CHAPTER 2

Enterprise Systems

Learning Objectives

After completing this chapter you will be able to:

- Describe the different types of enterprise systems (ES) and their roles in managing business processes.
- Identify the different vendors that supply ES and analyze their market positions.
- Explain the architecture of a typical ES.
- Compare and contrast the different types of master data in an ES.
- Portray the hypothetical company Super Skateboard Builders (SSB) in terms of its history and growth, products, customers, suppliers, and the types of information systems it uses.

In Chapter 1, we briefly introduced enterprise systems (ES) in the context of the business processes that are discussed in this book. In this chapter, we will explain enterprise systems in greater detail, and we will illustrate their role in managing and executing business processes. We will also describe different types of ES, explain the architecture of a typical ES, and identify the major ES software vendors in the market today. We will introduce you to the world's largest ES vendor, SAP®, in greater detail to improve your understanding of the business model that ES vendors employ and how ES vendors aid companies in managing their specific business processes. Next, we will discuss the various types of data in an ES and how they relate to business processes.

In addition to discussing ES, we will introduce the hypothetical company, Super Skateboard Builders (SSB), Inc., which we will use throughout the book to illustrate how business processes work and how ES support these processes. Finally, we will explain the simulated SAP environment that will be used for the chapter exercises in this book to help you understand how ES support business processes.

2.1 ENTERPRISE SYSTEMS

In today's competitive global environment, it is difficult to imagine how large companies could function without the aid of ES. Beginning in the 1960s, ES began to play a critical role in automating and managing repetitive, manual activities in large businesses. These systems have evolved from stand-alone systems running on large, expensive computers to packaged applications to distributed systems using cheaper, smaller systems. This evolution is briefly described next and is depicted in Figure 2.1. The three broad stages of evolution include the mainframe environment, client-server systems, and service-oriented architecture (SOA). Each of the stages in this evolution is discussed next.

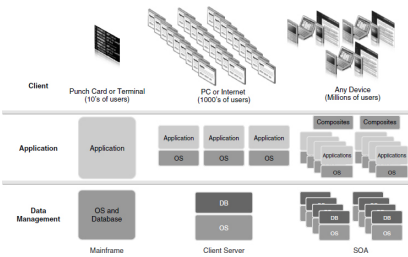

Figure 2.1 The evolution of enterprise systems

2.1.1 STAGE 1: STAND-ALONE MAINFRAME SYSTEMS

Components of an ES include **hardware**, **software**, and a **database**. In the early days of ES, hardware typically consisted of large, expensive mainframe computers. Software includes specialized **operating system** software needed to execute operations from the applications on the hardware and **custom applications** that provide capabilities needed to complete specific tasks, such as filling a customer order. Databases are used to store the data associated with the ES. Early databases were extremely complex and difficult to manage. Only very large companies could afford to acquire and implement an ES. In addition, the way users interacted with the ES was very primitive. They had to use a terminal that was physically connected to the mainframe to input commands to the system using *punch cards* (look it up on Wikipedia). There was no such thing as a computer monitor or a mouse.

For most of the 1960s, 1970s, and 1980s, IBM was the only company that could provide the mainframes, operating systems, and databases for the early ES. For several decades the mainframe architecture was the predominant technology that companies utilized to manage ES. It would take a combination of several innovations in technology and business practices, such as globalization, to break that paradigm and make ES accessible and affordable to a much broader range of businesses.

One of the major drawbacks of the mainframe architecture was its limited **scalability**. Scalability is a concept related to the number of users or the volume of operations that a given hardware/software combination can manage. Scalability is determined by many different aspects of the configuration of a hardware and software combination. In the world of personal computers, you might have experienced situations where you noticed that your computer runs very slowly when a certain application is running or that the performance of a particular video game is especially bad. In these cases, the application (or game) is using a large amount of hardware resources. To increase the scalability of your personal computer to handle the additional workload of the application or game, you could "upgrade" your computer by adding more memory or an improved video processing card. However, in an ES environment this is often very difficult to achieve. Usually, systems are designed to handle a maximum amount of users or operations and are not designed for easy hardware upgrades. Typically, the more scalability (i.e., the number of users or transactions) needed for an application, the higher the upfront costs in hardware, so companies usually purchase the "biggest" hardware they can afford for the job they need to accomplish. In the mainframe architecture, scalability was very low, meaning that only a handful of employees could use the system at any one time. Moreover, the custom application, operating system, and database were entirely contained on a single piece of hardware. This arrangement made increasing scalability prohibitively expensive because of the extremely high costs of bigger mainframe hardware. As a result, companies had to limit both the capabilities and the number of users for each system.

Another limitation of the early ES was that they were *custom designed*, meaning they were designed to address the specific needs of individual firms. Thus, they could not easily be used by other companies. Building a custom system is more expensive than utilizing an existing system, and most small and medium-sized firms could not afford this expense. This is another reason why only the largest companies utilized ES.

In the 1970s, software firms began to develop **packaged applications**. In contrast to proprietary applications that are designed specifically for one firm, packaged applications are generic software that can be used by many companies. By building one version of an ES, the software vendor can spread the cost of software development across many of its customers and thereby reduce the purchase price of the ES for each company. Companies could purchase packaged applications at a fraction of the cost of developing them themselves and then modify them to suit their particular needs. This approach, pioneered by **SAP** in 1972, generated significant cost savings for companies. As a result, the use of ES became much more widespread, at least among large companies.

Although the shift to packaged applications significantly reduced the costs of the total solution, the IBM mainframe hardware, operating system, and database needed to run the packaged application remained very

expensive. Thus, the use of ES was still generally confined to larger firms. However, certain technological advances during the 1980s and 1990s led to an increased use of ES. To begin with, hardware and software continued to evolve to become more efficient and capable and less expensive. For example, more advanced operating systems such as UNIX made it easier to manage complex hardware. Similarly, relational databases such as Informix made storing and accessing data much simpler. Another important development was the advent of networking technologies. Companies that are connected to *networks* can access ES remotely rather than having to physically install a system. Ultimately, however, it was the emergence of the **three-tier client-server** architecture that led to the explosive growth in the use of ES.

2.1.2 STAGE 2: CLIENT-SERVER ARCHITECTURE

Think of a desktop application that you routinely use, such as word processing, spreadsheet, or presentation applications. These applications consist of three components: (1) how you interact with the application (using menus, typing selecting), (2) what the application allows you to do (create formulae or charts, compose an essay), and (3) where the application stores your work (on your hard drive or flash drive). These three components or layers are called the **presentation layer**, **application layer**, and **data layer**, respectively. In the desktop applications mentioned earlier all three layers are contained in one system. In contrast, the three-tier client-server architecture separates these layers into three separate components. This three-tier architecture is illustrated in Figure 2.2. Some systems use a two-tier architecture in which the presentation layer is separated from the application and data layers.

Much of the work you do on the Internet uses either a two-tier or three-tier architecture. Your browser is the presentation layer. You connect to many systems (Web sites) that provide a variety of capabilities (e-mail, information sharing). The Web sites to which you connect combine the application and data layers. Actually, some of these applications use a three-tier architecture by separating the application and data layers, but this is not evident when you use these systems. Another example of a three-tier architecture is instant messaging (IM). When you use IM, you first have to download and install a small piece of software on your computer. This software is the presentation layer, or the user interface to the messaging system that you connect to. It is also the "client" component of the client-server system. The messaging system provides the application layer or the functionality (i.e., the ability to exchange messages with others) as well as the data layer (stores your login information, your contacts, and so on). The messaging system that you connect to is the "server" part of the client-server system.

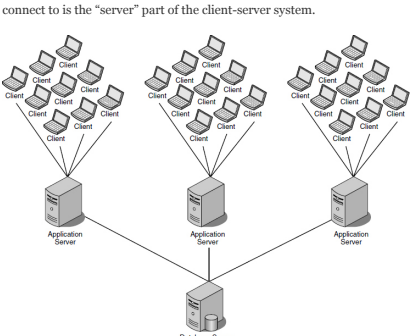

Figure 2.2 The three layers of the client-server architecture

These examples of three-tier architecture illustrate two key points. First, you have to download and install the presentation layer—that is, the client software—on your computer. This requirement clearly illustrates the separation of the system into parts. If you don't install the client software, you can't use the system. Contrast this to installing a desktop application, where the entire system is installed in one step, because all three layers are combined. Second, you can access the messaging system from any computer that is connected to the Internet. You can do this because the data (your login information and contacts) are stored on the server. This is another indication that the presentation layer is separated from the data and application layers.

Separating the three components in the client-server architecture has enabled ES to achieve much greater scalability and flexibility because the different layers can run simultaneously on different computers rather than exclusively on a single mainframe. Consequently, the ES can be run on considerably cheaper computers. Many companies, such as Sun Microsystems, provide a variety of smaller server computers that are so inexpensive that companies can usually buy several servers for the price of a single mainframe.

By distributing the workload of the application across multiple, smaller application servers, companies can achieve nearly infinite scalability by

simply adding more servers. However, to maintain data integrity, each application server has to store its records in a single database that resides on a separate server used solely for the database. Each application server can handle several simultaneous users. Thus, if more users need to utilize an ES, a company simply adds additional application servers to handle the increased workload.

The shift to the three-tier client server dramatically reduced the costs of acquiring, implementing, and using an ES while significantly increasing the scalability of the systems. These two forces transformed ES from a capability that only a few large companies could afford into a technology that tens of thousands of companies can now utilize.

2.1.3 STAGE 3: SERVICE-ORIENTED ARCHITECTURE

In the early 2000s, companies began to Web-enable their three-tier applications so that users could access the systems through a Web browser. During these years companies also benefited from new technologies that could help link, or *integrate*, many different client-server systems together in new and very valuable ways. These new technologies are collectively labeled **service-oriented architecture**, or **SOA**. By using **Web services**, companies could now integrate several client-server applications and create an enterprise **mashup**, or **composite applications**. Composite applications and mashups rely on Web services to send and receive data between and among ES. In addition, they execute newer and more specific processes than are found in the standard ES.

For example, if your company wants to see the map locations of every customer in San Francisco, you could create a composite application or mashup between the ES that contains your customer data and Google Maps. This new application would take user input for the city and state where you want to locate customers, retrieve the appropriate customer data (customer name, address, city, state, zip code) from the ES via a Web service, and then superimpose the customer addresses onto a Google Map for the users to view. The users have no idea which system the customer data are coming from or how these data are being sent to Google Maps. They simply input the city from which they want customer information, and a map pops up with all the customers' locations highlighted.

Take another example of a Web service involving a purchase of something via a company's Web site, which provides estimated shipping charges from various shipping companies and allows the customer to choose a shipping provider. To do this, the company's enterprise system electronically sends shipment information, such as weight, number of boxes, and the destination zip code, to each of the shipping companies' systems. Once the shipping companies receive the data, their systems instantly provide a quotation, via a Web service, and transmit it back. This information is displayed to the customer, who then chooses a provider. Next the company transmits the final shipping information to the selected shipping provider. The shipping provider then sends a confirmation with the delivery information and tracking number for the shipment, which is provided to the customer. To ensure that the customer does not get frustrated and terminate the order, all of these operations must happen in a few seconds. The systems involved could physically be located on servers sitting on several different continents. Web services provide a standardized and reliable way for multiple systems to communicate in a very fast and scalable way to bring added value and efficiency to a business process.

Companies such as SAP have invested billions of dollars to service-enable their core ES so that these systems can be exposed and connected to an infinite number of composite applications and third-party ES. By using SOA to integrate and expose the business processes and data inside an ES, companies can now create new composite applications quickly and inexpensively. In essence, SOA enables companies to build composite applications on top of their existing three-tier client-server applications without changing the existing applications. This gives companies an entirely new level of flexibility for an extremely low cost.

2.1.4 TYPES OF ENTERPRISE SYSTEMS

The paradigm shift in computing architecture occurred simultaneously with the business trends of globalization and business process reengineering (BPR). As companies began to expand their operations globally in the 1980s and began to view organizations from a process view rather than a functional view, BPR emerged in the 1990s as a way to reorient operations around business processes to better manage and control the globally distributed organization.

Enterprise resource planning (ERP) systems

The ES that is most closely associated with the BPR and globalization in the 1990s is **enterprise resource planning (ERP)**. SAP was the first company to create a fully integrated and global ERP system. ERP systems are the world's largest and most complex ES. They focus primarily on the internal operations of an organization, and they integrate functional and cross-functional business processes. Typical ERP systems include Operations (Production), Human Resources, Finance and Accounting, Sales and Distribution, and Procurement. Thus, ERP systems support

processes within a company, or *intracompany processes*. Figure 2.3 shows the solution map for the ERP system developed by SAP. The solution map identifies the functionality and processes supported by the system. This illustration should give you some appreciation for the scope and size of a typical ERP system.

As more companies acquired ERP systems, the next step in the evolution of ES was to connect these systems so they could support *intercompany processes*; that is, processes that take place between and among companies. Examples of connected ES are **supply chain management (SCM)** and **supplier relationship management (SRM)** systems, which connect a company's ERP system to those of its suppliers. SCM connects a company to other companies that supply the materials it needs to make its products. Typical SCM systems help companies plan for their production demand requirements and optimize complex transportation and logistics for materials. SRM systems typically manage the overall relationships with the materials suppliers. SRM systems contain functionality to manage the quotation and contracts processes.

On the other side of the manufacturing and sales process, **customer relationship management (CRM)** systems connect a company's ERP system to those of its customers. CRM systems provide companies with capabilities to manage marketing, sales, and service for its customers. **Product life cycle management (PLM)** systems help companies administer the processes of research, design, and product management. In effect, PLM systems help companies take new product ideas from the virtual drawing board all the way to the manufacturing facility. Figure 2.4 depicts the relationship among the different applications.

The collection of these systems is called an **application suite**. Suite vendors, such as SAP and Oracle, provide fairly comprehensive collections of applications that offer an enormous amount of functionality and cover most of the standard business processes in a company.

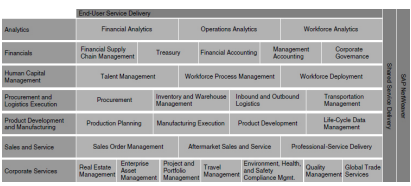

Figure 2.3 The SAP® ERP solution map

Source: Copyright SAP AG 2008

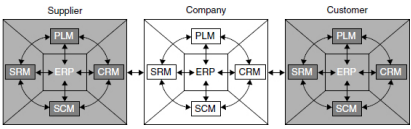

Figure 2.4 The ES application suite

Figure 2.4 identifies the various capabilities that are part of the SAP® Business Suite. To give you a glimpse of the vast scope of the capabilities contained in a suite, Figures 2.5–2.8 are solution maps for SAP® Supply Chain Management (SAP SCM), SAP® Supplier Relationship Management (SAP SRM), SAP® Product Lifecycle Management (SAP PLM), and SAP® Customer Relationship Management (SAP CRM) systems, respectively. It is important to note that one of the key values of having a complete suite of software is that the data and processes are integrated among the systems in the suite. That is, although these are separate systems, they are designed so that they work together in an integrated manner. This setup reduces a great deal of integration cost and effort.

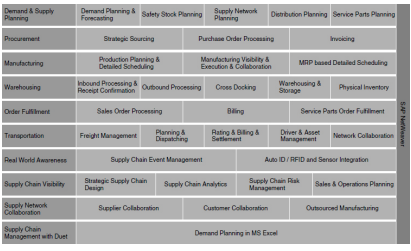

Figure 2.5 The SAP SCM solution map

Source: Copyright SAP AG 2008

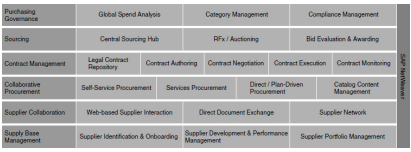

Figure 2.6 The SAP SRM solution map
Source: Copyright SAP AG 2008

Because this is an introductory textbook that focuses on basic business processes, we will discuss only intracompany processes and ERP systems. Keep in mind, however, that the emergence of intercompany business capabilities is one of the most important developments in the modern business environment. However, a fundamental understanding of the key business processes and ERP systems is a prerequisite to advanced topics such as supply chain management and customer relationship management because those processes "feed off" the core ERP-enabled business processes.

Best-of-breed applications

In addition to application suites, today's global companies typically have an ES landscape that includes custom and packaged applications from several vendors. The most common of these applications are **best-of-breed** applications and **niche applications**. Best-of-breed applications are typically isolated to one process or part of a process and have evolved from departmental applications. For example, i2 is a popular supply chain planning system, and Ariba is a popular procurement system. A typical company ES landscape will have one or two core ERP or suite ES plus several best-of-breed applications. Companies need all of these applications because of the silo structure discussed in Chapter 1 and the tendency for each functional group to implement its own systems to suit its specific needs. For the company to function effectively, however, all these applications must be tightly integrated.

Figure 2.7 The SAP PLM solution map
Source: Copyright SAP AG 2008

Figure 2.8 The SAP CRM solution map
Source: Copyright SAP AG 2008

Niche applications

In addition to the large and medium-sized ES vendors, there are thousands of smaller **independent software vendors (ISVs)** who offer highly specialized niche applications for various industries and functions. Many ISVs build composite applications that sit on top of an ES that was developed by a suite or best-of-breed vendor. These ISV composites frequently take advantage of the technical capabilities of SOA to bridge the gap between ES and provide very useful process capabilities to companies. For example, Vendavo is a small ISV that produces a pricing management tool to assist retailers in maximizing profit for retail sales. Vendavo must pull product and pricing information from several ES to do its job of analyzing pricing conditions to suggest the optimum price for a product. Nakisa is another small ISV who produces a visualization tool that dynamically displays the organizational chart of a company with detailed human resources information. Nakisa must take information from the human resources module of an ERP system and present it in a corporate intranet together with data from an organizational charting tool.

Thus far we have classified ERP systems based on their capabilities. However, ERP systems can also be categorized based on the size of the organization that uses them and the way the system's capabilities are delivered.

Size of the enterprise

Company size can be measured in terms of the number of employees or the company's total revenues. Here we focus on the number of employees. For large enterprises (companies with more than 1000 employees), SAP and Oracle tend to be the most popular vendors based on their scalability and industry-specific functionality. For midsized companies (1000–100 employees), SAP and Oracle are joined by Microsoft and Sage. For small companies (fewer than 100 employees), SAP and Intuit are quite popular.

Method of delivery

Traditional, on-premise ES delivery involves installing the software in a typical three-tier or SOA configuration on physical hardware located at

the customer site. Companies purchase the ES software and hardware and then physically install the ES at their facility. In the mid-2000s, a new method of delivery became widespread as companies such as Salesforce.com (http://Salesforce.com) and Net Suite began to deliver ES solutions entirely from the Web. This delivery model—known as software-as-a-service (SaaS)—enables companies to acquire certain ES functionalities without physically installing software on their servers. SAP also has recently launched a new SaaS version of its software for medium-sized companies called SAP® Business ByDesign.

2.2 TYPES OF DATA IN ES

So far we have provided a very broad overview of the evolution of ES and the current ES landscape. We now turn our focus to a more specific topic—the types of data in an ES. Data are the heart of any enterprise system. Every step in every process in an organization uses data created in a previous step and, in turn, creates data that will be used in subsequent steps.

The data in an ES are classified into three types: **organizational data**, **master data**, and **transaction data**. Organizational data are used to define the organizational structure of the business, and they rarely change over time. Master data define the key entities with whom an organization interacts, such as customers and suppliers. These data change, but only occasionally. Finally, transaction data reflect the day-to-day activities of the organization. As you might expect, these data are constantly changing. The bulk of the data in an ES consist of transaction data.

Consider the following example: A business sells some of its products to an established customer located in its California sales region. Data about the customers (name, address, etc.) and the product sold (product number, description, weight, etc.) are master data. The region where the sales occurred—California—is organizational data. The details about the sale (quantity, date shipped, etc.) are transaction data. We will discuss these types of data in greater detail in the following sections.

2.2.1 TRANSACTION DATA

When an organization completes a specific process activity or task, the data that it generates are transaction data. Transaction data typically include general data such as who did what, when, and where, as well as specialized data that relate to the specific task. For example, when a company sends a purchase order to a vendor, the following transaction data are generated: dates, quantities, name of the person requesting the material, name of the person approving the material, prices, where the shipment is to be delivered, and the shipment method. These data are generated each time a purchase order is created. In general, each time any activity takes place in the organization, data that are specific to that activity are created.

2.2.2 MASTER DATA

Master data describe the key entities associated with an organization. Typical entities are customers, vendors, products, and employees. Master data for customers and vendors include name, address, contact person, and a variety of negotiated terms such as billing and payment methods. For products, master data include product number, description, physical characteristics such as weight and color, handling requirements (e.g., fragile), and typical storage location. Employee master data include name, address, position data, payroll data, tax-related data, and benefits data.

Unlike transaction data, master data are not connected to a specific process or process step. However, process steps require master data to be completed. In fact, master data are used repeatedly in executing processes. For example, if master data don't exist for a customer, then the company can't create a sales order for that customer until it generates a new master record.

Just as master data are utilized across many processes, they also typically involve multiple functional areas. (Recall our discussion of the process and functional areas in Chapter 1.) Significantly, each of these functions defines or views the entity (e.g., customer) differently and uses these data differently. For example, with regard to customer master data the sales department maintains data such as address, contact person, and negotiated terms, and the accounting department maintains data about payment history and credit ratings.

Because the master data are shared across multiple functional areas, it is critical that a company maintain a single version of these data that is complete, accurate, and up to date. As a very simple example, consider what happens if sales and accounting each maintain their own version of a customer's data. If the customer's address changes, this change has to be made in both places. What happens if accounting fails to make this change and instead retains the old address? In that case sales will ship the goods to the customer, but the invoice will not be sent to the correct address. The company will then experience delays in receiving payment. Clearly, then, a single definition of data across the organization is essential to maintain consistency and accuracy.

2.2.3 ORGANIZATIONAL DATA

Figure 2.9 shows an organizational structure for a hypothetical company. This structure defines the way the different activities of the business are organized. In our example, the company has one global manufacturing operation and two sales operations. The manufacturing or production operation has facilities, or plants, in three locations: China, Mexico, and India. Sales operations are located in North America and the European Union. Sales operations are further divided into wholesale and retail divisions for both regions plus online sales activities for North America. This organizational structure is typically broken down into greater detail. For example, each of the production regions includes multiple factories in specific locations. In turn, each factory has various structures such as storage areas, shipping points, and receiving points.

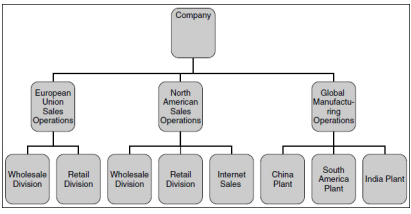

Figure 2.9 A hypothetical organizational structure

All these data are organizational data and are included in the ES as part of the *configuration* process. Configuration is the process of setting up the ES so it can support the work of the business, that is, the business processes. Many other activities are involved in configuring an ES besides defining the organizational structure. A discussion of these activities is beyond the scope of this book. Our primary objective in this discussion is to explain the concept of organizational data. In the next section we examine master data and transaction data in greater detail as they apply to a specific company.

2.3 SUPER SKATEBOARD BUILDERS (SSB), INC.

Super Skateboard Builders (SSB), Inc.[1] was founded in 1997 by John "Z-boy" Boeve, the current president of the company, with the help of a small trust John received from his maternal grandmother, a woman who motivated John to go to college and to continue to pursue his skateboarding passion. She marveled at John's skateboarding finesse and encouraged him to find some way to earn a reasonable living by capitalizing on his passion for the sport. John used the money from the trust to buy the necessary shop and office equipment (storage bins, an assembly table, desks, etc.) and lease a small building that would adequately house a skateboard assembly operation. (See Figure 2.10 for a layout of the facilities.) In addition, John invested in some computer equipment and basic office software. He was a strong believer in using the computer to store and track information related to any of life's worthwhile pursuits, especially if they were data-intensive—as he assumed the operation of his new company would be.

Over the next decade SSB grew rapidly. Although this growth was due in part to John's national name recognition—in high school and college he had racked up many awards competing across the country in skateboarding events—the main reason for the company's success was John's leadership, his vision, his management style, and his perseverance.

2.3.1 PRODUCT LINE

At the present time, SSB has four products. In SAP terminology, these products are trading goods, meaning they are simply purchased in their final form from a supplier and then sold to other retailers or distributors. As you can see in Table 2-1, SSB's four products are entry-level skateboards, helmets, T-shirts, and first aid (repair) kits for skateboards. With current sales exceeding $5 million, SSB is finding it hard to keep up with demand, even though they now employ twice as many workers as when they started. To meet this growing demand, John is considering manufacturing some skateboards in house instead of simply reselling skateboards made by SSB's suppliers.

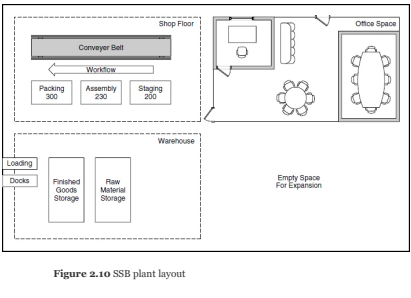

Figure 2.10 SSB plant layout

2.3.2 CUSTOMERS

SSB sells its products only to retailers who then resell the products directly to consumers. Thus, SSB is a wholesaler. Presently SSB has seven major customers. Master data about these customers are shown in Table 2-2.

2.3.3 VENDORS (SUPPLIERS)

As the company's sales increased, it began dealing with a wider variety and number of vendors to supply SSB with the necessary products. Currently SSB deals with seven suppliers on a regular basis. Table 2-3 lists SSB's vendors.

TABLE 2-1 SSB Product List

Product Number	Product Description	Purchase Cost	Selling Price (Wholesale)	MSRP
ENSB3000	Entry-Level Skateboard	$34.00	$45.00	$75.00
HLMT5000	Helmet	$20.00	$27.00	$45.00
SHRT4000	T-Shirt	$7.00	$10.00	$16.00
FAID6000	Skateboard First Aid Kit	$10.00	$16.00	$27.00

TABLE 2-2 SSB Customer List

Customer Number	Name	Address	City	State	Zip
1	World Wide Skateboard Distributors	1229 Main Street	Ann Arbor	MI	48109
2	Extreme Skateboard Sports, Inc.	5000 Rensellear Ave	Detroit	MI	48202
3	"Waldo" Autry	3012 Haslett Road	Lansing	MI	48906
4	West Michigan Sporting Goods, Inc.	6903 28th Street	Grand Rapids	MI	49508
5	Flying Acrobats, Inc.	274 Adams Street	Holland	MI	49423
6	MI Sporting Company	3000 Alpine Ave	Grand Rapids	MI	49544
7	Saginaw Dawgs	4005 State Street SE	Saginaw	MI	48710

2.3.4 EMPLOYEES

As we explained, although SSB is enjoying unexpected success, it is also having a difficult time handling this rapid growth. When the company was small, all the work was done by five people—John, Maria, and three people in the plant. John and Maria handled all the purchasing and accounting tasks, while Catherine, the plant manager, also supervised the warehouse. Much of the work was done informally. For example, when a customer order came in, it was handled—but no one could explain exactly how. When the company needed material from its suppliers, Catherine just figured out what they needed and called a supplier to order it.

As SSB expanded, John hired additional people to handle sales and purchasing and to manage the warehouse activities. The current organizational chart is shown in Figure 2.11. As the number of employees increased, coordinating their work became more important and also more challenging. In the early years John knew what was going on in the entire company. As the company expanded, however, it became more and more difficult to keep up with the activities of all the employees as well as the ever-increasing volume of customer orders, purchases, and so on.

TABLE 2-3 SSB vendors

Vendor Number	Vendor Name	City	State	Zip
100000	Nutcase Supplies	Grand Rapids	MI	49525
100001	Skatelubbers, Inc.	Grand Rapids	MI	49525
100002	Grand Skateboard Supplies	Grand Rapids	MI	49525
100003	Black Widow Skateboards, Inc.	Holland	MI	49424
100004	Spotted Owl Lumber	Cascade	MI	49546
100005	Van Go Paint Supplies	Kentwood	MI	49508
100006	The Dutch Monster	Grand Rapids	MI	49504
100007	Whitewater Development Company	Jonesboro	AR	72401

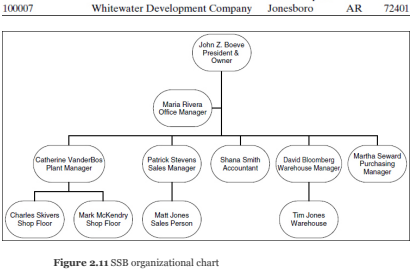

Figure 2.11 SSB organizational chart

2.3.5 INFORMATION SYSTEMS

For years the company had utilized three stand-alone (not networked) computers, each of which ran the same popular office suite. Using spreadsheets and some databases that John had cobbled together, they had managed to enter all pertinent information and keep track, somewhat accurately, of orders, suppliers, and inventory. Under this arrangement the employees often had to run from computer to computer to swap files with one another on floppy disks to be certain that everyone had the latest version of this or that database or spreadsheet.

John was aware that this system was a disaster waiting to happen. For example, SSB once had used the wrong version of the inventory file to make decisions about ordering materials. As a result they ordered far too

many helmets. At first they reassured themselves that this was not really a problem because they eventually would sell the inventory. However, it soon became apparent that the extra helmets took up valuable storage space that could have been used for materials that were needed right away. For a short time some of the extra helmets were actually stored in John's office and in the break room. In addition, shortly after these excess helmets arrived and were paid for, John had to exercise a line of credit to pay for other materials that were needed immediately. Paying for the excess inventory had depleted the company's cash reserves. Exercising the line of credit solved the cash problem, but it also forced the company to incur unnecessary interest charges.

Maria was also having difficulty with the accounting. John's spreadsheets and databases were of little help when it came to keeping the books. Maria was always scrambling to translate the data in these files into information she could then reenter into the accounting software she was using. Data in her system never seemed to be in sync with the files John had set up. One step John took to alleviate this problem was to hire a full-time accountant —Shana—to manage the books. This also freed up Maria's time to better manage the office.

Faced with these and other problems, John sought to apply some of what he had learned in his information system courses in college. He decided to upgrade the computer systems to eliminate having multiple versions of data on different machines and to enable all his employees to share information more efficiently and effectively. All the computers in the company were networked, and a leading enterprise system—SAP—was implemented. The expectation was that everyone now would have access to the same information—no more running around trading spreadsheets or double-checking on who had the version with the latest information. An added plus was that the books would immediately reflect any changes that took place with inventory, sales, payments, receipts, and so on. Finally, the company would be able to do away with Maria's old independent accounting system, meaning that Maria would no longer have to translate and reenter information.

John had high hopes for the new system. Of course, the hard part was just beginning. Everyone had to learn how to use the system, not just in terms of the mechanics but, more important, in terms of how the system would support their work. More specifically, every employee had to understand the various business processes that SSB executed, his or her part in these processes, and the role of the new ES in enabling these processes. There was much learning to be done. College had not quite prepared John to view his company from a process perspective.

We will use SSB as a running example throughout this book to illustrate how key processes work in an organization. We will explain how the different people in SSB are involved in the processes, what their role is, how they interact with others in the organization and how they are dependent on each other to ensure the smooth execution of processes. We will also consider many examples of the negative consequences that occur when SSB adopts a silo view rather than a process view. Ultimately, it is our goal to use the SSB example to make clear to you importance of the skills identified in Chapter 1—strategic thinking, communication, collaboration, and information literacy.

We will be using the ES developed by SAP to illustrate the role of such systems in supporting processes. We will not be using an actual ES; rather, we will use a simulated version of the system. A simulation will allow you to focus on the processes and how they are executed in an ES without having to learn how to use specific software.

We next provide a brief history of SAP, followed by an explanation of how the simulated version of SAP's ES will be used in the book.

2.4 SAP OVERVIEW

Over the course of more than three decades, SAP has evolved from a small, regional enterprise into a world-class international company. In 2008, SAP was the global market leader in ES solutions. Given SAP's enormous global customer base, it is highly likely that at some point during your career you will work in a company that runs SAP software. The evolution of SAP as a company is briefly described next and is depicted in **Figure 2.12**. Because SAP is a German corporation, all its financial statements are denominated in euros, the currency used by the majority of the European Union member nations.

In 1972, five former IBM employees launched a company called **S**ystems **A**pplications and **P**roducts in Data Processing in Mannheim, Germany. SAP is the acronym for the original name of the company, and each letter is pronounced individually, just like IBM or ABC. Their vision was to develop standard application software for real-time business processing. After a year of development, they introduced their first financial accounting software, which formed the basis for what later came to be known as the "R/1 system." "R" stood for real-time data processing. R/1 was an ES based on the IBM mainframe architecture. Users utilized punch cards to execute transactions in the system. By the end of the 1970s, SAP had expanded the capabilities of R/1 significantly. In addition,

the company introduced a new, more powerful product called SAP R/2, which could take advantage of the first text-only video terminals for user inputs.

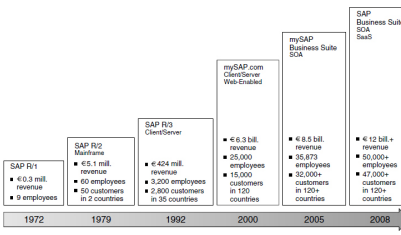

Figure 2.12 The evolution of SAP

Source: SAP AG 2008

In the early 1980s, SAP moved into a new headquarters located in the tiny town of Walldorf, Germany, near Heidelberg, where it still has its global headquarters today. By this time, 50 of the 100 largest German industrial firms were already SAP customers. In order to expand throughout the rest of Europe, SAP designed SAP R/2 to handle different languages and currencies—a revolutionary capability in those early days of enterprise software. Then, in 1992, SAP developed a new ERP solution known as SAP R/3 that was based on the three-tier client-server architecture. The client-server concept, which provided uniform appearance of graphical user interfaces, consistent use of relational databases, and the ability to run on computers from different vendors, met with overwhelming approval from customers. With SAP R/3, SAP ushered in a new generation of enterprise software. To this day, the client-server architecture is the standard in business software.

In late 1999, SAP introduced its new suite of products, which included SAP CRM, SAP SRM, SAP SCM, and SAP PLM applications. These extended capabilities provided companies with collaborative tools to work with their partners and customers on extended business processes. Recall that ERP systems focus on optimizing business processes within a company (intracompany). Collaborative applications, such as the SAP Business Suite, focus on optimizing processes that involve interactions between companies (intercompany) and their customers and partners. With the global adoption of the Internet, the SAP Business Suite could now manage nearly every business process for a company's entire worldwide operations.

In the early 2000s, SAP began to Web-enable the SAP Business Suite of applications and rearchitect them to SOA standards. In 2007, SAP delivered the world's first SOA-based ES suite. It also provided several new capabilities for its customers to create composite applications. Today, more than 12 million users work each day with SAP solutions. The company has almost 50,000 customers in 120 countries, and its products are installed on more than 120,000 servers. SAP's software is translated into 33 languages and is customized to meet the specific needs of 25 major industries. SAP is the world's third-largest independent software vendor, behind Microsoft and Oracle. Perhaps more important, it is the largest enterprise software vendor, ahead of Oracle and Microsoft.

Although this textbook is not an actual SAP training manual, it will provide you with valuable exposure to the SAP ERP system that could prove useful in your future career. In addition, even if you don't work at a company that uses SAP software, you can apply the knowledge of business processes you will obtain from this book to every ES system.

2.5 SAP SOFTWARE (SIMULATION)

To make the concepts discussed in each chapter of this book more real to you, we have included a simulated SAP ERP environment where you will also have the opportunity to execute all the process steps discussed in each chapter. Hopefully, you will become aware of the significant advantages of process execution in ES compared to the manual, or paper-based world.

The simulated SAP environment is based on the actual SAP ERP system that is used by more than 50,000 of the world's best-managed companies. However, because the goal of this textbook is to help you master business process concepts and not to master the ES software, we have simplified the user interface and transactional capabilities significantly.

You will access the simulations via WileyPLUS, and your professor will direct you to the assignments. NOTE that you must already have registered for WileyPLUS using a registration code that came packaged with this text. (If you bought a used textbook, you may need to purchase a registration code seperately. For more information about buying a registration code, see the Preface of this book, in the section titled "WileyPLUS - Information for Students.")

The simulations are accessible from nearly any computer with Adobe Flash version 9 or later installed in your browser. We have tested the simulations with Firefox, Safari, and Internet Explorer. Other browsers should work, but if you have problems, please use one of the tested browsers (they're all free).

You will see that Chapters 2 through 6 have exercises that your instructor can assign. These exercises consist of two parts plus a quiz. The first part is a guided tour of the process or information related to that chapter of the textbook. This guided tour will show you where to click to find information, where to input data, and how to complete the process. There will be several windows with details and explanations of each step. Pay very close attention to the guided exercise! You will need this information for the second part of the exercises.

In the second part you will complete the exercise on your own, without any help or hints. It will include the same steps and activities as the guided exercise, but some of the information used in the guided practice will be changed a bit. The exercise is designed to help you learn how processes are executed but to prevent you from making mistakes. You must click in the right spot or enter the right information to proceed to the next step. If you pay attention in the guided practice, you should not experience any trouble completing the second part on your own.

In addition, your professor may assign you to complete a short quiz about the simulations you just completed. This quiz will be found in the Assignments area of WileyPLUS. These questions are fairly simple, but you must pay attention during the practice and exercise to get the answers right. We recommend that you complete the entire series of practice, exercise, and quiz in one sitting to maximize your learning opportunity.

Let's get started with the exercises! In this chapter we have introduced SSB. The exercises will ask you to retrieve and review a variety of master data from our simulated system. In particular, you will retrieve master data related to customers, vendors, and products. Remember, you will first complete the guided exercise and then complete a similar exercise on your own. Finally, you will take a short quiz. After you have completed the quiz, the system will grade it and send your score to your professor's grade book. You may view your score in the gradebook tab of WileyPLUS. Please check with your instructor as to his or her requirements with respect to the simulated exercises. Note that you may repeat these exercises as often as is necessary to thoroughly understand the master data in our simulated system.

CHAPTER SUMMARY

In this chapter we have considered several aspects of enterprise systems and their role in managing business processes. We have described the evolution of ES from the early days of expensive mainframe systems through the paradigm shift to cost-effective and scalable three-tier client-server applications to today's SOA-based systems.

Enterprise systems can be understood by looking at the functionality they contain. Suites and best-of-breed and niche applications are utilized in most companies today. Suites have the advantage of containing most of the business processes needed to run a company in a tightly integrated package. Best-of-breed solutions tend to have slightly better functionality, but come with the added cost of integrating with other applications and systems. Niche applications are provided by smaller ISVs who build on top of or in between suites and best-of-breed applications to address very specific business processes in companies.

ERP systems are the heart and soul of most companies' ES landscape. ERP processes are extended and complemented by CRM, SCM, SRM, and PLM functionality to collaborate with suppliers and customers.

We reviewed the history of the world's leading ES provider, SAP. SAP's growth and expansion ran parallel with the evolution and adoption of ES solutions around the world. SAP has grown from a tiny German software provider to become the world's largest ES vendor. Nearly every major company in the world depends heavily on SAP solutions to manage their business.

Process steps executed in an ES depend on master data, which are data about key entities such as customers, materials, and vendors. Transactional data are captured at every step of the process to record all the relevant information about what was done in each step. These data are later used to help improve and manage the process. Organizational data are used to define the organizational structure of the company.

We also discussed Super Skateboard Builders (SSB), Inc., the fictional company that we will use throughout this book to illustrate how companies execute the basic business processes. SSB is a wholesaler of skateboards and skating products. It is considering starting to make skateboards rather than simply buying from a vendor and reselling. SSB uses information systems to manage its relationships with vendors, customers, and employees.

KEY TERMS

application layer

application suite

best-of-breed

composite applications

custom applications

customer relationship management (CRM)

data layer

database

enterprise resource planning (ERP)

hardware

independent software vendors (ISVs)

mashups

master data

niche applications

operating system

organizational data

packaged applications

presentation layer

product lifecycle management (PLM)

SAP

scalability

service-oriented architecture (SOA)

software

supplier relationship management (SRM)

supply chain management (SCM)

three-tier client-server

transaction data

Web services

REVIEW QUESTIONS

1. Describe the differences among the three generations of enterprise systems—mainframe architecture, client-server architecture, and service-oriented architecture.

2. Explain the functions of the different systems in an application suite. How are they related?

3. What are the roles of organizational data, master data, and transaction data in an ES?

4. What functional areas are included in SSB's organizational structure? Is this a common organizational structure? Draw the organizational structure of a small or medium-sized company that you are familiar with.

5. Describe the key problems SSB faced with its use of technology to manage its operations before it implemented an enterprise system. How can the ES improve SSB's operations?

ASSIGNMENTS

1. Provide two examples of organizational data, master data, and transaction data within the context of your university or another organization you are familiar with.

2. Service-oriented architecture (SOA) is touted as a technology that will drastically change the way organizations utilize enterprise systems. Research the use of SOA in organizations, and argue whether SOA will be the next major technological development.

3. In this chapter we provided you with examples of vendors that provide suite, best-of-breed, and niche applications. Conduct your own research, and develop a list of vendors and the types of applications they develop.

[1] The SSB case and the various exercises and assignments included in this book are based on the efforts of several faculty members in the Seidman College of Business, Grand State University, that were funded by grants by the Seidman College of Business. We acknowledge efforts of the Seidman faculty and are grateful to the Seidman College for granting us permission to use the data in this book.