellers and vendors. The Amazon.com Associates program enables Web sites to link to Amazon.com’s site and to earn money for sales generated through the link. For Associates, Web Services provide a way to make their Web sites fresher and more dynamic. For example, Associates can use Amazon.com’s Web Services to dynamically retrieve prices, generate lists of products, display search results, produce recommendation lists, and even add items to the Amazon.com shopping cart directly on their Web sites (*Amazon.com* 2007). Other companies use Amazon.com’s development platform to sell their products on Amazon.com’s Web site. In this case, Amazon.com’s new Web Services enable these companies to do things such as manage their inventory, generate orders, and produce competitive pricing information. For an application in a large bank, see Case 19.1 (p. 8).

Advantages of and Barriers to Implementing Web Services

Web Services offer a number of distinct advantages over previous programming initiatives that have attempted to solve the problem of interoperability (i.e., getting software and applications from different vendors running on different hardware and operating systems to communicate with one another in a transparent fashion). According to Deitel et al. (2003) and Shirky (2002), these advantages include the following:

- ▶ Web Services rely on universal, open, text-based standards that greatly simplify the problems posed by interoperability and that lower the IT costs of collaborating with external partners, vendors, and clients.

- ▶ Web Services enable software running on different platforms to communicate, reducing the cost and headaches of multiple platforms running on everything from mainframes to servers to desktops to PDAs.
- ▶ Web Services promote modular programming, which enables reuse by multiple organizations.
- ▶ Web Services are easy and inexpensive to implement because they operate on the existing Internet infrastructure. They also offer a way to maintain and integrate legacy IT systems at a lower cost than typical EAI efforts.
- ▶ Web Services can be implemented incrementally rather than all at once.

However, Web Services are not a panacea. Among the current barriers inhibiting widespread adoption of the technology are:

- ▶ The standards underlying Web Services are still being defined; thus, interoperability is not automatic. Even in those instances where the standards are relatively stable, it still requires programming skill and effort to implement and deploy Web Services.
- ▶ One area where the Web Services standards are not well defined is security. In terms of interoperability, the good news is that Web Services enable distributed applications to communicate with ease. The bad news is that Web Services also enable applications to bypass security barriers (such as firewalls) with ease. Standards such as XML, SOAP, WSDL, and UDDI say nothing about privacy, authentication, integrity, or nonrepudiation.

In an effort to bridge the gap between these standards and existing security standards (such as public key encryption), several vendors and organizations have proposed and are currently debating a number of Web Service security standards. One of these is WS-Security, which is being proposed by Microsoft, IBM, and VeriSign.

- ▶ Although Web Services rely on XML as the mechanism for encoding data, higher-level standards are still required, especially in B2B applications. For example, if two banks want to communicate, they still need standards to define the specific content that is being communicated. This is where standards such as OFX (Open Financial Exchange), which defines the content of transactions between financial institutions, come into play. The lack of coordination among all interested parties for high-level standards is why Web Services will be adopted first within organizations and later across organizations.

SECOND-GENERATION WEB SERVICES

As the Internet matures and Web software becomes more sophisticated, an exciting second generation of Web Services is leading to less programming and more assembly, putting the power of the programmer into the hands of the user. A number of innovative products that provide a drag-and-drop interface to weave Web Services together to build small applications are making their way onto the market (Malik 2006). Essentially, all that is needed is an Internet connection, a browser, and a good imagination. A representative sample of user-friendly software tools that take the programmer out of building custom business applications is shown in Exhibit 19.7. A number of techniques supporting this move to empower the user-developer are discussed in the following sections.

EXHIBIT 19.7 Representative Products to Create User-Generated Web Applications		
Product Name	Company	URL
Coghead	Coghead	<i>coghead.com</i>
Teqlo	Teqlo	<i>teqlo.com</i>
Dabble DB	Smallthought Systems	<i>dabbledb.com</i>
Project Runner	BEA Systems	<i>bea.com/framework.jsp?CNT=index.htm&FP=/content/products&WT.ac=topnav_products</i>
Project Builder	BEA Systems	<i>bea.com/framework.jsp?CNT=index.htm&FP=/content/products&WT.ac=topnav_products</i>
QEDWiki project	IBM	<i>ibm.com</i>

Web 2.0

A second generation of Web Services that emphasizes online collaboration and information sharing among users.

social network

A social network is a category of Internet applications that help connect friends, business partners, or other individuals together using a variety of tools.

Ajax

A Web development technique for creating interactive Web applications.

Really Simple Syndication (RSS)

A family of Web-feed formats used to publish frequently updated digital content.

viral video

Video clip that gains widespread popularity through the process of Internet sharing, typically through e-mail or IM messages, blogs, and other media-sharing Web sites.

mashup

Combination of two or more Web sites into a single Web site that provides the content of both sites (whole or partial) to deliver a novel product to consumers.

Web 2.0

The latest generation of Web Services is **Web 2.0**. The term *Web 2.0* was coined by O'Reilly Media (O'Reilly 2005) to refer to a second generation of Web Services, which emphasizes online collaboration and information sharing among users. Web 2.0 engages users on a seemingly one-to-one basis in a way that first-generation Web tools were never able to accomplish. The most popular Web 2.0 applications include **social networks**. A social network is a category of Internet applications that help connect friends, business partners, or other individuals using a variety of tools, for example, **Ajax**, a Web development technique for creating interactive Web applications, and **Really Simple Syndication (RSS)**, a family of Web-feed formats used to publish frequently updated digital content.

The earliest example of an Internet social network was Classmates (classmates.com). More recently, Yahoo! Days (yahoo.com), MySpace (myspace.com), and Facebook (facebook.com) have become increasingly popular tools to connect with and make friends. As these technologies mature, a social marketplace is evolving. This move to an enterprise use of Web 2.0 that harnesses the power of social networks and the marketplace to allow enterprises to expand their contact base and/or advertise and sell their products and services has been termed Enterprise 2.0 (McAfee 2006). An example of an innovative social network is presented in Case 19.2.

Not surprisingly, new software tools, such as IBM's WebSphere Community Edition (Rauch 2007), Intel's SuiteTwo (LaMonica 2006), Microsoft's Expression Web (Rapoza 2006), and Oracle's WebCenter (Hinchcliffe 2006), are being introduced to support the Enterprise 2.0 environment as companies realize its potential to engage customers and generate content for their Web sites in a way not previously possible. The potential to add an additional dimension to a flat consumer Web site and to develop closer connections to a company's customer base is becoming very attractive to an increasing number of businesses. The latest generation of the Web does not just link information, it links people. For example, a company can allow consumers who visit its Web site to create their own profiles and rate and comment on products. Consumers can also create their own topic areas within forums and build online communities around shared interests pertaining to the company's products. Those sites with streaming video capabilities can also encourage consumers to provide user-supplied videos of customers using their products and solicit consumer feedback on beta versions of product commercials before final release. Just imagine the immeasurable value to small firms that are unable to afford large market research and advertising campaigns; essentially, a Web 2.0 environment provides an ongoing 24/7 customer focus group, supplying firms with an unending stream of data to analyze and act upon. In February 2007, Patricia Seybold, founder and CEO of the Patricia Seybold Group, encouraged firms to consider the enhanced capabilities provided by Web 2.0 when she commented that, "companies should be leveraging online tools and online communities to leverage customer fragmentation and to address more customers' needs, not fewer." The move to the development of Web 2.0 tools is shown in Exhibits 19.8 (p. 24) and 19.9 (p. 25).

Viral Video

The video dimension of Web 2.0 capabilities is referred to as *viral video*. The term **viral video** refers to video clip content that gains widespread popularity through the process of Internet sharing, typically through e-mail or IM messages, blogs, and other media-sharing Web sites. Popular sites that embody this concept include YouTube (youtube.com), AOL Comedy (comedy.aol.com/viralvideos), ifilm (ifilm.com) and VEOH (veoh.com). The marketing potential of viral video has yet to be fully explored, but companies that offer viral video services to businesses are beginning to emerge, and companies are beginning to use preexisting social networks to produce exponential increases in brand awareness. See Wilson (2005) for the six principles of viral marketing. Exhibit 19.10 diagrams the emergence of mass social media combining Web Services, social networks, and viral video.

Mashups

Another novel approach to developing second-generation Web sites is the concept of a *mashup*.

A **mashup** is essentially the combination of two or more Web sites into a single Web site that provides the content of both sites (whole or partial) to deliver a novel product to

CASE 19.2

EC Application

A SOCIAL NETWORKING SLAM DUNK BY PORTLAND TRAIL BLAZERS

When the Portland Trail Blazers launched the first social network for professional basketball fans on February 9, 2007, fans quickly used the new Web site (iamatrailblazersfan.com), to help Trail Blazers' power forward Zach Randolph gain a slot in the NBA All-Star game the following week.

The site, the purpose of which is to help Portland Trail Blazers' management better communicate with the team's fans and increase ticket sales, signed up an astounding 2,500 registered users in the first 3 days of operation. The Portland, Oregon, NBA franchise partnered with Hands on Portland to connect the team, the fans, and the Portland community in an innovative way.

Utilizing Affinity Circles proven social networking platform, the Trail Blazers created an online community so fans could share their passion for the team. "Portland fans' passion for the Trail Blazers is legendary, and many people in our community share that same passion for technology, so merging the two to create an official online community to tap into those passions makes perfect sense," said team President and General Manager, Steve Patterson. "Our network will further connect fans throughout the city, across the country, and around the world. Partnering with Affinity Circles allows us to deliver an authentic experience for our fans with a proven trusted social networking platform."

On the site, fans can create profiles, join groups, share photos, post comments, and much more. The site is designed for easy and fast deployment, and because it is a hosted solution, it can grow as fast as the fan base requires.

The site gives fans the opportunity to use their online connections with one another as a jumping off point for real connections in the community and on the court. The online community features a network for volunteerism via Portland's volunteer center and Trail Blazers' community partner, Hands On Portland. It also provides the opportunity for members to form groups that can meet up to play games at local basketball courts or attend a Trail Blazers game, all with the comfort of knowing that personal communications are taking

place within an official setting sanctioned by the Trail Blazers' organization.

"Our partnership with the Trail Blazers allows us to extend our track record of creating trusted social networking community for well known affinity-based organizations while demonstrating the application of this technology in the professional sports arena," said Steve Loughlin, chief executive officer of Affinity Circles. "The market response to our products has been tremendous, and we believe our growth opportunities are significant. We are excited about working closely with the Trail Blazers to launch the first fan network for an NBA team, an online experience that will serve the team, its fans, and the city of Portland."

The team joins a growing number of businesses that are embracing Web 2.0 tools such as wikis, blogs, and podcasts to create social networks to enable customers to add content and create virtual communities around their brand. These include General Motors Corp. (for their Pontiac and OldSpice brands), American Society for Prevention of Cruelty to Animals (ASPCA), and Travelocity's Experience Finder, a new site it launched to help customers choose travel destinations.

Sources: Compiled from Havenstein (2007), Sasse (2006), and Portland Oregon Trail Blazers (2007).

Questions

1. What advantages, other than those discussed in the case, might the Trail Blazers gain from the social network it has established?
2. Search the Web for other sports entities that have established social networks to communicate with their customers. How many did you find? In what ways are they the same or different from iamatrailblazersfan.com?
3. Make a list of other types of companies that you think could benefit from a social networking site and discuss why.

consumers. Mashups reach into the Application Program Interface (API) for a given application and extract information, including Web page elements, and use them to launch a new application that adds value. The most popular type of mashup relates to maps (e.g., Google Maps, Yahoo! Maps), which can be combined with other data sources to produce interesting results (Hoff 2006). Examples of such value-adding mashups include chicagocrime.org, which combines local crime stats with Google Maps to show crimes committed by geographical area, and 1001Seafoods.com, which mashes up a database of fishing holes with Google Maps to create a U.S. map that shows fishing sites and offers directions and information on the location and sources for bait and tackle. Combining Google Maps and real estate information, Zillow (zillow.com) enables users to zoom in on a neighborhood of interest to see the environment and examine home values, home sales' history, and so on. For those who prefer to rent, HousingMaps (housingmaps.com) offers a combination of Craigslist rentals with

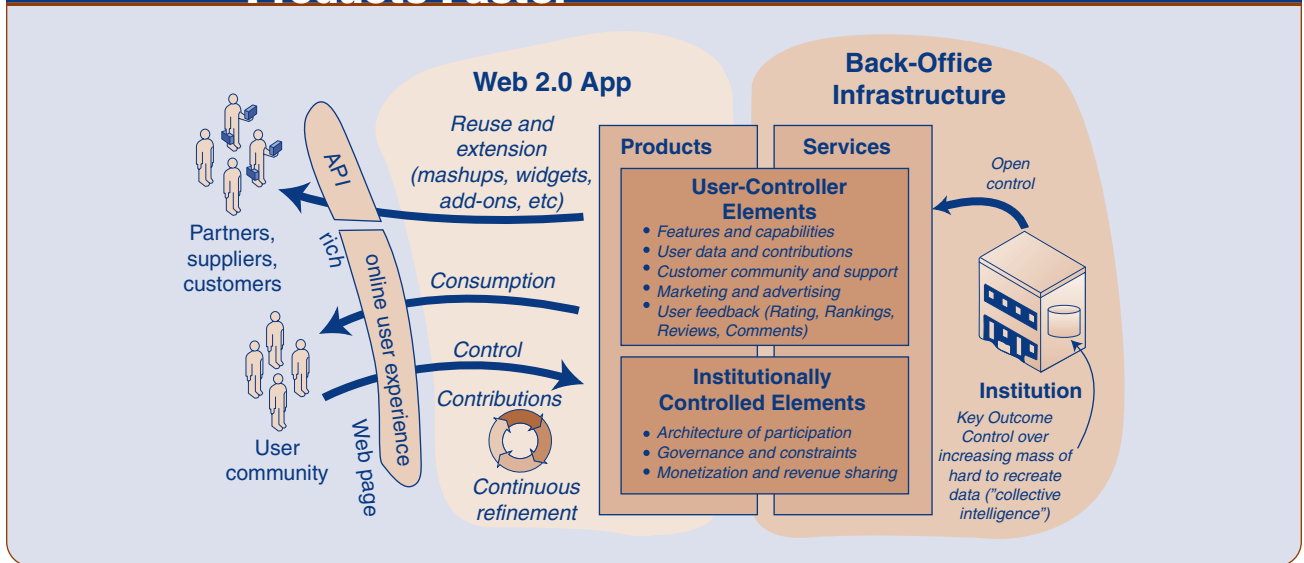
EXHIBIT 19.8 The Move to Product Development 2.0

	Product Development 1.0	Product Development 2.0
Primary customer interaction channel	Telephone, mail, face-to-face, one-way media (print, TV, radio, etc.), e-mail	World Wide Web, e-mail, IM
Source of innovation	Organizations	Customers
Innovation cycle	Months, years	Minutes, hours, days, weeks
Content creators	Internal producers	External producers
Feedback mechanisms	Market research, satisfaction surveys, complaints, focus groups	Analytics, online requests, user-contributed changes
Customer engagement style	Controlled, well-defined process	Spontaneous and chaotic
Product development process	Upfront design	Less upfront, much more emergent
Product architecture	Closed, not designed for easy extension or reuse by others; walled garden	Open, very easy to extend, refine, change and add on to, ecosystem friendly, designed (and legal) for widespread remixing and mashups
Product development culture	Hierarchical, centralized, <i>not invented here</i> , somewhat collaborative, expert driven	Egalitarian, decentralized, remix instead of reinvent, highly collaborative, wisdom of crowds
Product testing	Internal, dedicated test groups, hand-picked select customers	Users as testers
Customer support	Customer service	User community
Product promotion	One-way marketing and advertising	Viral propagation, explicit leveraging of network effects, word of mouth, user-generated and two-way advertising
Business model	Product sales, customer service and support fees, service access charges, servicing high-demand products	Advertising, subscriptions, product sales, Servicing all product niches (the long tail), unintended uses
Customer relationship	External buyer (consumer)	Partner and—increasingly remunerated—supplier (<i>consumers as producers</i>)
Product ownership	Institution, particularly executive management and shareholders	Entire user community
Partnering process	Formal, explicit, infrequent, mediated	Ad hoc, thousands of partners online, disintermediated
Product development and integration tools	Heavyweight, formal, complex, expensive, time consuming, enterprise oriented	Lightweight, informal, simple, free, fast, consumer oriented
Competitive advantage	Superior products, legal barriers to entry (IP protections), brand name advantage, price, popularity, distribution channel agreements	Number one or two market leader, leveraging crowd sourcing effectively, mass customization, control over hard-to-create data, end user sense of ownership, popularity, cost-effective customer self-service, audience size, best-of-breed architectures of participation

Google Maps to show rental opportunities in nearly 40 regions across the United States. Other mashups combine travel, shopping, sports, news, video, and photo sites. An interesting site that tracks the mashup phenomenon is ProgrammableWeb (programmableweb.com), which offers a mashup dashboard showing all the latest mashup sites. One of its most powerful tools is its “mashup matrix” (programmableweb.com/matrix), which presents approximately 200 Web sites in a grid with dots that highlight the intersections where two sites have been brought together in a mashup. At each intersection between sites, pop-up boxes list all mashup that have been created using the sites; in many cases, there are multiple sites at each intersection. As of March 2007, the ProgrammableWeb matrix recorded the existence of nearly 1,600 mashups, with new mashups appearing at the rate of three new mashups per day. The rapid increase of number of mashups is shown in Exhibit 19.10.

Current software available to assist in user development of mashups includes Dapper (dapper.net), an add-on to the Firefox browser, and Yahoo! Widget Engine 3.1

EXHIBIT 19.9 Product Development 2.0: Using the Web to Put Users in Control and Co-Create Better, Richer Products Faster



Source: Dion Hinchcliffe Web 2.0 Blog at web2.wsj2.com.

(widgets.yahoo.com), a JavaScript runtime engine for Windows and Mac OS X that lets users run small files called Widgets (e.g., alarm clocks, calculators, weather, stock reports, and hundreds of other applications) to create a mashup Web site or desktop.

All of these new Web development tools and techniques require the support of an integrative architecture, known as service-oriented architecture (SOA), which is described next.

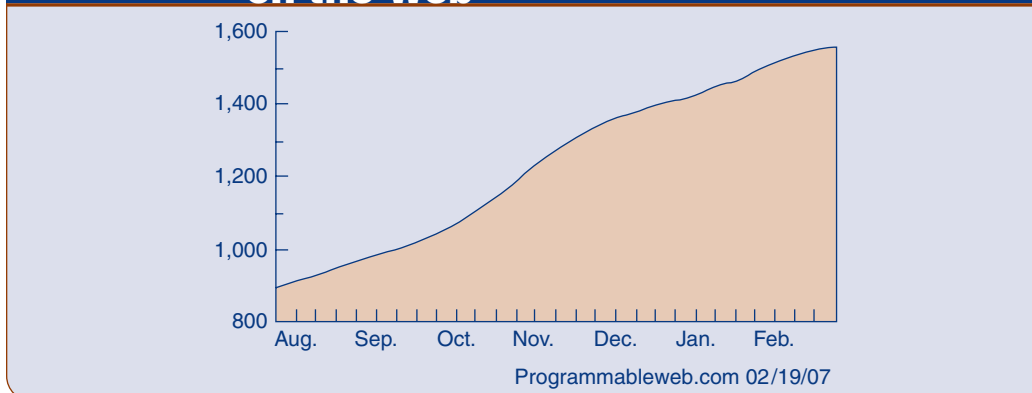
SERVICE-ORIENTED ARCHITECTURE (SOA)

Service-oriented architecture (SOA) is a term that has emerged to describe executable components—such as Web Services—that can be invoked by other programs that act as clients or consumers of those services (Finkelstein 2005). Exhibit 19.11 depicts the conceptual roles and operations of an SOA. The basic premise of SOA is to *reuse* and *reconnect* existing IT assets (also called *services*) rather than more time consuming and costly reinvention of new systems. A growing number of firms are addressing the constraints of their current IT architecture and infrastructure by moving to an SOA. In June 2005, a Gartner Group report (Cantara 2005) named SOA as one of the five hottest technology topics of

service-oriented architecture (SOA)

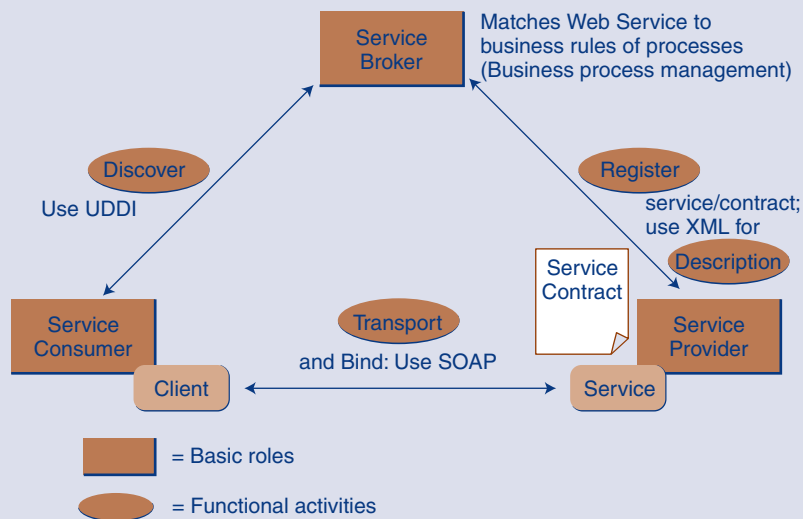
An application architecture in which executable components, such as Web Services, can be invoked and executed by client programs based on business rules.

EXHIBIT 19.10 The Rapid Increase of Mashups on the Web



Source: Dion Hinchcliffe Web 2.0 Blog at web2.wsj2.com.

EXHIBIT 19.11 The Three Conceptual Roles and Operations of a Service-Oriented Architecture



2005 and predicted that 80 percent of all software development projects will be SOA driven by the year 2008. The benefits of SOA adoption are shown in Exhibit 19.12.

The three basic SOA roles are the service provider, the service consumer, and the service broker. A *service provider* makes the service available and publishes the contract that describes its interface. It then registers the service with a *service broker*. A *service consumer* queries the service broker and finds a compatible service. The service broker gives the service consumer directions on where to find the service and its service contract. The service consumer uses the contract to bind the client to the service. For the three conceptual roles to accomplish the three conceptual operations, an SOA system must supply four core functional architecture components:

- ▶ **Transport.** The transport component represents the formats and protocols used to communicate with a service.
- ▶ **Description.** The description component represents the language used to describe a service. The description provides the information needed to tie it to a service.
- ▶ **Discovery.** The discovery component represents the mechanism used to register or advertise a service and to find a service and its description.
- ▶ **Registration.** The registration component registers the services of a service provider with a service broker.

With SOA and Web Services, functions within existing programs and suites, as well as functions within ERP (enterprise resource planning), CRM (customer relationship management), SCM (supply chain management), and other packages, can be invoked automatically and executed anywhere in the world based on business rules. A developer does not need to know how the programs work, only the input that they require, the output they provide, and how to invoke them for execution. Developers can also swap out one service and replace it with another service

EXHIBIT 19.12 Benefits of Service Oriented Architecture

- Reduced integration cost
- Improved business/IT alignment
- Extension of leveraging of existing IT investments
- Faster time to assemble new applications
- Lowest IT maintenance cost

that is designed to achieve the same or an enhanced result without having to worry about the inner workings of the services. SOA and Web Services are expected to totally transform the way EC systems are built, moving from slow, error-prone manual coding to automation.

According to Then (2005), the essence of an SOA is reusable services and the ability to link disparate applications. EC service components (such as credit card processing or bills payment) can be created once and then reused to drive better cost-efficiency from EC and IT investments. What an SOA aims to achieve is to align IT with real business requirements. Hence, if a business user has a requirement, it can be translated into services in an SOA environment. A number of software vendors are currently competing in the SOA environment, including IBM, with its award winning WebSphere Portal V6, which offers improved efficiency and productivity through a “significantly enhanced user experience and additional personalization capabilities” (IBM 2007a). A list of other SOA products can be found at en.wikipedia.org/wiki/List_of_SOA_related_products. For a comprehensive discussion on the usability of SOA, see Bieberstein et al. (2005), Tyler (2005), and en.wikipedia.org/wiki/Service-oriented_architecture. An up-and-coming subset of SOA that provides business partners with a way to connect, integrate, and mashup is *Web-oriented architecture*. The term **Web-oriented architecture (WOA)** was coined by Nick Gall of the Gartner Group and is used to describe a set of Web protocols (e.g., HTTP and plain XML) as the most dynamic, scalable, and interoperable Web Services approach. According to Hinchcliffe (2006), the main difference between a traditional SOA and a WOA is that a WOA advocates REST, an increasingly popular, powerful, and simple method of leveraging HTTP as a Web Service. **Representational State Transfer (REST)** refers to a collection of architectural principles. The term also is used in a loose sense to describe any simple interface that transmits domain-specific data over HTTP without an additional messaging layer. Hinchcliffe is quick to note, however, that “WOA is definitely not the hammer for every job.” He goes on to say that certain applications, particularly in the high-end of the enterprise, require the more sophisticated component of SOA. However, for the vast majority of uses, it would appear that WOA is the most interoperable, easiest to implement, and most highly scalable technique for building open Web Services that anyone can use.

To demonstrate the integration of Web Services and SOA, Application Case 19.3 provides an example of a Web-based service provider.

Another concept in IT infrastructure that has matured to enable the development of e-commerce applications is *virtualization*.

VIRTUALIZATION

Virtualization enables the sharing and/or aggregation of physical resources, such as operating systems, software, and IT services, in a way that hides the technical detail from the end users and reduces the per unit service cost. Because the virtualization system sits between the guest and the hardware, it can control the guest’s use of CPU, memory, and storage, even allowing a guest OS to migrate from one machine to another.

Originally developed in the 1960s to partition mainframe hardware, virtualization declined in popularity as minicomputers and PCs provided a more efficient, affordable way to distribute processing power. Recently, decreasing hardware costs, underutilization of resources, escalating maintenance costs, and security issues have caused IT professionals to take another look at virtualization as a way to help businesses address issues of scalability, security, and management of their IT infrastructure. The benefits of virtualization are shown in Exhibit 19.13.

Virtualization Products

Not surprisingly, there has been a visible surge in virtualization-related startups, acquisitions, and partnerships, as well as related academic research. Dozens of virtualization products are available, offered by a number of small and large companies. Some examples in the operating systems and software applications space are VMware (vmware.com), Xen (xensource.com), and Microsoft Virtual Server (microsoft.com/windowsserversystem/virtualserver/default.aspx).

Major IT players also have shown a renewed interest in the technology. For example, IBM—a long-time pioneer in the area of virtual machines—has not only strengthened its own virtualization-related products but has forged a relationship with VMware. Sun and HP also are improving their virtualization offerings (IBM 2007b). Both Intel and AMD now make processors that explicitly support virtualization. Microsoft acquired Connectix Corporation, a developer

Web-oriented architecture (WOA)

A set of Web protocols (e.g., HTTP and plain XML) as the most dynamic, scalable, and interoperable Web Service approach.

Representational State Transfer (REST)

Refers to a collection of architectural principles.

virtualization

A technique for hiding the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources.

EXHIBIT 19.13 Benefits of Virtualization

Partitioning

- Multiple applications and operating systems can be supported within a single physical system.
- Servers can be consolidated into virtual machines on either a scale-up or scale-out architecture.
- Computing resources are treated as a uniform pool to be allocated to virtual machines in a controlled manner.

Isolation

- Virtual machines are completely isolated from the host machine and other virtual machines. If a virtual machine crashes, all others are unaffected.
- Data do not leak across virtual machines, and applications can only communicate over configured network connections.

Encapsulation

- Complete virtual machine environment is saved as a single file; easy to back up, move, and copy.
- Standardized virtualized hardware is presented to the application, guaranteeing compatibility.

of virtualization software for Windows and Macintosh platforms, and Softricity, an application streaming developer. EMC acquired VMware, and Cisco acquired several companies that develop products in the storage, file, and network virtualization spaces. Some interesting insights into the new virtualization phenomenon are provided by Dr. Mladen Vouk of North Carolina State University in Insights and Additions 19.1. Dr. Vouk is one of the principals of the development and operation of the Virtual Computing Laboratory in Raleigh, North Carolina.

CASE 19.3

EC Application

VERIO SETS ITS SIGHTS ON ONLINE SMALL-TO-MEDIUM FIRMS

On December 13, 2006, Verio (*verio.com*) announced the roll out of its latest version of its Web-based shopping cart tool—ShopSite 8.1. The software will help its Web hosting customers create and manage successful online stores without complex installation or high cost. ShopSite 8.1 enables SMEs to manage inventory, payments, and sales records on their Web sites. It is compatible with multiple payment processing services, including PayPal, Authorize.Net, and WorldPay.

Support for the new Google Checkout allows merchants to accept credit card payments without having to obtain a merchant account, and Verio customers who configure ShopSite in accordance with Payment Card Industry Security Practices guidelines can be confident that their credit card and order information are safe and secure. “By incorporating a direct e-commerce solution such as ShopSite into their Web sites, Verio customers can help to increase customer loyalty by improving the shopping experience on their sites,” said Dennis Boyle, Verio’s chief operating officer in a statement.

ShopSite 8.1 is certified by VISA as a Payment Application Best Practices applicant and will automatically pass purchasing information by VeriSign’s Fraud Control system. Additionally, a new auction-integration function allows customers with eBay Seller accounts to offer products for auction on eBay directly from the ShopSite back-office interface.

DY Home Décor, an online retail store that sells home décor merchandise, selected Verio Web hosting with the ShopSite tool, which integrated seamlessly with its site. Through the ShopSite tool, DY Home Décor is able to provide its customers with a high level of security and sophisticated

features, such as FedEx real-time quotes, which complements the company’s global shipping capabilities.

ShopSite 8.1 Starter comes preinstalled with Verio’s Hosting 2000 and Windows eStorefront hosting plans; it can be added to other Verio hosting plans for \$9 per month. Customers can upgrade ShopSite to the Manager or Pro level for an additional monthly fee.

Verio hosts Web sites at various service levels. At each level, Verio offers state-of-the-art services, a tier-1 network, and a team of experts to assist in selecting applications or designing a system. It also promises availability (get started quickly), reliability (99.9 percent uptime), confidentiality (secure files), data integrity (daily backups are performed to protect files from loss or corruption), throughput (pages come up quickly), scalability (an account can be upgraded as the client grows), support (24/7 access), control (easily changed Web design), and information (Web site traffic statistics, etc.). Verio clients can acquire self-service accounts or simple transactional functionality, or they can work closely with Verio to design and build robust systems to be hosted by Verio.

Sources: Compiled from WHIR News (2006) and Prince (2006).

Questions

1. Identify the major features of ShopSite 8.1.
2. What are the value-added services Verio offers?

Insights and Additions 19.1 A Virtualization Example: “The Next Big Thing”

At the computational-resource level, virtualization focuses on the delivery of operating system or software application functions. This includes the virtual desktop infrastructure, traditional server-based solutions, Web-based solutions, terminal server solutions, and more recently, application streaming and “image”-level services. An underlying assumption of virtualization is that storage and networking are available and may also be virtualized.

Virtualization Methods

Depending on how much and which parts of the resource(s) are shared, one can distinguish a plethora of virtualization methods operating system-level virtualization and application virtualization (e.g., software as a service). These methods range from **emulation** (enables computer programs to run on a platform other than the one for which they were originally written) and **simulation** (attempts to gather a great deal of runtime information, as well as reproducing a program’s behavior; emulation attempts to model to various degrees the state of the device being emulated) to **native virtualization** (software that leverages hardware-assisted capabilities available in the latest processors to provide near-native performance).

Typical virtualization categories are illustrated in Exhibit 19.14. Today, an operating system typically runs on “bare metal,” or natively, on a computer (e.g., a desktop unit, a racked server). During its operation, it has sole use of all the computer resources assigned to it (input/output devices, memory, processor, etc.), but it may share storage and network resources. An advantage is that the end user owns the equipment and uses it at will. The downside is that the end user may be responsible for installation and maintenance; that as hardware ages its capabilities may not match new software; and, if it breaks, another resource on which to work with one’s software and data (provided it is backed-up) may not be available. Furthermore, the resource may be idle a lot of time. Remote shared-server-based services may offer an alternative for improving utilization, functionality, and maintenance, but without virtualization the cost may not be justifiable.

Implementation

Individual virtualization products (e.g., VMware) can be installed and used on their own. However, most enterprises typically have a range of resource needs—from single desktops to server-based services to high-performance computing—and a single solution usually does not offer all the

emulation

A software emulator allows computer programs to run on a platform (computer architecture and/or operating system) other than the one for which they were originally written.

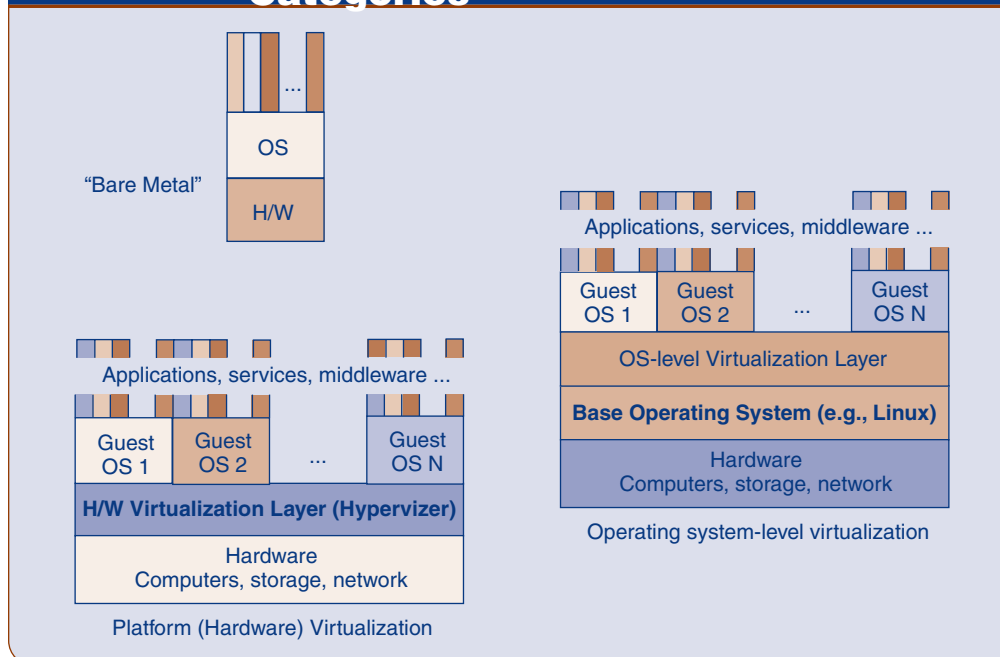
simulation

Attempts to gather a great deal of runtime information as well as reproducing a program’s behavior.

native virtualization

Leverages hardware-assisted capabilities available in the latest processors to provide near-native performance.

EXHIBIT 19.14 An Illustration of Virtualization Categories



The VCL is supported in part by the IBM Shared University Program, NC State College of Engineering and Information Technology Division and DOE SciDAC grant DE-FC02-01ER25484 (Used with permission of Dr. Mladen Vouk, North Carolina State University - vouk@ncsu.edu).

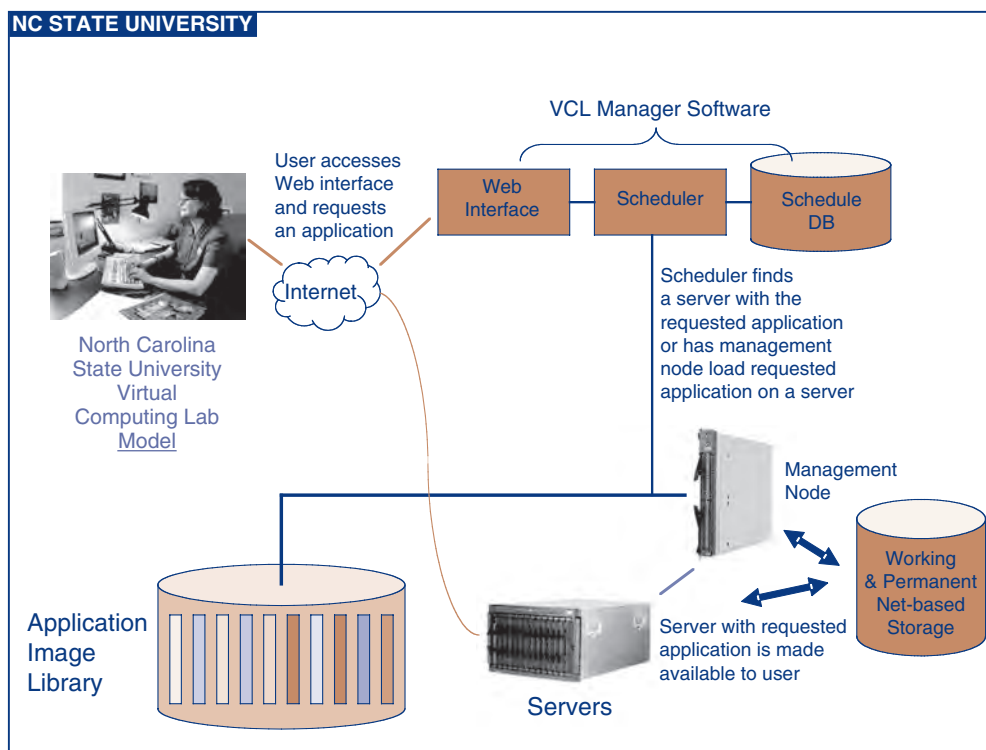
(continued)

Insights and Additions 19.1 (continued)

options. One way to handle this is through a distributed resource integration and utilization and load-balancing solution that can offer a wide variety of IT services.

To illustrate the concept, we focus on the NC State University Virtual Computing Laboratory (VCL). VCL is a secure, production-level, on-demand, broad capability utility computing architecture and technology for accessing solutions based on real and virtualized computational, storage, and software resources. The key differentiators of VCL are its simplicity; versatility, security, and cost-effectiveness; broad resource-based approach to “virtualization”; and its flexible delivery of resource services through “images.”

A user accesses VCL through a Web interface to select a combination of applications, operating systems, and services from a menu. If a specific combination is not already available as an “image,” an authorized user can construct one from the VCL library components. This customization capability is very much in the spirit of what services engineering and management is all about. VCL Manager software then maps the request onto available software application images and available hardware resources; it then schedules it for immediate use (on demand) or for later use. VCL delivers a range of options—from single real or virtual computer laboratory “seats” or desktops, to single applications on demand, to classroom-size groups, to enterprise server solutions, to research clusters, to high-performance computing services.



This work is supported in part by the IBM Shared University Program, NC State College of Engineering and Information Technology Division, and DOE SciDAC grant DE-FC02-01ER25484.

Choosing the Platform

Computational hardware and storage can be any appropriate platform—from an IBM blade center, to a collection of diverse desktop units or workstations, to an enterprise server, to a high-performance computing engine. Storage, communications, and other hardware are subsumed. VCL Manager provides appropriate aggregation or disaggregation of the available hardware resources before mapping the requested image(s) onto that hardware. The solution is deployed in VCL management, image, and portal nodes. It allocates the appropriate number of server resources for the task at hand, automatically installing the images, and distributing them across those resources to optimize performance.

Conclusion

VCL is a simple, but effective, virtualization tool that enables NC State University students and faculty to tap into the power of high-end laboratories directly from their laptops or computers from anywhere on campus—or from anywhere in the world. From the NC State perspective, the network of VCL nodes and services NC State operates is a *grid that works*. It is secure, and it has a scalable, maintainable, and sustainable architecture for delivery of a variety of diverse service environments *anytime* and *anyplace on demand* or *by reservation*. It has *broadly adjustable* server resources matched to service environment requirements ranging from virtual servers, to single a physical server, to multiple physical servers for *dedicated* single users or *shared* multiple users. Most importantly its architecture is *SOA compliant*.

Source: Dr. Mladen Vouk, North Carolina State University, Box 8206, Raleigh, NC 27695, vouk@ncsu.edu.

OTHER DEVELOPMENT OPTIONS

Besides the three major options for developing EC applications (buy, develop in-house, and outsource/lease), several other options are currently available and are appropriate under certain circumstances:

- ▶ **Join an e-marketplace.** With this option, the company “plugs” itself into an e-marketplace. For example, a company can place its catalogs in Yahoo!’s marketplace. Visitors to Yahoo!’s store will find the company’s products and will be able to make purchases. The company pays Yahoo! monthly space-rental fees. In such a case, Yahoo! is a hosting service for the company as well. As for development, the company will use templates to build its store, and it can start to sell after only a few hours of preparation work.
- ▶ **Join an auction or reverse auction third-party site.** Joining a third-party site is yet another alternative. Again the plug-in can be done quickly. Many companies use this option for certain e-procurement activities.
- ▶ **Joint ventures.** Several different joint-venture partnerships may facilitate e-business application development. For example, four banks in Hong Kong developed an e-banking system. In some cases, a company can team up with another company that already has an application in place.
- ▶ **Join a consortium.** This option is similar to the previous one, except that the company will be one of the e-market owners. Thus, the company may have more control over the market architecture.
- ▶ **Hybrid approach.** A hybrid approach combines the best of what the company does internally with an outsourced strategy to develop contracted partnerships. Hybrid models work best when the outsourced partner offers a higher level of security, faster time-to-market, and service-level agreements.

Section 19.3 ▶ REVIEW QUESTIONS

1. Define insourcing.
2. List some of the pros and cons of using packaged EC applications.
3. Consider the differences between utility computing, ASP, and SaaS. Compare and contrast the benefits and drawbacks of each of these outsourcing approaches.
4. Consider and describe other development options.
5. List some reasons why it is difficult for applications running in different environments on different computers to communicate with one another.
6. Define Web Services.
7. What role does XML play in Web Services?
8. Describe the key technologies underlying Web Services.
9. What types of Web Services do Amazon.com and Google offer application developers?
10. What are some of the advantages of Web Services?
11. What are some of the factors limiting the adoption of Web Services?
12. Define service-oriented architecture (SOA) and list its advantages. Describe the relationship between SOA and Web Services EC.
13. Define virtualization and describe its relationship with SOA.
14. Discuss ways in which virtualization might assist a small manufacturing firm to offer an e-commerce application to its customers.

19.4 CRITERIA FOR SELECTING A DEVELOPMENT APPROACH

If a company decides to buy or lease an EC application, the following representative selection criteria need to be considered:

- ▶ **Flexibility.** Commercial packages need to be modified or adapted to the specific requirements of an application. Therefore, it is important to evaluate the extent to

which a package can be adapted and the willingness of the vendor to perform or support the adaptation.

- ▶ **Information requirements.** The selected package should satisfy the information requirements of the EC application under development. Information collection, storage, and retrieval capabilities and the database structure should be carefully examined.
- ▶ **User friendliness.** User friendliness is especially important for B2C, G2C, and some B2B sites. In these cases, if an application is difficult for the average visitor or customer to use, then it will have an immediate impact on its use and subsequently the bottom line.
- ▶ **Hardware and software resources.** The computer type (e.g., desktop, laptop, main-frame) and the operating system (e.g., Windows, LINUX, Mac) required by the package must be compatible with the existing platform. The CPU and storage requirements must also be compatible with existing specifications or easily accommodated.
- ▶ **Installation.** The installation effort required to implement the package is another important consideration. Some packages are complex and their installation requires extensive consultation. The installation process may also take a considerable amount of time and expertise.
- ▶ **Maintenance services.** Because EC application requirements are changing constantly, ongoing maintenance is required. It is important to consider how often the package needs to be updated and whether the vendor provides assistance for its maintenance.
- ▶ **Vendor quality and track record.** It is less risky to acquire an EC package from a vendor who has a good reputation and track record than from one with a less-than-stellar or unknown reputation. A vendor's quality can be indicated by its experience in the particular application and its sales and financial records, as well as its responsiveness to clients' requests. Vendor support may include online help and CRM programs, as well as PRM tools. To minimize risk, minor applications should be acquired first.
- ▶ **Estimating costs.** The costs of EC projects are usually difficult to assess and often underestimated. In addition to the obvious costs associated with EC development, it is also important to factor in the costs of installation, integration, customization, training, and maintenance.
- ▶ **Personnel.** Staffing requirements should be planned for in advance to ensure that the organization has the appropriate human resources for systems development (in the case of in-house development), implementation, operation, and maintenance. Currently, it is difficult to recruit and retain IT personnel with appropriate knowledge and experience in EC application development. Special expertise acquired from external consultants can be expensive.
- ▶ **Technological evolution.** Planning ahead for technological evolution facilitates the upgrade of EC applications and enables the organization to adopt innovations more quickly than the competition. It is, therefore, important to allow for flexibility in the application design so that the chosen options do not impose major limitations on future choices. Given the rapid pace of IT evolution, it is sometimes preferable to develop EC applications incrementally to take advantage of the latest developments in the technology.
- ▶ **Scaling.** System **scalability** refers to how big a system can grow in various dimensions to provide more service. Scalability can be measured in several ways, including the total number of users, the number of simultaneous users, and the transaction volume. These dimensions are not independent because scaling up the size of the system in one dimension can affect the other dimensions. The growth of scale is facilitated or constrained by the system architecture.
- ▶ **Sizing.** The required size and performance of an application are also difficult to predict because the growth of the user population of certain EC applications is hard to anticipate. Overloading the application decreases performance. For regular IT applications, deterioration in performance may affect productivity and user satisfaction; for EC applications, it could result in a major loss of business.
- ▶ **Performance.** System performance is a critical factor for business success, particularly when the system is used for EC. In addition to convenience, good performance also brings customers and competitive advantages. Performance is measured by two main metrics: latency and throughput. **Latency** measures the time required to complete an operation such as downloading a Web page. It is an indicator of the users' experience with

scalability

How big a system can grow in various dimensions to provide more service; measured by total number of users, number of simultaneous users, or transaction volume.

latency

The time required to complete an operation, such as downloading a Web page.

the system. **Throughput** measures the number of operations completed in a given period of time. It indicates the capacity or number of users that a system can handle. Throughput and latency are interrelated. An increase in either measure directly affects the other.

- ▶ **Reliability.** Reliability is an essential requirement for a successful system. System failures and downtime are costly. When an EC application fails, business is interrupted; at best the company loses sales; at worst it loses customers. System reliability can be enhanced through the use backup systems.
- ▶ **Security.** Security is critical to the adoption and diffusion of EC. Data and information flow in EC, as well as stored data, may include private and/or proprietary information. Thus, a selected package must meet strict security requirements. Systems, communication, and data security must be addressed early in the design of EC applications, *not* after their implementation. In addition to technological solutions such as firewalls and encryption, physical and procedural security measures must also be enforced.

For additional information on selection criteria, see Whitten and Bentley (2007) and the e-Security Toolkit at e-security-e-commerce-security.com/.

Section 19.4 ▶ REVIEW QUESTIONS

1. List some of the major criteria to consider when deciding whether to buy or lease an EC application.
2. Define latency.
3. Define throughput.
4. Discuss the business implications of having inadequate security on an e-commerce site.

19.5 E-COMMERCE SOFTWARE PACKAGES AND SUITES

Whether a company opts to purchase or lease its EC applications, two basic categories of software are available: functional packages, such as electronic catalogs, and merchant server software, such as EC suites.

FUNCTIONAL SOFTWARE PACKAGES

Standard functional software packages are available from a large number of vendors specifically for storefront construction (see Chapter 16). An example can be found at monstercommerce.com/shopping_cart_features_new.asp. This site offers dozens of software packages in the following areas: setting up your site, merchandising, inventory, payment options, hosting, shopping, tax, sales analysis, databases and systems, customer service, site design and layout, repeat customer accommodation, and security. Each includes a wide variety of features.

MonsterCommerce (monstercommerce.com) offers software both for small businesses and for large corporations (see demo at the site). The site also offers integrated solutions that are offered by several other vendors (e.g., storefront.com). An up-to-date list of vendors and the EC software packages they offer can be found at thesoftwarenetwork.com/eCommerce%2DSoftware.

Electronic catalogs are the virtual version of traditional product catalogs (see Chapter 2). Like its paper counterpart, an electronic catalog contains written descriptions and photos of products, along with information about various promotions, discounts, payment methods, and methods of delivery. Electronic catalogs are included in **merchant server software**, which includes features that make it simple and relatively inexpensive to develop (usually less than \$10,000). A catalog operation includes a straightforward pricing and product configuration.

Merchant Server Software

Merchant server software commonly includes the following features:

- ▶ Templates or wizards for creating a storefront and catalog pages with pictures describing products for sale
- ▶ Electronic shopping carts that enable consumers to gather items of interest until they are ready for checkout

throughput

The number of operations completed in a given period of time; indicates the number of users that a system can handle.

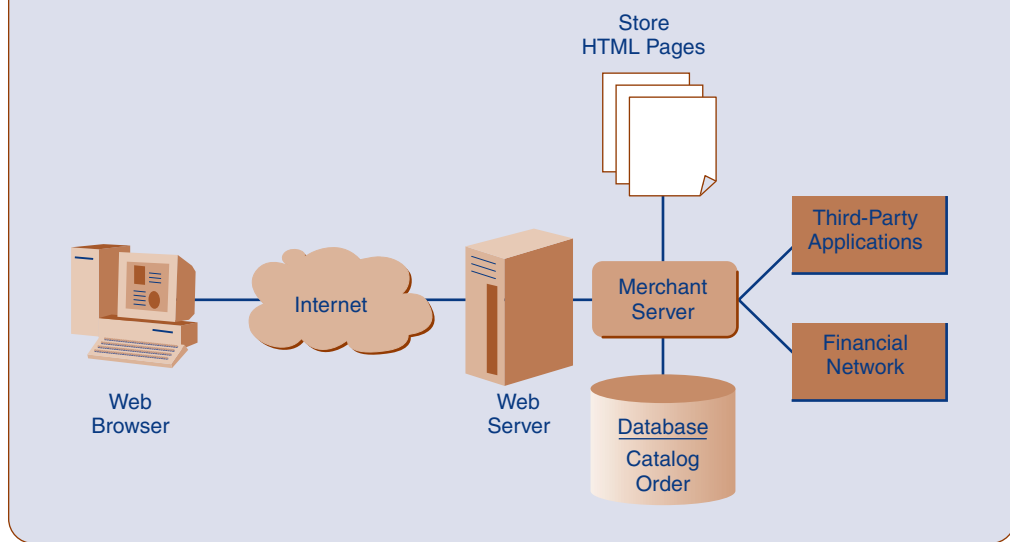
electronic catalog

The virtual-world equivalent of a traditional product catalog; contains product descriptions and photos, along with information about various promotions, discounts, payment methods, and methods of delivery.

merchant server software

Software for selling over the Internet that enables companies to establish selling sites relatively easily and inexpensively.

EXHIBIT 19.15 Merchant Server Architecture



- ▶ Web-based order forms for making secure purchases (either through SSL encryption or the SET protocol)
- ▶ A database for maintaining product descriptions, pricing, and customer orders
- ▶ Integration with third-party software for calculating taxes and shipping costs and for handling distribution and fulfillment

Exhibit 19.15 outlines the major components of merchant server software. As shown, a single server is used to handle product presentation, order processing, and payment processing (Treese and Stewart 2003). Likewise, a single database is used to store the catalog (i.e., product descriptions) and handle the details of customer orders. The pages of the electronic catalog are created dynamically from the product descriptions contained in the catalog database. For those merchants with only a few products for sale, it is not necessary to store the product descriptions in a database. Instead, the pages of the Web catalog can be created ahead of time.

EC SUITES

EC suite

A type of merchant server software that consists of an integrated collection of a large number of EC tools and components that work together for EC applications development.

An **EC suite** is a type of merchant server software that consists of an integrated collection of a large number of EC tools and components that work together for EC applications development. EC suites offer builders and users greater flexibility, specialization, customization, and integration in supporting complete front- and back-office functionality. In an EC suite, the functionality is distributed across a number of servers and databases instead of relying on a single server and database, as with less sophisticated merchant server systems. The elements displayed in Exhibit 19.16 are indicative of the components contained in a typical EC suite, the processes supported by an EC suite, and the back-end databases and operational systems utilized by the processes.

Over the past few years, the EC suite marketplace has experienced a substantial amount of consolidation. Among the major products that remain on the market are Microsoft's Commerce

EXHIBIT 19.16 Major Components of an EC Suite

Process Supported	Back-end databases	Operational Systems Utilized
Catalog Creation and Update	Catalog	
Customer Management	Customer	
Order Management	Order	Fulfillment Systems
Accounts Receivable	Payment	Financial Network

Server 2007 (microsoft.com/commerceserver/default.aspx), IBM's WebSphere Commerce suite (www-306.ibm.com/software/websphere/), and Oracle's EC applications (download-east.oracle.com/docs/cd/B10501_01/rac.920/a96598/ebizapps.htm). Up-and-coming EC suites include SalesPro (webbusconnect.com/serve.aspx?page=ec_ecomm_suite), OnLine Suites (onlinesuites.com), and AIT, Inc.'s E-Commerce Suite (ait.com/ecommercesuite.php).

Microsoft's Commerce Server 2007

Microsoft's Commerce Server 2007 (microsoft.com/commerceserver/default.aspx) offers a comprehensive framework for building tailored EC solutions. The framework consists of six main systems:

- ▶ **Product Catalog System.** Enables the creation, management, and syndication of customer-specific and location-specific catalogs with specialized pricing and sophisticated search capabilities.
- ▶ **Targeting System.** Creates and deploys multilingual merchandising and advertising campaigns that support advanced discounting, cross-selling, and customer profiling.
- ▶ **Profiling System.** Enables catalog personalization, pricing, business processing, merchandising, and advertising to specific needs of customers, suppliers, and partners.
- ▶ **Business Processing Pipelines System.** Enables firms to tailor order and merchandising processes to handle currency conversions, multiple shipments, and complex discounting.
- ▶ **Business Analytics System.** Offers businesses the ability to analyze, forecast, and mine the business data resulting from EC activities and processes, including clickstream usage, purchase histories, browsing behaviors, campaign effectiveness, and currency preferences, so that they can make informed decisions about the success of their online business.
- ▶ **User Management Console: Business Desk.** Enables business managers to respond quickly to changing customer and business needs through a centralized intuitive console for managing customer and partner profiles, personalization rules, product catalogs and pricing, merchandising and advertising campaigns, and business analysis.

Microsoft's Commerce Server comes in three editions—Standard, Developer, and Enterprise—which are designed to handle medium, large, and extremely high-traffic sites, respectively. All three editions are built on top of Microsoft's Windows operating system, SQL Server database, and the Visual Studio .NET development environment. All three editions also operate seamlessly with Microsoft's other .NET servers (e.g., Microsoft BizTalk Server and Microsoft's Content Management Server). The new version of Commerce Server, released in late 2006 (*Microsoft.com 2007a*), enables enterprises to accelerate and automate the delivery of online services and products to drive their revenue growth and reduce operational costs. An example of a business portal built using Microsoft's Commerce Server 2007 and BizTalk Server 2006 is described in Case 19.4.

IBM's WebSphere Commerce Suite

IBM's WebSphere Commerce suite (www-306.ibm.com/software/websphere/) is a comprehensive EC development platform designed to support B2C, B2B, or private exchange business models. The suite provides the following functions:

- ▶ Order management that optimizes movement of products through the supply chain
- ▶ Collaborative filters that enable an enterprise to better understand customers' buying patterns and preferences
- ▶ Portal capabilities that provide customers with personalized access to multiple commerce and noncommerce site applications
- ▶ Localization support that enables customized price, tax, and shipping calculations in the currency format and language dictated by the shopper's locale
- ▶ E-coupons that can be used by customers during online shopping
- ▶ Additional bundled products, including WebSphere Catalog Manager and WebSphere Payment Manager

IBM's suite is built on open industry standards such as Java, Java Servlets, JavaServer Pages (JSP), Enterprise JavaBeans (EJB), and XML. These standards make it easier to integrate new products with existing back-office transaction systems and databases.

CASE 19.4

EC Application

SEAL COMPANY EXPECTS \$5 MILLION REVENUE BOOST FROM INNOVATIVE INTERNET ORDERING SYSTEMS

Gulf Coast Seal distributes the O-rings, seals, and gaskets that keep oil and gas pipelines secure and leak proof. The company's business is built on lots of low-cost parts, some costing as little as 3 cents each. Some customers routinely place \$1 orders, with the average order size being just \$10. With orders bouncing between people, paper, faxes, and phones, Gulf Coast Seal calculated that its order-processing costs were between \$50 and \$75 per order, so these small orders were hurting the company's profitability.

The problem was the lack of electronic links between its customers' ERP systems and its own. A customer would create an order on its ERP and then phone or fax the order to Gulf Coast Seal. The company's small sales staff, who doubled as order entry and customer service staff, took the order, checked that the delivery date was possible, and cleared up any ambiguities. The salesperson then entered the order into Gulf Coast Seal's character-based ERP system. The whole process was laborious and required 3 to 5 days before an order hit Gulf Coast Seal's ERP system. The more orders, the bigger the backlog, and the louder customers grumbled. The company tried hiring more salespeople, but that only drove up operational and training costs.

To address these problems, the 130-person company in Houston, Texas, made a \$37,000 investment in an e-commerce system that it expects to yield up to \$5 million in additional revenue within 3 years by serving more customers and serving current customers better. Manual order processing had grown increasingly expensive as the company's order volume grew. Gulf Coast Seal wanted to introduce electronic efficiencies to drive costs down and increase customer satisfaction.

Using Microsoft Commerce Server 2007 and Microsoft BizTalk Server 2006, Gulf Coast Seal created a full-service e-ordering site that enables customers to place orders, check order status, and self-service every aspect of their account. Not only will the system enable Gulf Coast Seal to handle more customers, but to do so without increasing staff. Customers love the faster, easier ordering, and Gulf Coast Seal has reduced order-processing costs from as much as \$75 to just 3 cents per order. Today, Gulf Coast Seal customers can visit the company's Web site and sign in to a personalized site where they can place orders, view open invoices and orders, review their account history, and even look at Gulf Coast Seal's inventory to see whether the company has needed parts in stock. Customers can manage their own account profiles, including account IDs, passwords, and shipping addresses. They can view accounts receivable and transaction summaries, as well as view or print invoices.

Sources: Compiled from gulfcoastseal.com/ (accessed January 2007) and Microsoft (2006).

Questions

1. What business problems were addressed by Microsoft Commerce Server 2007 and Microsoft BizTalk Server 2006?
2. What are the business benefits provided by the Microsoft Commerce Server 2007 and Microsoft BizTalk Server 2006?
3. Log on to the Internet and explore other software options that Gulf Coast Seal might have considered and compare them with the Microsoft solution chosen.

IBM WebSphere delivers application infrastructure and integration software that helps companies address key priorities in an on-demand world. The suite offers a flexible operating environment that can easily adapt to support companies' efforts to drive business growth. It provides tools to help companies streamline and extend business processes to deliver the right information to the right people at the right time and enhance the productivity of their employees.

Oracle's EC Products

Oracle E-Business Suite Release 12 (oracle.com/applications/e-business-suite-release.html and oracle.com/applications/oracle-e-business-suite-release-12-brief.pdf) provides applications and technology to assist companies to compete more effectively in the worldwide marketplace. Oracle E-Business Suite Release 12 provides a vast array of applications aimed at supporting marketing, selling, and servicing of customers, suppliers, and partners online and enables businesses to think globally to make better decisions, work globally to be more competitive, and manage globally to lower costs and increase performance. With hundreds of cross-industry capabilities spanning enterprise resource planning, customer relationship management, and supply chain planning, this new release helps companies manage the complexities of global business environments. The following applications are aimed specifically at B2C and B2B operations:

- ▶ **Oracle iStore.** Enables merchants to build, deploy, manage, and personalize online storefronts. iStore is one of Oracle's key applications. It supports product catalog and content management, interactive and complex selling, personalized pricing, flexible check and payment options, account and contract management, and postsales order and shipping services. iStore integrates easily with Oracle's online marketing and eMerchandising functionality.
- ▶ **Oracle Marketing.** Provides automation and tools for the entire marketing process, ranging from initial marketing analysis to determine what and who should be targeted, campaign planning, budget and list maintenance, and multichannel execution, to campaign monitoring.
- ▶ **Oracle iPayment.** Offers risk management capabilities, transaction routing features, and a flexible payment architecture that supports every major online payment option.
- ▶ **Oracle Quoting.** Automates the creation and management of quotes for customized sales and service.
- ▶ **Oracle iSupport.** Provides customers with the ability to service and assist themselves over the Web.
- ▶ **Oracle Configurator.** Interactively captures, configures, and validates specialized manufacturing, sales, and service orders from customers, suppliers, and partners.

Like IBM's EC suite, Oracle's EC applications are built on open industry standards such as Java, Java Servlets, JSP, EJB, and XML. As one would expect, Oracle's applications rest on the market-leading Oracle database.

OTHER EC SUITES

The following are some additional EC suites that enable an organization to quickly build an e-business.

LiteCommerce 2.2

LiteCommerce 2.2 (litecommerce.com/products.html?designer) enables Web designers to create EC stores without having to program. Users can download templates to a workstation and then design their EC store in WYSIWYG HTML editors such as MS FrontPage and Macromedia Dreamweaver.

Wireless E-Com Suite

Wireless E-Com Suite (wm-us.com/e-commerce.htm) is designed for small- and medium-size Internet stores. With the Wireless E-Com Suite, a company can run one store with an unlimited number of categories and products. Features include customer care, product catalogs, product details, merchandising and inventory tools, security features, shipping and tax tools, sales analysis and tracking capabilities, and repeat-customer features.

ASPDotNetStoreFront

ASPDotNetStoreFront for Interprise Suite (IS) (aspdotnetstorefront.com/c-1-e-commerce-storefronts.aspx) provides a complete state-of-the-art e-commerce storefront powered by the Microsoft ASP.NET platform. Interprise Solutions and AspDotNetstorefront have partnered to create AspDotNetStorefront for Interprise Suite (IS), a powerful "Web to warehouse" solution that combines both e-commerce and back-office operations in one real-time solution.

ASPDotNetStorefront is fully integrated, so order, inventory and customer information is shared by both the e-commerce Web site and Interprise Suite. In addition to providing seamless interaction between the Web site and back-end operations, AspDotNetStorefront IS contains hundreds of industry-leading e-commerce features and supports search engine keyword integration and optimization.

ATG Commerce

ATG's (atg.com) comprehensive e-commerce product suite, rated number one by Forrester Research in 2006 (Mendelsohn 2006), enables customers to implement, monitor, and continuously improve personalized e-commerce applications. ATG helps keep customer interactions consistent across all contact channels and throughout the customer lifecycle. ATG Commerce is a

comprehensive, highly scalable solution to automate the complete lifecycle of online sales, marketing, and service. Its flexible, component-based architecture enables firms to personalize the online buying experience for their customers, making it easy for them to find desired products, comparison shop, register for gifts, preorder products, redeem coupons, and execute many other useful features.

Section 19.5 ■ REVIEW QUESTIONS

1. List the major features of an electronic catalog.
2. Describe the basic business systems in Microsoft's Commerce Server 2007.
3. Describe the functions supported by IBM's WebSphere Commerce suite.
4. Describe the key EC applications provided by Oracle for building B2C and B2B sites.
5. As the owner of a small- to medium-sized online business, which of the e-commerce suites described in this section would you find most useful? Explain.

19.6 CONNECTING TO DATABASES AND OTHER ENTERPRISE SYSTEMS

The major integration areas discussed in this chapter are connecting to databases and to back-end systems.

CONNECTING TO DATABASES

Virtually every EC application requires database access. For example, when a customer orders a product online, the product description, inventory count, and order information are likely to be retrieved from and stored in one or more databases (see Exhibit 19.17). An EC application can be connected to a back-end database in a variety of ways. Today, most of these connections are accomplished via a **multitiered application architecture**, such as the one depicted in Exhibit 19.17. This architecture has four tiers:

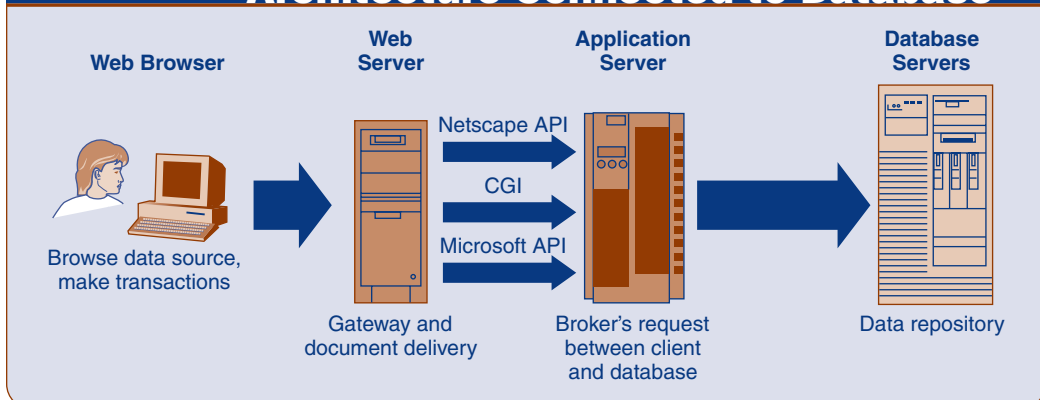
1. A Web browser that presents data and information to and collects data from the end user
2. A Web server that delivers Web pages, collects the data sent by the end user, and passes data to and from the application server
3. An application server that executes business rules (e.g., user authorization), formulates database queries based on the data passed by the Web server, sends the queries to the back-end database, manipulates and formats the data resulting from the database query, and sends the formatted response to the Web server
4. A database server in which the data are stored and managed and database requests are processed

This separation of functions makes it easier to change any tier (or layer) without impacting the other layers. Thus, an application server can be designed to interface or communicate with a wide variety of databases and database management systems (e.g., Oracle, MS SQL Server, DB2).

multitiered application architecture

EC architecture consisting of four tiers: Web browsers, Web servers, application servers, and database servers.

EXHIBIT 19.17 Example of Multitiered Application Architecture Connected to Database



Source: Systinet. *Introduction to Web Services Architecture*. Burlington, MA: Systinet Corp., 2002, p. 6.

In some cases, the data being accessed are stored in an existing (legacy) database (e.g., inventory or order databases). In these cases, it is better to tie the application server directly to the legacy database rather than duplicating the data in a database established solely for the EC application. This approach ensures that the data are up-to-date, that they are consistent across the applications accessing the data, that a minimum of storage space is used, and that there is only one database to create and maintain rather than two.

INTEGRATING EC APPLICATIONS AND BACK-END SYSTEMS

Several technologies can be used to integrate an EC application with a back-end database. Many of the commercial electronic catalogs and EC suites have built-in integration capabilities. If a company wants to build its own database interface, a couple of options are available. First, all of the Web scripting languages (e.g., PHP, JSP, and Active Server Pages [ASP]) have commands that simplify the process. More specifically, these scripting languages enable a programmer to build Web pages that can issue queries to a back-end (relational) database and process the database's response to the query. Second, a number of specialized application servers are available that simplify the task of integrating an EC application with one or more back-end databases. Among these specialized servers, BEA Inc.'s WebLogic Server 9.2 (bea.com) is a market leader.

In addition to connecting to back-end databases, most EC applications also require integration with a variety of other systems—ERP, CRM, SCM, EDI, data warehouses, and other important internal systems—both inside and outside the company. Again, electronic catalogs and EC suites usually have built-in modules for integration with these systems. The integration can also be handled with a class of software called **enterprise application integration (EAI)**. These products focus on the integration of large systems (see Schneider 2004). TIBCO (tibco.com), webMethods (webmethods.com), and IBM's WebSphere InterChange Server (ibm.com) are examples of companies that have offerings in the EAI arena.

enterprise application integration (EAI)
Class of software that integrates large systems.

MIDDLEWARE

As discussed earlier, companies and organizations are now building enterprise-wide EC systems by integrating previously independent applications together with new developments (see Coffee 2004). EC applications also must be connected to items such as the partners' systems or to public exchanges. Such connections are referred to as *integration*. EC users interact with Internet applications through a variety of devices whose characteristics and performance figures span an increasingly wide range. Applications use communication protocols and intermediate software that resides on top of the operating systems to perform the following functions:

- ▶ Hiding distribution (i.e., the fact that an application is usually made up of many interconnected parts running in distributed locations)
- ▶ Hiding the heterogeneity of the various hardware components, operating systems, and communication protocols
- ▶ Providing uniform, standard, high-level interfaces to the application developers and integrators so that applications can be easily composed, reused, ported, and made to interoperate
- ▶ Supplying a set of common services to perform various general-purpose functions to avoid duplicating efforts and to facilitate collaboration between applications

The intermediate software layers have come to be known under the generic name of **middleware**. Middleware is essentially a separate program that provides an interface between diverse client and server systems. Its main function is to mediate interaction between the parts of an application or between applications. For more information, see middleware.objectweb.org.

IBM is the leading provider of middleware. The company offers a number of on-demand solutions for communication, government, retail, banking, financial markets, and many other industries. IBM Middleware (ibm.com/middleware) helps automate systems, integrate operations, connect people, and develop software.

middleware
Separate products that serve as the glue between two applications; sometimes called *plumbing* because it connects two sides of an application and passes data between them.

Section 19.6 ▶ REVIEW QUESTIONS

1. Describe the basic elements of a multitiered application architecture.
2. List the ways in which an EC application can be connected to back-end databases and other transaction processing systems.
3. Define *middleware* and describe its attributes.

19.7 VENDOR AND SOFTWARE SELECTION

Few organizations, especially SMEs, have the time, financial resources, or technical expertise required to develop today's complex e-business systems. This means that most EC applications are built with hardware, software, hosting services, and development expertise provided by outside vendors. Thus, a major aspect of developing an EC application revolves around the selection and management of these vendors and their software offerings. Martin et al. (2002) identified six steps in selecting a software vendor and a package, as illustrated in Exhibit 19.18.

STEP 1: IDENTIFY POTENTIAL VENDORS

Potential vendors can be identified from software catalogs, lists provided by hardware vendors, technical and trade journals, consultants experienced in the application area, peers in other companies, and Web searches. These sources often yield so many vendors and packages that one must use some preliminary evaluation criteria to eliminate all but a few of the most promising ones from further consideration. For example, one can eliminate vendors that are too small or that have no track record or that have a questionable reputation. Also, packages may be eliminated if they do not have the required features or will not work with available hardware or the operating system, communications network, or database management software.

STEP 2: DETERMINE THE EVALUATION CRITERIA

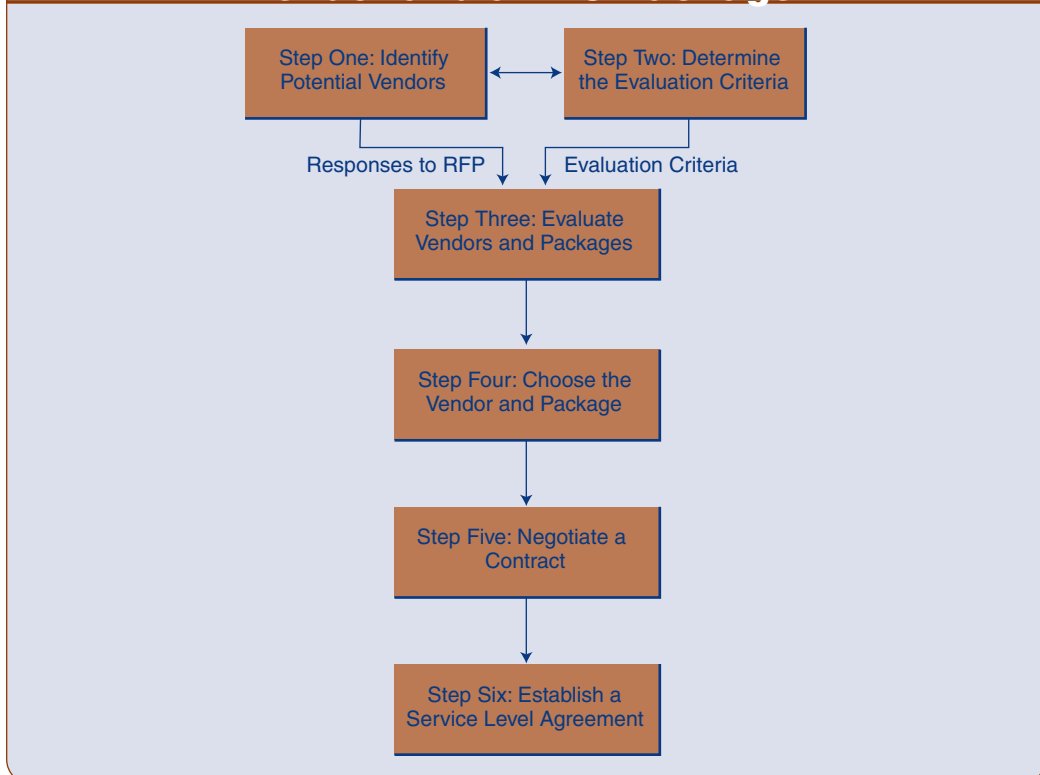
The most difficult and crucial task in evaluating a vendor and a packaged system is to determine a weighted set of detailed criteria for choosing the best vendor and package. Some areas in which detailed criteria should be developed are vendor characteristics, functional requirements of the system, technical requirements the software must satisfy, the amount and quality of documentation provided, and vendor support of the package.

These criteria should be documented in a **request for proposal (RFP)**, which is sent to potential vendors to invite them to submit a proposal describing their software package

request for proposal (RFP)

Notice sent to potential vendors inviting them to submit a proposal describing their software package and how it would meet the company's needs.

EXHIBIT 19.18 The Process of Selecting a Software Vendor and an EC Package



and how it would meet the company's needs. The RFP provides the vendors with information about the objectives and requirements of the system, the environment in which the system will be used, the general criteria that will be used to evaluate the proposals, and the conditions for submitting proposals. It may also request a list of current users of the package who may be contacted, describe in detail the form of response that is desired, and require that the package be demonstrated at the company's facilities using specified inputs and data files.

STEP 3: EVALUATE VENDORS AND PACKAGES

Vendor responses to an RFP generate massive volumes of information that must be evaluated to determine the gaps between the company's needs (as specified by the requirements) and the capabilities of the vendors and their application packages. Often, the vendors and packages are given an overall score by assigning an importance weight to each of the criteria, ranking the vendors on each of the weighted criteria (say 1 to 10) and then multiplying the ranks by the associated weights. A short list of potential suppliers can be chosen from those vendors and packages with the highest overall scores.

STEP 4: CHOOSE THE VENDOR AND PACKAGE

Once a short list has been prepared, negotiations can begin with vendors to determine how their packages might be modified to remove any discrepancies with the company's desired EC application. Thus, one of the most important factors in the decision is the additional development effort that may be required to tailor the system to the company's needs or at least to integrate it into the company's environment. Additionally, the opinions of the users who will work with the system and the IT personnel who will have to support the system must be considered.

STEP 5: NEGOTIATE A CONTRACT

The contract with the software vendor is very important. Not only does it specify the price of the software, but it also determines the type and amount of support to be provided by the vendor. The contract will be the only recourse if the system or the vendor does not perform as specified. Furthermore, if the vendor is modifying the software to tailor it to the company's needs, the contract must include detailed specifications (essentially the requirements) of the modifications. Also, the contract should describe in detail the acceptance tests the software package must pass.

Contracts are legal documents, and they can be quite tricky. Experienced contract negotiators and legal assistance may be needed. Many organizations have software purchasing specialists who assist in negotiations and write or approve the contract. They should be involved in the selection process from the start. If an RFP is used, these purchasing specialists may be very helpful in determining its form and in providing boilerplate sections of the RFP.

STEP 6: ESTABLISH A SERVICE LEVEL AGREEMENT

Service level agreements (SLAs) are formal agreements regarding the division of work between a company and its vendors. Such divisions are based on a set of agreed-upon milestones, quality checks, "what-if" situations, how checks will be made, and what is to be done in case of disputes. If the vendor is to meet its objectives of installing EC applications, it must develop and deliver support services to meet these objectives. An effective approach to managing SLAs must achieve both facilitation and coordination. SLAs do this by (1) defining the partners' responsibilities, (2) providing a framework for designing support services, and (3) allowing the company to retain as much control as possible over their own systems.

service-level agreement (SLA)

A formal agreement regarding the division of work between a company and a vendor.

Section 19.7 ▸ REVIEW QUESTIONS

1. List the major steps in selecting an EC application vendor and package.
2. Describe a request for proposal (RFP).
3. Describe a service level agreement (SLA).

19.8 USAGE ANALYSIS AND SITE MANAGEMENT

To improve EC Web sites, it is advisable to monitor what customers are doing there (usage analysis). Both B2C and B2B Web sites require a thorough understanding of the usage and patterns—the who, what, where, when, and how. This can be done by analyzing Web log files.

LOG FILES

Every time a user accesses a Web server, the server logs the transaction in a special access log file. **Access logs** are text files in which each line of the file details an individual access. Regardless of the type of Web server, access logs use a common log file format. This makes them easy to analyze and compare. Because log files can become quite voluminous, it is hard to analyze the accesses by hand. For this reason, most Web server EC software vendors provide free software for analyzing access log files. Commercial products that provide more sophisticated log analyses also are available (e.g., NetIQ's WebTrends). For an example, see Application Case 19.5.

Access logs provide a variety of statistics that can be used for analyzing and improving marketing and advertising strategies. Among the more valuable statistics are:

- ▶ **Pageviews by time slot.** Pageview statistics allow frequent review of the number of site accesses. Grouping pageviews by “time bucket” (time slot) also enables a company to ascertain the time slots, such as morning, afternoon, or evening, during which customers visit the site.
- ▶ **Pageviews by customers' log-in status.** This information helps determine whether requiring customers to log in is worthwhile. For instance, if the number of pageviews of customers who log in is substantially greater than those who do not, the company may find the login requirement effective and worthwhile.

access log

A record kept by a Web server that shows when a user accesses the server; kept in a common log file format, each line of this text file details an individual access.

CASE 19.5

EC Application

PERSONAL CREATIONS BOOSTS ONLINE SALES BY 31 PERCENT WITH A REVAMPED WEB SITE

Founded in 1989, Personal Creations is a top provider of personalized gifts in the United States. It created its online catalog (*personalcreations.com*) in 1996, and the Web site was extended to an integrated order processing system in 1999. Between 1999 and 2002, traffic to *personalcreations.com* increased to a point where technical problems were hindering Web site performance. The company required a more robust and scalable Web site that would better meet its business objectives by encouraging more visitors to click through the site and helping more shoppers to complete their purchases.

To redesign its new site, Personal Creations adopted the Fry Flagship framework, a quick-to-market EC solution from the WebTrends Insight Network partner, Fry Inc. By analyzing how visitors were navigating through the old site, Personal Creations and Fry developed a new site that led visitors more easily to the information and products they were looking for.

The revamped Web site highlights the personalization image of Personal Creations and gives customers ideas for how to use products (such as “Cherish Your Memories” and “Start a Family Tradition”) instead of just promoting specific products. Its homepage fits the screen without requiring shoppers to scroll down or across. The new Web site includes a breadcrumb trail indicating where a visitor has been and some “top picks” pages so visitors can see the most popular

products for any given product category or occasion. Product browsing is now easier, and visitors are more likely to click through the site. Personal Creations also set up a system for cross-selling products on the new site. The system is based on cross-sale recommendations by way of product affinity tables created with WebTrends analysis of customer purchasing behavior.

The combined effect of the changes, including the improved homepage, navigation, and checkout, has been a 31 percent increase in online sales for Personal Creations since the launch of the revamped Web site in October 2002. Currently, the Web site is a critical business channel, accounting for nearly 55 percent of the company's overall revenue.

Sources: Compiled from Webtrends.com (2004) and NetIQ.com (2004).

Questions

1. What are the technical and business requirements of the new Personal Creations Web site?
2. Why is an in-depth analysis of Web site activity and Web visitor behavior critical to the design of a Web site?

- ▶ **Pageviews by referrers.** Some customers are drawn or referred to the site by clicking on banners or links on other Web sites. Knowing the source of such referrers is useful for assessing the effectiveness of the location of banners, and customers' interests can also be determined from the nature of the Web site with those banners.
- ▶ **Pageviews by visitor's hardware platform, operating system, browser, and/or browser version.** These types of pageviews enable a company to obtain information on the hardware platform (e.g., Mac or PC) and browser type (e.g., Internet Explorer or Netscape) used by the viewer.
- ▶ **Pageviews by visitor's host.** This type of pageview provides information on the customers' host sites. Knowing where customers are coming from can enable the company to target potential customers via popular hosts such as Yahoo!

Some of the marketing and business questions to which these statistics can be applied are listed in Exhibit 19.19.

E-COMMERCE MANAGEMENT TOOLS

Managing the performance of a Web site is a time-consuming and tedious administrative task. Several vendors offer suites of products or individual packages that can assist with the management process. One of the more prominent vendors of IT and Web management tools is BMC Software, Inc. (bmc.com). Some of the products offered by BMC include:

- ▶ **Patrol for E-Business Management.** This package includes Patrol for Internet Services, which measures Web response time; Patrol for Firewalls, which provides firewall administration; and Patrol for Microsoft or Netscape application servers.
- ▶ **MainView for E-Business Management.** This package includes MainView for WebSphere, which aids in the management of mainframe-based EC applications; MainView for Network Management, to monitor mainframe network connections; and MainView for Systems Management, to provide systems administration.

Other EC management tools include site version control tools, combined utilities/tools, server management and optimization tools, and authoring/publishing/deployment tools that include significant site management or testing capabilities. A detailed list of vendors is provided by Hower (2005). The following are some examples:

- ▶ **COAST WebCentral.** This site management tool from Coast Software, Inc. (e-tamed.co.uk/products/coast4.html) monitors compliance standards related to privacy, information assurance, accessibility, and Web governance. Capabilities include high-level summary reports, site scores, and trend analyses using intuitive browser-based reports; roll-up of results from different business units or global Web sites for overall views of site compliance; complete inventory of Web sites; and comparison inventory reports to compare different versions of a Web site.
- ▶ **Maxamine Knowledge Platform.** This integrated site management tool from Maxamine Inc. (maxamine.com) combines site content, structure, and visitor traffic data into one searchable database, enabling detailed analysis of what is going on behind the scenes of Web operations.
- ▶ **WebCEO.** Companies can use this tool from Radiocom Ltd. (webceo.com) to maintain, promote, and analyze Web sites. It includes a link checker, a WYSIWYG editor, and FTP/publishing, traffic analysis, and site monitoring capabilities.
- ▶ **ManageEngine Applications Manager.** This site management tool from AdventNet (adventnet.com) works with a variety of Web servers, database servers, service types, and OSs.

Section 19.8 ▶ REVIEW QUESTIONS

1. Define *access log files*.
2. List some of the types of statistics provided by an access log.
3. Describe some of the uses of an access log.
4. Describe EC site management activities and list some tasks.

EXHIBIT 19.19 Areas of Usage Analysis and Sample Business Questions for Online Stores

Area of Analysis	Business Questions
Overall store performance	<ul style="list-style-type: none"> • What is the sales value for a specific period of time, say, 1 week? • What is the number of customer visits for the day? • What is the store conversion rate for the week? • What is the sales value index for the week?
Advertising	<ul style="list-style-type: none"> • Which banner ads are pulling in the most traffic? • How many sales are driven by each banner ad? • What products do shoppers select from a particular banner? • What is the conversion rate for each banner ad?
External referrals (from others to your site)	<ul style="list-style-type: none"> • Which portal sites are pulling in the most traffic? • Which are generating the most sales? • How many sales are generated by each referral site/search engine? • What products do shoppers from a particular portal site purchase?
Shopper segmentation	<ul style="list-style-type: none"> • How many visitors are from a specific domain? • What is the distribution of first-time vs. repeat shoppers? • What characterizes shoppers of a particular set of products? • What characterizes shoppers who abandon shopping baskets?
Product grouping	<ul style="list-style-type: none"> • How much do cross-sells/up-sells contribute to gross revenue? • What are the best performing cross-sell pairs? Worst? • What is the overall conversion rate for cross-sells/up-sells?
Promotions and recommendations	<ul style="list-style-type: none"> • How much do promotions contribute to gross revenue? • Which promotions are generating the most sales? • What is the overall conversion rate for promotions? • What is the overall conversion rate for recommendations? • At what levels in the site hierarchy are the best promotions located?
Shopping metaphor	<ul style="list-style-type: none"> • What generates the most sales value: searching or browsing? • How much does searching contribute to gross revenue? • What is the conversion rate for searching?
Design features	<ul style="list-style-type: none"> • What are the features of links customers most frequently click? • What are the features of links from which customers most frequently buy? • What parts of pages do customers most frequently buy from? • Do products sell better in the upper-left corner?
Product assortment	<ul style="list-style-type: none"> • What are the top sellers for the week? • What is the conversion rate for a particular department? • How is a product purchased: purchase frequency and quantity? • What characterizes the products that end up being abandoned? • How much of the sales of each product are driven by searching?

MANAGERIAL ISSUES

A number of managerial issues relate to the concepts presented in this chapter. Some of these are as follows:

- 1. What is our business perspective?** When one thinks of the Web, one immediately thinks of the technology. But some of the most successful sites on the Web rely on basic technologies—freeware Web servers, simple Web page design, and limited bells and whistles. What makes these sites successful is not the technology but their owners' understanding of how to meet the needs of their online customers.
- 2. Do we have a systematic development plan?** The cost of developing and maintaining even a small EC site can be substantial. To ensure success, development and maintenance issues need to be approached

systematically, just like any other IT development project. Within this plan, the specification of the EC architecture is crucial. If the architecture is wrong, the entire project is at risk.

3. **Insource or outsource?** Many large-scale enterprises are capable of building and running their own EC Web sites. However, EC Web sites may involve complex integration, security, and performance issues. For those companies venturing into the EC arena, a key issue is whether the site should be built in-house (insourced), thus providing more direct control, or outsourced to a more experienced provider. Outsourcing services, which enable companies to start small and evolve to full-featured functions, are available through many ASPs, ISPs, telecommunication companies, Internet malls, and software vendors that offer merchant server and EC applications.
4. **How should Web Services be deployed?** Many organizations face problems of integrating systems, applications, and data. The IT costs associated with getting applications and databases running on different hardware and operating systems to communicate with one another are substantial. Web Services offer the means to solve this interoperability problem in an open, straightforward, and efficient fashion without the need for new hardware or application reprogramming. The key to deploying Web Services is to create a stepped plan for their introduction, concentrating on those systems where integration needs are greatest.
5. **How should we choose a vendor/software?** Because most EC applications are built from packaged applications and components or outsourced to a third party, the success of the EC applications rests on choosing the best vendor and package. Like any other part of

the development process, a detailed list of selection criteria is needed for the selection process.

6. **Have we analyzed the data?** All EC sites provide the means to gather data about system usage. These data should be analyzed frequently to modify and redesign an existing site to better meet the needs of current and prospective customers and users. This analysis can also be used to personalize the experience of these same users.
7. **Should users be involved?** The direct and indirect users of an EC system are likely to be the most knowledgeable individuals concerning requirements and which alternatives will be the most effective. Users are also the most affected by a new information system. IS analysts and designers, however, are likely to be the most knowledgeable individuals concerning technical and data management issues. These professionals, too, are likely the most experienced in arriving at viable EC systems solutions. The right mixture of user involvement and information systems expertise is crucial.
8. **How should we manage development risks?** The development of EC applications involves risk. Systems may not be completed, completed too late, or require more resources than planned. The risk is large in enterprise systems. For information on how to manage such risk, see Scott and Vessey (2002).
9. **How shall we plan for service-oriented architecture (SOA)?** With SOA, an organization creates EC services components once and reuses these assets to drive better cost-efficiency from its IT investments. It is important that we define an enterprise approach to SOA and align IT with real business requirements. Also, Web Services may be acquired from other sources, either public or private.

RESEARCH TOPICS

Here are some suggested topics related to this chapter. For details, references, and additional topics, refer to the Online Appendix A “Current EC Research.”



1. Building an E-Commerce Application

- ▶ Compare software designed to assist in building an E-commerce application
- ▶ Framework for E-Commerce development
- ▶ Barriers to developing E-Commerce applications
- ▶ Cultural issues in developing a global E-Commerce application

2. Outsourcing

- ▶ Compare various outsourcing options
- ▶ Explore emerging new outsourcing options
- ▶ Pricing outsourcing options

3. Web Services

- ▶ Emerging software to support Web services
- ▶ New opportunities for Web service applications

- ▶ Effects on various industry sectors of Web service availability
 - ▶ Effects of entry of large software vendors into Web service arena
 - ▶ Pricing Web services
 - ▶ Services management issues
 - ▶ Case studies of Web service applications
- 4. **Technical Aspects of Building an E-Commerce Application**
 - ▶ Compare hardware alternatives for e-commerce applications
 - ▶ Usage analysis
 - ▶ Determining costs and benefits of different hardware configurations

SUMMARY

In this chapter, you learned about the following EC issues as they relate to the learning objectives.

1. **The major steps in developing an EC application.** Because of their cost and complexity, EC sites need to be developed in a systematic fashion. The development of an EC site should proceed in steps. First, an EC application portfolio is defined based on an organization's strategy. Second, the EC architecture is created. Next, a decision is made whether to build, buy, or outsource the development. Third, the system is installed, tested, and deployed. Finally, the system goes into maintenance mode, with continual changes being made to ensure the system's continuing success.
2. **The major EC applications and their major functionalities.** Every type of EC application has a long list of functional requirements. Fortunately, most of these requirements can be met by packaged applications. Online storefronts can be developed with the aid of electronic catalog or merchant server software. Similarly, B2C, B2B, and exchange applications of all sorts can be constructed from components that have the listed functionalities.
3. **The major EC application development options, along with their benefits and limitations.** EC sites and applications are rarely built from scratch. Instead, enterprises buy a packaged EC suite and customize it to suit their needs, or they outsource the development to a third party. A new generation of Web tools is taking the programmer out of the development process and empowering more users to develop their own Web sites. The selection of one option over another should be based on a systematic comparison of a detailed list of requirements that examines important considerations such as flexibility, information needs, user friendliness, hardware, and software resources.
4. **EC application outsourcing options.** Many enterprises elect to outsource the development and maintenance of their EC sites and applications. The most common type of EC applications outsourcing is the use of application service providers (ASPs). Utility computing is an emerging option. An enterprise can rely on an existing e-marketplace or exchange. An online storefront can be hosted by an Internet mall. Or an enterprise could enter into a joint development agreement with a venture partner or a consortium. Again, the choice depends on the functional requirements of the EC site or application, the costs involved, the time frame, and the available IT resources.
5. **The major components of software packages and EC application suites.** An online storefront has the same requirements as a brick-and-mortar storefront. Simple sites can be built from packaged electronic catalog or merchant server software. More complex online storefronts and other types of EC sites (e.g., B2B, exchanges, etc.) can be built from comprehensive EC suites such as Microsoft's Commerce Server or IBM's WebSphere Commerce suite. A payment gate and a site search engine are useful.
6. **Methods for connecting an EC application to back-end systems and databases.** Virtually every EC application requires access to back-end relational databases and other transaction systems (e.g., ERP, SCM, CRM, etc.). Integration can be accomplished in a variety of ways, including using integration modules supplied with electronic catalog or EC suite packages, customizing the integration with a Web scripting language (e.g. PHP, Active Server Pages [ASP], or JSP), employing specialized application servers, employing a full-blown EAI tool, or using XML-based technologies.
7. **The rise of Web Services and XML.** Web Services 2.0 is the newest technology aimed at solving the integration and interoperability problems (getting applications running in different computer environments to communicate with one another). Web Services generally rely on open standards, including XML, SOAP, WSDL, and UDDI, to overcome the problems. WSDL defines the operations that a Web Service can perform. To invoke a Web Service, an application sends a SOAP or XML message to the service, which in turn responds with a SOAP/XML message. Companies can publish their Web Services in a UDDI registry, so that any application can take advantage of the operations the services perform. Although Web Services require minimal changes and reprogramming of existing systems and applications, they still require advanced programming skills to implement and deploy them. Toward this

end, hardware and software vendors such as Microsoft and IBM have provided software development environments to ease the task.

8. **Understand service-oriented architecture and its relationship to EC.** With service-oriented architecture (SOA) and Web Services, functions within EC applications such as order taking and billing can be automatically invoked and executed anywhere in the world based on business rules. Developers can swap out one EC service and replace it with another service without having to worry about the inner workings of the two services. SOA is expected to transform the way we build EC systems in the future from slow, error-prone manual coding to an automated discipline.
9. **Criteria used in selecting an outsourcing vendor and package.** A systematic process should be used in selecting a third-party tool or an outsourcing company. Among the key steps in making the selection are: (1) identifying potential vendors and packages, (2) detailing the evaluation criteria, (3) using the criteria to produce a short list of

possible vendors, (4) choosing a candidate from the short list, (5) negotiating the deal and modifications needed to meet overall application needs, and (6) establishing an SLA to define who is responsible for specific aspects of the development and maintenance and what quality metrics will be used for the services to be rendered.

10. **The value and uses of EC application log files.** Most EC applications produce log files of detailed system usage. The data in these files can be analyzed with an eye toward modifying the application's content and flow. In this way, the application can be better aligned with the enterprise's marketing and advertising strategies. In the same vein, the application can be adjusted to meet users' needs.
11. **The importance of usage analysis and site management.** Usage analysis is about monitoring customer activities at EC Web sites. Web log files provide a variety of statistics about Web site usage patterns that can be useful for analyzing and improving marketing and business strategies.

KEY TERMS

Acceptance testing	10	Native virtualization	29	Simulation	29
Access logs	42	Outsourcing	12	Social network	22
Ajax	22	Policy-based resource management tools	14	Software as a service (SaaS)	15
Application service provider (ASP)	14	Policy-based service-level-management tools	14	Throughput	33
EC architecture	7	Really Simple Syndication (RSS)	22	Turnkey approach	12
EC Suite	34	Representational State Transfer (REST)	27	Unit testing	10
Electronic catalog	33	Request for proposal (RFP)	40	Universal Description, Discovery, and Integration (UDDI)	18
Emulation	29	Reusability	11	Usability testing	10
Enterprise application integration (EAI)	39	Scalability	32	Utility (on-demand) computing	13
Insourcing	11	Security protocols	18	Viral video	22
Integration testing	10	Service level agreement (SLA)	41	Virtualization	27
Interoperability	11	Service-oriented architecture (SOA)	25	Web-oriented architecture (WOA)	27
Latency	32	Simple Object Access Protocol (SOAP)	18	Web Service	17
Mashup	22			Web Services Description Language (WSDL)	18
Merchant server software	33			Web 2.0	22
Middleware	39				
Multitiered application architecture	38				

QUESTIONS FOR DISCUSSION

1. Discuss the advantages of leasing an application over purchasing one.
2. A large company with a number of products wants to start selling on the Web. Should it use a merchant server or an EC application suite? Assuming it elects to use an EC application suite, how would you determine whether the company should outsource the site or run it in-house?
3. A large chemical manufacturing company is interested in starting an online exchange. What are some of the ways it could achieve this goal?

4. A firm decides to make its EC Web site more dynamic by tying its application to a back-end database. What are some of the ways in which the firm could accomplish this task?
5. An online vendor wants to hook its shopping cart application to a credit card authorization Web site. How could this be done with Web Services? How would the authorization site advertise its services for other sites to use?
6. An enterprise wants to modify its EC site so that it conforms more closely with the company's overall business strategies. What sorts of online data are available for this purpose? What types of business strategy questions can be addressed by these data?
7. Define Web Services. List the major areas of Web Services applications. Discuss three major advantages and three major disadvantages of Web Services.
8. In what ways do you think a Web site's log files may violate consumers' privacy?
9. You have decided to use a third-party application to develop and deploy a sell-side B2B site. Create a checklist for determining which third-party EC application products will best meet your application requirements.
10. Identify the major reasons why utility computing tools may become the next big thing in EC applications development.
11. Do you think the technology has a lot of catching up to do before utility computing will become a reality for companies with many computing needs?
12. Discuss the relationship between service-oriented architecture (SOA) and Web Services.
13. Discuss the relationship between Web 1.0 and Web 2.0. Consider the differences in delivery of EC applications under the two methods of delivery.

INTERNET EXERCISES

1. Access the Choice Mall Web site (choicemall.com). Visit some of the online stores in the mall. What are the functionalities of the mall? What are some of the benefits of the online mall to the participating vendors? To shoppers? Do you think a shopper is better off using an online mall or using a search engine such as AltaVista to locate a store providing a product of interest? In what ways could Choice Mall improve the chances that buyers will make return visits?
2. Visit a large online storefront of your choice. What functions does it provide to shoppers? In what ways does it make shopping easy? In what ways does it make shopping more enjoyable? What support services does it provide?
3. Go to the WebTrends site (webtrends.com). What types of information does its Analysis Suite provide? How can this information be used to improve a Web site? What types of tracking information are not provided by this suite? (See Internet Exercise #1.)
4. Visit IBM's site (ibm.com). Find its WebSphere product. Read some recent customer success stories. What makes this software so popular?
5. Go to the World Wide Web Consortium's discussion of Web Services architecture (w3.org/tr/ws-arch). Based on this discussion, what role does a UDDI play in Web Services? If a company wants to publicize its Web Services, what means are available?
6. Visit Sun Microsystems (sun.com). What type of development platform does Sun provide for creating and deploying Web Services? What are the capabilities and benefits of the platform?
7. Visit the Microsoft Web site (microsoft.com). Find its BizTalk product. What kind of software is this? What role could it play in an EC application?
8. Enter webservices.org and find material on Web Services and EC. Prepare a report.
9. Enter monstercommerce.com and ecommerce-shopping-cart.biz and write a report comparing their products.
10. Search the Internet to find a mashup site that is not mentioned in the text. Determine the components that it has "mashed" and write a report on the value added that the mashup provides.



TEAM ASSIGNMENTS AND ROLE PLAYING

1. Select a series of Web sites that cater to the same type of buyer (e.g., several Web sites that offer CDs or computer hardware). Divide the sites among several teams and ask each team to prepare an analysis of the different sorts of functions provided by the sites, along with a comparison of the strong and weak points of each site from the buyer's perspective.
2. Several vendors offer products for creating online stores. The Web sites of these vendors usually list those online stores that currently use their software (customer success stories). Assign each team a number of vendors. Each team should prepare reports comparing the similarities and differences among the vendors' sites and evaluating the customers' success stories. Do the customers take advantage of the functionality provided by the various products?
3. As a team, explore the desired capabilities of various EC applications (B2B, B2C, auctions, portals, G2C, etc.). Look at the capabilities of these applications and at their functionalities, and then compare the two (see Section 19.2 for a list of functionalities). If the functionalities of the applications are not sufficient, explain what additional functionalities are needed.
4. Amazon.com provides Web Services for its Associates (amazon.com/webservices), as well as other product sellers and vendors. Assign one team to explore the services for Associates and another to find out about the services for sellers and vendors. Describe the services provided by each. What are the benefits of these services? What companies are currently using these services? Go to their sites and describe how they are using these services.
5. Each team selects a software vendor that supports integration (e.g., see oracle.com, intersystem.com, icode.com, tibco.com, microsoft.com, or webmethods.com). Prepare a report on how they connect EC applications (such as ordering) to back-office systems.
6. Each team brainstorms a new and innovative mashup site. Discuss and document the value it would provide and define the market to which the site would appeal. Present (sell) your site to the other teams and take a vote on the best mashup created by all teams.

Real-World Case

MOLDING A NEW VISION FOR E-COMMERCE AT D-M-E

D-M-E Company may not be a household name, but for moldmakers, mold designers, and molders who make parts for the plastics industry, D-M-E is one of the world's leading manufacturers and suppliers of mold base assemblies and components, moldmaking equipment and supplies, and other specialized systems. Offering the industry's broadest and deepest line of mold-tooling technologies, D-M-E's mission is to help its customers produce better parts, reduce their costs, and accelerate time-to-market, as well as enable moldmakers to devote more time to refining their molds. D-M-E has five machining facilities in North America and Europe and direct and joint venture operations worldwide.

Web Site Grows in Stature

D-M-E's current Web site plays an increasingly important role as a greater number of customers go online to purchase products. Approximately 5 percent of D-M-E's sales come from the Web, a number that is growing at the rate of 20 percent a year. To accommodate this growth and

encourage more customers to use the Web site, D-M-E realized it needed a more robust online environment. It decided to upgrade to IBM WebSphere Commerce Professional Edition while leveraging the iSeries environment—providing the scalability and stability required to support the business-to-business environment. The solution includes IBM's DB2 Universal Database, WebSphere Application Server, and HTTP server.

Out-of-the-Box Functionality

Although D-M-E's IT staff is small and has limited experience in Web development, it found IBM's WebSphere Commerce an easy-to-use, complete solution for the Web. In configuring a new store for its Web site, D-M-E only had to modify the sample store template to customize it for its needs. D-M-E also was able to migrate its product catalog, because WebSphere Commerce imports the necessary HTML and JavaServer Pages files, catalog information, payment and tax details, and shipper information to build a complete store archive template that is ready for

customization. Todd Oliver, a system programmer for D-M-E, pointed out that “WebSphere Commerce is a great product for companies like ours because it didn’t require that we have a lot of experience with Java technologies and JavaServer Pages.”

Easy Administration, Better Customer Service

D-M-E has leveraged WebSphere Commerce catalog management tooling to simplify the management of its product catalog, the main feature of its Web site. Internally, D-M-E’s business users can now make changes to catalog products and pricing without requesting help from the IT department, making both groups more productive. The workload on the customer service department has also decreased as more customers check product pricing and order status online.

On the customer side, users have the assurance that the D-M-E Web site will be available 24/7. This encourages more customers to use the site, in turn creating greater efficiencies internally as customers and D-M-E affiliates rely more upon the Internet to place orders. D-M-E still has customers who place orders by phone and fax, but now the stability and robustness of the site provides customers with another reliable sales channel.

Improvements Drive Incremental Sales

Customers can now easily order multiple products at a time online. “With the old system, customers had to know the SKU number for a product, type it in, click ‘go’ and repeat the same process for each product they wanted,” explains Oliver. “With WebSphere Commerce, we added a quick order page that allows customers to add up to 10 products at a time.”

Order processing speed has also increased sevenfold, and the search capabilities of WebSphere Commerce allow customers to define and limit their searches for better results. Also popular with customers is the automated e-mail notification containing forgotten password information, a capability of WebSphere Commerce.

Since the introduction of the Web site, incremental sales have increased 23 percent due to faster order processing and the capability for customers to order more than one product at a time. One measure of success of the new D-M-E Web site is the absence of customer complaint calls fielded by the IT department. Instead of three to four calls per week from customers with questions and/or concerns about the Web site, D-M-E now rarely receives calls.

Application Development Environment Gets Kudos

When D-M-E migrated its Web site to WebSphere Commerce, it faced two hurdles: a small IT staff and limited experience in Web development. Working with IBM

Business Partner Dynamik Technologies, Stobbe and Oliver designed, constructed, tested, and deployed the new site using IBM Rational Application Developer for WebSphere (formerly known as WebSphere Studio Application Developer). Though the group faced some development challenges, their work was simplified by the Rational development environment.

As an integrated development tool, Rational Application Developer for WebSphere accelerates Java 2 Enterprise Edition (J2EE) technology-based application development with a comprehensive set of visual productivity tools, templates, and wizards. D-M-E found the search capabilities, self-help, debugging tools and built-in test environment to be the most useful. Jay Stobbe, Manager of IT systems and administration for D-M-E estimated that IBM WebSphere Commerce cut D-M-E’s Web development time by at least half.

The Future Looks Bright

In the near future, D-M-E plans to take advantage of the marketing tools available in the IBM WebSphere Commerce Professional Edition to increase revenues and grow its customer base. “We have not even begun to explore all the possibilities WebSphere Commerce provides,” says Stobbe. The D-M-E marketing department is excited about leveraging capabilities, such as support for running promotions and executing special offers by automatically presenting targeted cross-selling, up-selling, and advertisements to new and existing customers, as well as the analytics and business intelligence tools. Thanks to WebSphere Commerce, D-M-E has the opportunity to explore areas it had not even dreamed of and has the flexibility to easily add new applications.

Sources: IBM (2005) and D-M-E (2007).

Questions

1. From D-M-E’s point of view, what kind of e-commerce application is this: e-procurement, sell-side, collaborative commerce, or other? Justify your answer.
2. Why did D-M-E decide to utilize WebSphere for its portal? Explain in detail.
3. Imagine you were in charge of selecting the application to be used with the portal. What criteria would you use in making your choice? How does IBM’s WebSphere meet these criteria?
4. How can D-M-E leverage this application with other e-commerce processes?

REFERENCES

- Amazon.com*. "Web Services FAQs." amazon.com/gp/browse.html/ref=sc_bb_1_0/103-7982286-7763064?no=3435361&mc=a361942tsj2aja&node=3434651 (accessed March 2007).
- Aron, R., E. K. Clemons, and R. Reddi. "Just Right Outsourcing: Understanding and Managing Risk." *Proceedings, 38th HICSS*, Big Island, HI, January 3–6, 2005.
- Beer, S. "Google Apps Begins Software as a Service Paradigm Shift." *iTWire*, November 14, 2006. itwire.com.au/content/view/5439/983 (accessed February 2007).
- Bieberstein, N., M. Fiammante, K. Jones, and R. Shah. *Service-Oriented Architecture Compass—Business Value, Planning, and Enterprise Roadmap*. Upper Saddle River, NJ: Pearson Education, 2005.
- Birman, K. *Reliable Distributed Systems: Technologies, Web Services, and Applications*, 1st ed. New York: Springer Publishing Company, 2005.
- Business.com*. "The Search Engine for Managing and Growing your Business 2007." business.com/directory/internet_and_online/ecommerce/b2b_solutions/ (accessed February 2007).
- Cantara, M. "Market Focus: Trends and Forecasts for IT Professional Services for Web Services and SOA." Gartner, Inc., June 2005.
- Coffee, P. "Data Integration Is IT's Tricks." *eWeek*, February 2004.
- Dahanayake, A., H. G. Sol, and Z. Stojanovic. "Methodology Evaluation Framework for Component-Based System Development." *Journal of Database Management*, March 2003.
- Daylami, N., T. Ryan, L. Olfman, and C. Shayo. "Determinants of Application Service Provider (ASP) Adoption as an Innovation." *Proceedings of 38th HICSS*, Big Island, HI, January 3–6, 2005.
- Deitel, H., et al. *Web Services: A Technical Introduction*. Upper Saddle River, NJ: Prentice Hall, 2003.
- D-M-E. "About D-M-E." dme.net/wwwdme/1about.asp (accessed March 2007).
- Dutta, A., and R. Roy. "Offshore Outsourcing: Counteracting Forces and Their Dynamic Effects." *Proceedings, 38th HICSS*, Big Island, HI, January 3–6, 2005.
- Duvall, M. "SOA: TD Banknorth Is Banking on IT." October 2, 2006. baselinemag.com/article2/0,1540,2022612,00.asp (accessed March 2007).
- Erl, T. *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Upper Saddle River, NJ: Prentice Hall PTR, 2004.
- Finkelstein, C. "The Enterprise: Service-Oriented Architecture (SOA)." *DMReview.com*, January 2005. dmreview.com/article_sub.cfm?articleid=1016488 (accessed February 2007).
- Havenstein, H. "Customers Courted with Web 2.0: Social Networking Inspires Marketers." *Computerworld*, February 19, 2007.
- Hinchcliffe, D. "The SOA with Reach: Web-Oriented Architecture." 2006. blogs.zdnet.com/Hinchcliffe/?p=27 (accessed February 2007).
- Hoff, R. D. "Mix, Match and Mutate." *BusinessWeek Online*, 2006. businessweek.com/print/magazine/content/05_30/b3944108_mz063.htm.
- Hower, R. "Web Site Test Tools and Site Management Tools." softwareqatest.com/qatweb1.html (accessed March 2007).
- IBM. "D-M-E Molds New Vision for E-Commerce with IBM Solution." December 21, 2005. www-306.ibm.com/software/success/cssdb.nsf/CS/HSAZ-6KB34W?OpenDocument&Site= (accessed February 2007).
- IBM. "IBM WebSphere Portal Express V6.0: A Portal, Web Content Management, Document Management, and Collaboration Solution for Small and Medium Businesses." January 2007a. www-306.ibm.com/common/ssi/fcgi-bin/ssialias?infotype=an&subtype=ca&htmlfid=897/ENUS207-017&appname=isource (accessed March 2007).
- IBM. "IBM Virtualization: Optimize Your Infrastructure." February 2007b. www-03.ibm.com/systems/virtualization/index.html?ca=vedemot&met=web&me=escallout (accessed February 2007).
- Iverson, W. *Real World Web Services*, 1st ed. Cambridge, MA: O'Reilly, 2004.
- Jakovljevic, P. J. "Architecture Evolution: Service-Oriented Architecture versus Web Services." September 20, 2006. technology-evaluation.com/Research/ResearchHighlights/ERP/2006/09/research_notes/TU_ER_PJ_09_20_06_1.asp (accessed March 2007).
- Kanter, J., and J. J. Walsh. "Toward More Successful Project Management." *Information Strategy: The Executive Journal*, Summer 2004.
- Kendall, K. E., and J. E. Kendall. *Systems Analysis and Design*, 6th ed. Upper Saddle River, NJ: Prentice Hall, 2005.
- Kern, T., and J. Kreijger. "An Exploration of the ASP Outsourcing Option." *Proceedings, 34th HICSS*, Hawaii, January 2001.
- Kishore, R., H. R. Rao, K. Nam, S. Rajagopalan, and A. Chaudhury. "A Relationship Perspective on IT Outsourcing." *Communications of the ACM*, December 2003.
- Kucharvy, T. *The Five Rules for Jump-Starting the Utility-Computing Market*. Boston, MA: Summit Strategies, Inc., January 2003.
- LaMonica, M. "Intel Unveils 'Web 2.0' Software Suite." *CNETNews.com*, November 7, 2006. news.com.com/Intel+unveils+Web+2.0+software+suite/2100-1012_3-6133294.html (accessed April 2007).
- Malik, O. "Rolling Your Own Applications." *Business 2.0*, 2006.
- Martin, E. W., et al. *Managing Information Technology*, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2002.

- McAfee, A. "Enterprise 2.0: The Dawn of Emergent Collaboration." *MIT Sloan Management Review* 47, no. 3 (2006).
- McKeefry, H. L. "Mom and Pop Camping Shop Gets E-Commerce Boost." *eWeek*, May 1, 2006. eweek.com/article2/0,1759,1955999,00.asp (accessed February 2007).
- Mendelsohn, T. "ATG Ups the Ante for B2C Commerce Platforms." *The Forrester Wave Vendor Summary*, Q3 2006." September 14, 2006. forrester.com/rb/search/results.jsp?Ntt=ATG&Ntk=MainSearch&Ntx=mod_e+matchallany&N=0+100010+100011+100012+100013+100020 (accessed March 2007).
- Microsoft.com. "Case Study: Gulf Coast Seal." June 20, 2006. microsoft.com/casestudies/casestudy.aspx?casestudyid=49807 (accessed February 2007).
- Microsoft.com. "Introducing Commerce Server 2007." 2007a microsoft.com/commerceserver/default.msp (accessed March 2007).
- Microsoft.com. "The Business Case for Portals." 2007b. microsoft.com/uk/windowsserversystem/portals/business/case/default.msp (accessed March 2007).
- NetIQ.com. "PersonalCreations.com Boosts Online Sales Over 30 Percent with WebTrends Reporting Service." January 13, 2004. netiq.com/news/releases/release.asp?cid=20040112175217TMIN (accessed March 2007).
- Newcomer, E., and G. Lomow. *Understanding SOA with Web Services (Independent Technology Guides)*. Upper Saddle River, NJ: Pearson Education, 2005.
- Nickerson, R. "An E-Commerce System Model." *Eighth Americas Conference on Information Systems*, Dallas, TX, August 9–11, 2002.
- O'Reilly, T. "What Is Web 2.0? Design Patterns and Business Models for the Next Generation of Software." 2005. oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html (accessed February 2007).
- Overby, S. "Bringing I.T. Back Home." *CIO Magazine*, March 1, 2003.
- Portland Oregon Trail Blazers. "Trail Blazers and Hands on Greater Portland Launch First Social Networking Site for Fans in Partnership with Affinity Circles." February 24, 2007. nba.com/blazers/news/Trail_Blazers_and_Hands_on_Gre-209896-1218.html (accessed February 2007).
- Prince, B. "Verio Sets Sights on Online SMEs." *eWeek.com*, December 12, 2006. eweek.com/article2/0,1895,2071648,00.asp (accessed December 2006).
- Rammer, I., and M. Szpuszta. *Advanced .Net Remoting*. Berkeley, CA: Apress, 2005.
- Rapoza, J. "Microsoft Takes Web Development Leap." *eWeek*, 2006.
- Rauch, M. "Virtuality: How IBM Uses Web 2.0 to Grow Its Brand from Wikis to Viral Video to Second Life." *Sales and Marketing Management*, January–February 2007.
- Reynolds, J. *The Complete E-Commerce Book: Design, Build, and Maintain a Successful Web-Based Business*. Gilroy, CA: CMP Books, 2004.
- RightNow Technologies, Inc. "TD Banknorth Inc. Bolsters Competitively Differentiated Customer Service, Supports Business Growth, and Reduces Costs with RightNow." 2007. rightnow.com/resource/casestudy.php?id=2062&product=9 (accessed February 2007).
- Rodriguez, J., S. Chan, B. Gonzalez, G. Kroner, M. Parlangelo, S. Schwidler, and A. Vena. "IBM WebSphere Portal V5: A Guide for Portlet Application Development." *IBM International Technical Support Organization*, 2004.
- Sasse, A. "Trail Blazers and Hands on Portland Team Up." *Portland Trail Blazers Press Release*, November 3, 2006, affinitycircles.com/news/hop_blazer_rls_11_3_06.pdf (accessed February 28 2007).
- Schneider, G. P. *Electronic Commerce: The Second Wave*, 5th ed. Boston, MA: Course Technology, 2004.
- Schwalbe, K. *Information Technology Project Management*, 4th ed. Boston, MA: Course Technology, 2006.
- Scott, J. E., and I. Vessey. "Managing Risks in Enterprise Systems Implementations." *Communications of the ACM*, April 2002.
- Shirky, C. "Web Services: An Executive Summary." *XML.com*, April 12, 2002. webservices.xml.com/pub/a/ws/2002/04/12/execreport.html (accessed March 2007).
- Shurville, S., and J. Williams. "Managing In-House Development of a Campus-Wide Information System." *Campus-Wide Information Systems* 22, no. 1 (January 2005).
- Sommerville, I. *Software Engineering*, 7th ed. Boston: Addison Wesley, 2004.
- Stal, M. "Web Services: Beyond Component-Based Computing." *Communications of the ACM*, October 2002.
- Systinet. *Introduction to Web Services Architecture*. Burlington, MA: Systinet Corp, 2002.
- Tabor, R. *Microsoft .NET XML Web Services*. Indianapolis, IN: Sams, 2002.
- Tachyon Solutions. "Campmor, Inc.: An E-Business Success by Any Measure." tachyonsolutions.com/company/resources/campmorecomstudy.pdf (accessed February 2007).
- Then, J. "The SOA Puzzle." *Computerworld Malaysia*, April 2005. computerworld.com.my/ShowPage.aspx?pagetype=2&articleid=506&pubid=4&issueid=34 (accessed March 2007).
- Treese, G. W., and L. C. Stewart. *Designing Systems for Internet Commerce*, 2d ed. Reading, MA: Addison-Wesley, 2002.
- Tyler, D. "SOA: Is It Right for You?" *ITToolbox.com*, September 20, 2005. blogs.ittoolbox.com/emergingtech/soa/archives/soa-is-it-right-for-you-part-i-5834 (accessed February 2007).
- Utilitycomputing.com. "Softlayer Delivers Infrastructure for Web 2.0." utilitycomputing.com/print/857.asp (accessed February 2007).
- Webtrends.com. "Personal Creations Boosts Online Sales by 31 Percent with Webtrends." 2004. webtrends.com/upload/cs_personalcreations_001.pdf (accessed January 2007).

- WHIR News. "Web Host Verio Releases ShopSite 8.1." December 13, 2006. thewhir.com/marketwatch/121306_Web_Host_Verio_Releases_ShopSite_8.1.cfm (accessed February 2007).
- Whitten, J., and Bentley, L. *Systems Analysis and Design Methods*, 7th ed. New York: Irwin/McGraw-Hill, 2007.
- Wilson, R. F. "The Six Simple Principles of Viral Marketing." *Web Marketing Today*, February 1, 2005. wilsonweb.com/wmt5/viral-principles-clean.htm (accessed February 2007).
- Xia, W., and G. Lee. "Grasping the Complexity of IS Development Projects." *Communications of the ACM*, May 2004.
- Zhao, J., H. Lu, and V. B. Misic. "Towards Electronic Commerce Management Systems: Concepts and Architecture." *Proceedings of the First International Conference on Telecommunications and Electronic Commerce*, Nashville, TN, November 19–22, 1998.