CHAPTER 2

Understanding and Modeling Organizational Systems

Learning Objectives

Once you have mastered the material in this chapter you will be able to:

1. **Understand that organizations and their members are systems and that analysts need to take a systems perspective.**
2. **Depict systems graphically using context-level data flow diagrams, entity-relationship models, and use cases and use case scenarios.**
3. **Recognize that different levels of management require different systems.**
4. **Comprehend that organizational culture impacts the design of information systems.**

To analyze and design appropriate information systems, systems analysts need to comprehend the organizations they work in as systems shaped through the interactions of three main forces: the levels of management, design of organizations, and organizational cultures.

Organizations are large systems composed of interrelated subsystems. The subsystems are influenced by three broad levels of management decision makers (operations, middle management, and strategic management) that cut horizontally across the organizational system. Organizational cultures and subcultures all influence the way people in subsystems interrelate. These topics and their implications for information systems development are considered in this chapter.

ORGANIZATIONS AS SYSTEMS

Organizations and their members are usefully conceptualized as systems designed to accomplish predetermined goals and objectives through people and other resources that they employ. Organizations are composed of smaller, interrelated systems (departments, units, divisions, etc.) serving specialized functions. Typical functions include accounting, marketing, production, data processing, and management. Specialized functions (smaller systems) are eventually reintegrated through various ways to form an effective organizational whole.

The significance of conceptualizing organizations as complex systems is that systems principles allow insight into how organizations work. To ascertain information requirements properly and to design appropriate information systems, it is of primary importance to understand the organization as a whole. All systems are composed of subsystems (which include information systems); therefore, when studying an organization, we also examine how smaller systems are involved and how they function.

Interrelatedness and Interdependence of Systems

All systems and subsystems are interrelated and interdependent. This fact has important implications both for organizations and for those systems analysts who seek to help them better achieve their goals. When any element of a system is changed or eliminated, the rest of the system's elements and subsystems are also significantly affected.

For example, suppose that the managers of an organization decide not to hire administrative assistants any longer and to replace their functions with networked PCs. This decision has the potential to significantly affect not only the administrative assistants and the managers but also all the organizational members who built up communications networks with the now departed assistants.

All systems process inputs from their environments. By definition, processes change or transform inputs into outputs. Whenever you examine a system, check to see what is being changed or processed. If nothing is changed, you may not be identifying a process. Typical processes in systems include verifying, updating, and printing.

Another aspect of organizations as systems is that all systems are contained by boundaries separating them from their environments. Organizational boundaries exist on a continuum ranging from extremely permeable to almost impermeable. To continue to adapt and survive, organizations must be able first to import people, raw materials, and information through their boundaries (inputs), and then to exchange their finished products, services, or information with the outside world (outputs).

Feedback is one form of system control. As systems, all organizations use planning and control to manage their resources effectively. Figure 2.1 shows how system outputs are used as feedback that compares performance with goals. This comparison in turn helps managers formulate more specific goals as inputs. An example is a U.S. manufacturing company that produces red-white-and-blue weight-training sets as well as gun-metal gray sets. The company finds that one year after the Olympics, very few red-white-and-blue sets are purchased. Production managers use that information as feedback to make decisions about what quantities of each color to produce. Feedback in this instance is useful for planning and control.

The ideal system, however, is one that self-corrects or self-regulates in such a way that decisions on typical occurrences are not required. An example is a supply chain system for production planning that takes into account current and projected demand and formulates a proposed solution as output. An Italian knitwear manufacturer that markets its clothing in the United States has just such a system. This company produces most of its sweaters in white, uses its computerized inventory information system to find out what colors are selling best, and then dyes sweaters in hot-selling colors immediately before shipping them.

Feedback is received from within the organization and from the outside environments around it. Anything external to an organization's boundaries is considered to be an environment. Numerous environments, with varying degrees of stability, constitute the milieu in which organizations exist.

Among these environments are (1) the environment of the community in which the organization is physically located, which is shaped by the size of its population and its demographic profile, including factors such as education and average income; (2) the economic environment, influenced by market factors, including competition; (3) the political environment, controlled through state and local governments; and (4) the legal environment, issuing federal, state, regional, and local laws and guidelines. Although changes in environmental status can be planned for, they often cannot be directly controlled by the organization.

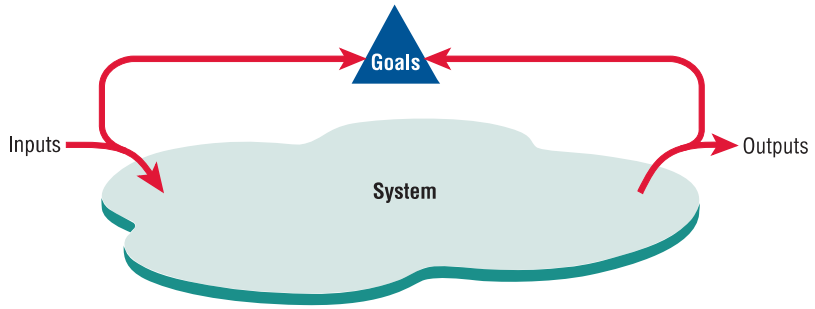


FIGURE 2.1

System outputs serve as feedback that compares performance with goals.

CONSULTING OPPORTUNITY 2.1

The E in Vitamin E Stands for Ecommerce

"Our retail shops and mail-order division are quite healthy," says Bill Berry, one of the owners of Marathon Vitamin Shops, "but to be competitive, we must establish an ecommerce Web site." His father, a co-owner, exclaims, "I agree, but where do we start?" The elder Berry knew, of course, that it wasn't a case of setting up a Web site and asking customers to email their orders to the retail store. He identified eight different parts to ecommerce and realized that they were all part of a larger system. In other words, all the parts had to work together to create a strong package. His list of elements essential to ecommerce included the following:

1. Attracting customers to an ecommerce Web site.
2. Informing customers about products and services offered.
3. Allowing customers to customize products online.
4. Completing transactions with customers.
5. Accepting payment from customers in a variety of forms.
6. Supporting customers after the sale via the Web site.
7. Arranging for the delivery of goods and services.
8. Personalizing the look and feel of the Web site for different customers.

Bill Berry read the list and contemplated it for a while. "It is obvious that ecommerce is more complex than I thought," he says. You can help the owners of Marathon Vitamin Shops in the following ways:

1. Make a list of the elements that are interrelated or interdependent. Then write a paragraph stating why it is critical to monitor these elements closely.
2. Decide on the boundaries and ultimate scope of the system. That is, write a paragraph expressing an opinion on which elements are critical for Marathon Vitamin Shops and which elements can be explored at a later date.
3. Suggest which elements should be handled in-house and which should be outsourced to another company that may be better able to handle the job. Justify your suggestions in two paragraphs, one for the in-house jobs and one for the outsourced tasks.

Related and similar to the concept of external boundary permeability is the concept of internal openness or closedness of organizations. Openness and closedness also exist on a continuum, because there is no such thing as an absolutely open or completely closed organization.

Openness refers to the free flow of information within the organization. Subsystems such as creative or art departments often are characterized as open, with a free flow of ideas among participants and very few restrictions on who gets what information at what time when a creative project is in its infancy.

At the opposite end of the continuum might be a defense department unit assigned to work on top-secret defense planning affecting national security. Each person needs to receive clearance, timely information is a necessity, and access to information is only on a "need to know" basis. This sort of unit is constrained by numerous rules.

Using a systems overlay to understand organizations allows us to acknowledge the idea of systems composed of subsystems; their interrelatedness and their interdependence; the existence of boundaries that allow or prevent interaction between various departments and elements of other subsystems and environments; and the existence of internal environments characterized by degrees of openness and closedness, which might differ across departments, units, or even systems projects.

Virtual Organizations and Virtual Teams

Not all organizations or parts of organizations are visible in a physical location. Entire organizations or units of organizations can now possess virtual components that permit them to change configurations to adapt to changing project or marketplace demands. Virtual enterprises use networks of computers and communications technology to bring people with specific skills together electronically to work on projects that are not physically located in the same place. Information technology enables coordination of these remote team members. Often virtual teams spring up in already-established organizations; in some instances, however, organizations of remote workers have been able to succeed without the traditional investment in a physical facility.

There are several potential benefits to virtual organizations, such as the possibility of reducing costs of physical facilities, more rapid response to customer needs, and helping virtual employees to fulfill their familial obligations to growing children or aging parents. Just how important it will be to meet the social needs of virtual workers is still open to research and debate. One example of a need for tangible identification with a culture arose when students who were enrolled in an online virtual university, with no physical campus (or sports teams), kept requesting items such as sweatshirts, coffee mugs, and pennants with the virtual university's logo imprinted on them. These items are meaningful cultural artifacts that traditional brick-and-mortar schools have long provided.

Many systems analysis and design teams are now able to work virtually, and in fact, many of them marked the path for other types of employees to follow in accomplishing work virtually. Some applications permit analysts who are providing technical assistance over the Web to "see" the software and hardware configuration of the user requesting help, in this way creating an ad hoc virtual team composed of the analyst and user.

Taking a Systems Perspective

Taking a systems perspective allows systems analysts to start broadly clarifying and understanding the various businesses with which they will come into contact. It is important that members of subsystems realize that their work is interrelated. Notice in Figure 2.2 that the outputs from the production subsystems serve as inputs for marketing and that the outputs of marketing serve as new inputs for production. Neither subsystem can properly accomplish its goals without the other.

Problems occur when each manager possesses a different picture of the importance of his or her own functional subsystem. In Figure 2.3 you can see that the marketing manager's personal perspective shows the business as driven by marketing, with all other functional areas interrelated but not of central importance. By the same token, the perspective of a production manager positions production at the center of the business, with all other functional areas driven by it.

The relative importance of functional areas as revealed in the personal perspectives of managers takes on added significance when managers rise to the top through the ranks, becoming strategic managers. They can create problems if they overemphasize their prior functional information requirements in relation to the broader needs of the organization.

For example, if a production manager is promoted but continues to stress production scheduling and performance of line workers, the broader aspects of forecasting and policy making may suffer. This tendency is a danger in all sorts of businesses: where engineers work their way up to become administrators of aerospace firms, college professors move from their departments to become deans, or programmers advance to become executives of software firms. Their tunnel vision often creates problems for the systems analyst trying to separate actual information requirements from desires for a particular kind of information.

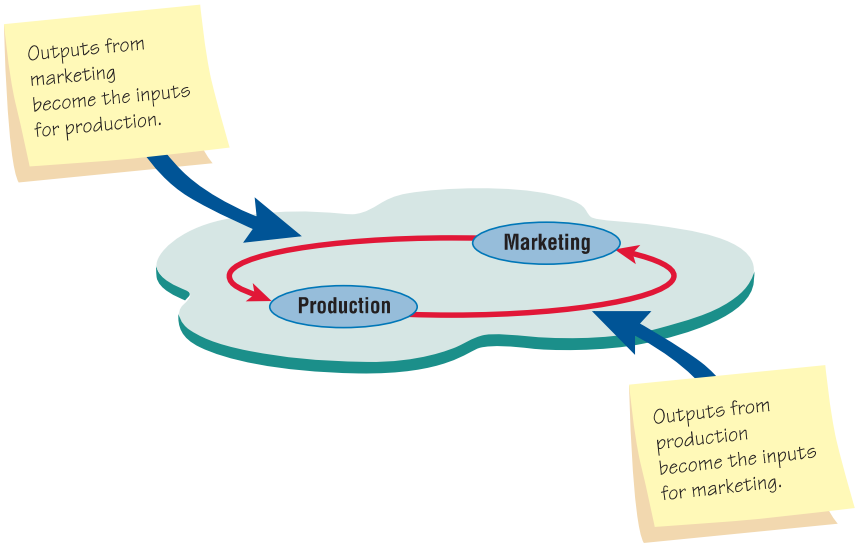
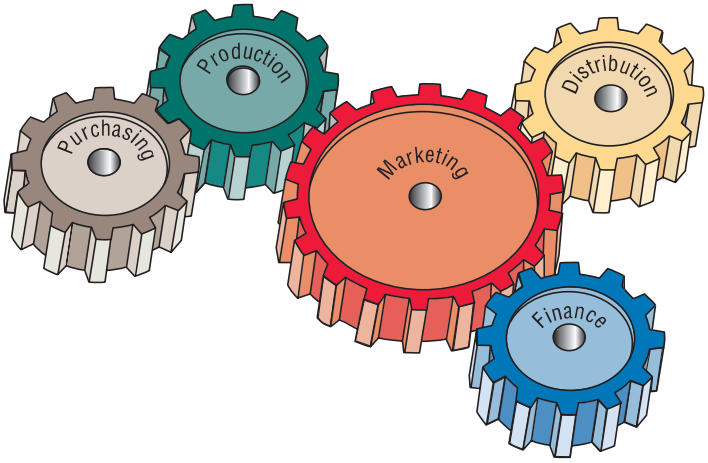


FIGURE 2.2

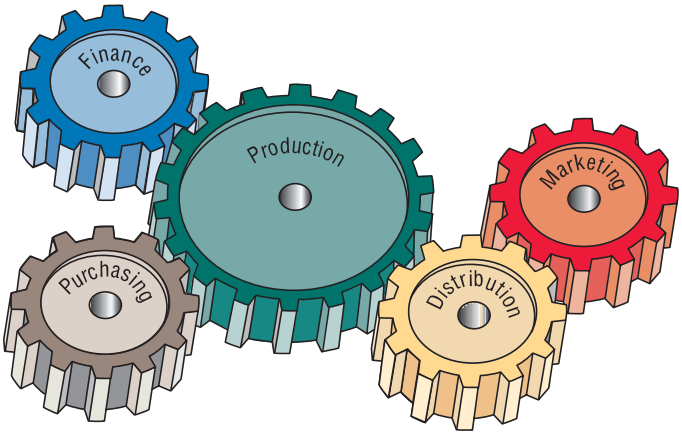
Outputs from one department serve as inputs for another such that subsystems are interrelated.

FIGURE 2.3

A depiction of the personal perspective of functional managers shows that they feature their own functional area as central to the organization.



How a Marketing Manager May View the Organization



How a Production Manager May See the Organization

Enterprise Systems: Viewing the Organization as a System

Enterprise systems, often referred to as enterprise resource planning (ERP) systems, is a term used to describe an integrated organizational (enterprise) information system. Specifically, ERP is software that helps the flow of information between the functional areas in the organization. It is a customized system that, rather than being developed in-house, is usually purchased from one of the software development companies well-known for its ERP packages, such as SAP or Oracle. The product is then customized to fit the requirements of a particular company. Typically, the vendor requires an organizational commitment in terms of specialized user or analyst training. Many ERP packages are designed to run on the Web. ERP, although growing in popularity, is also being viewed with some skepticism.

ERP evolved from materials requirements planning (MRP), the information systems designed to improve manufacturing in general and assembly in particular. ERP systems now include manufacturing components and thus help with capacity planning, material production scheduling, and forecasting. Beyond manufacturing (and its service counterpart), ERP includes sales and operations planning, distribution, procurement, and managing the supply chain. It therefore significantly affects all the areas in the organization, including accounting, finance, management, marketing, and information systems.

Implementing an ERP solution may be frustrating because it is difficult to analyze a system currently in use and then fit the ERP model to that system. Furthermore, companies tend to design their business processes before ERP is implemented. Unfortunately, this process is often rushed and the proposed business model does not always match the ERP functionality. The result is further customizations, extended implementation time frames, higher costs, and often the loss

of user confidence. Analysts need to be aware of the magnitude of the problem they are tackling when trying to implement ERP packages.

DEPICTING SYSTEMS GRAPHICALLY

A system or subsystem as it exists within the corporate organization may be graphically depicted in several ways. The various graphical models show the boundaries of the system and the information used in the system.

Systems and the Context-Level Data Flow Diagram

The first model is the context-level data flow diagram (also called an environmental model). Data flow diagrams focus on the data flowing into and out of the system and the processing of the data. These basic components of every computer program can be described in detail and used to analyze the system for accuracy and completeness.

As shown in Figure 2.4, the context-level data flow diagram employs only three symbols: (1) a rectangle with rounded corners, (2) a square with two shaded edges, and (3) an arrow. Processes transform incoming data into outgoing information, and the content level has only one process, representing the entire system. The external entity represents any entity that supplies or receives information from the system but is not a part of the system. This entity may be a person, a group of people, a corporate position or department, or other systems. The lines that connect the external entities to the process are called data flows, and they represent data.

An example of a context-level data flow diagram is found in Figure 2.5. In this example, the most basic elements of an airline reservation system are represented. The passenger (an entity) initiates a travel request (data flow). The context-level diagram doesn't show enough detail to indicate exactly what happens (it isn't supposed to), but we can see that the passenger's preferences and the available flights are sent to the travel agent, who sends ticketing information back to the process. We can also see that the passenger reservation is sent to the airline. The context-level data flow diagram serves as a good starting point for drawing the use case diagram (discussed later in this chapter).

In Chapter 7 we see that a data flow contains much information. For example, the passenger reservation contains the passenger's name, airline, flight number(s), date(s) of travel, price, seating preference, and so on. For now, however, we are concerned mainly with how a context level defines the boundaries of the system. In the preceding example, only reservations are part of the process. Other decisions that the airline would make (for example, purchasing airplanes, changing schedules, pricing) are not part of this system.

The context-level data flow diagram is one way to show the scope of the system, or what is to be included in the system. The external entities are outside of the scope and something over which the system has no control.

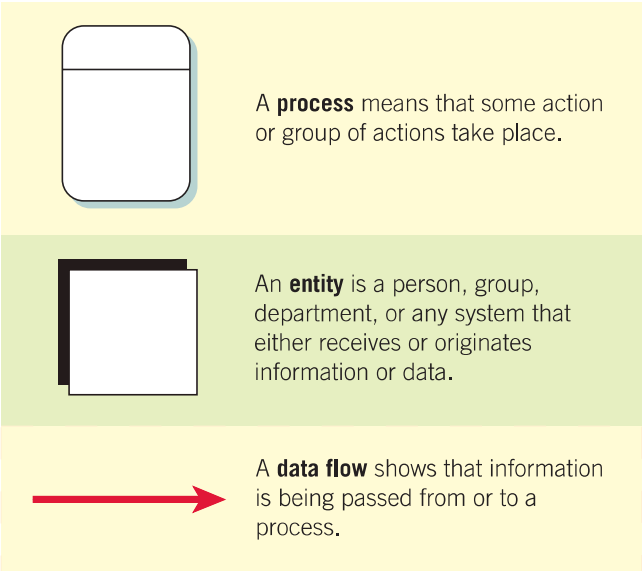


FIGURE 2.4

The basic symbols of a data flow diagram.

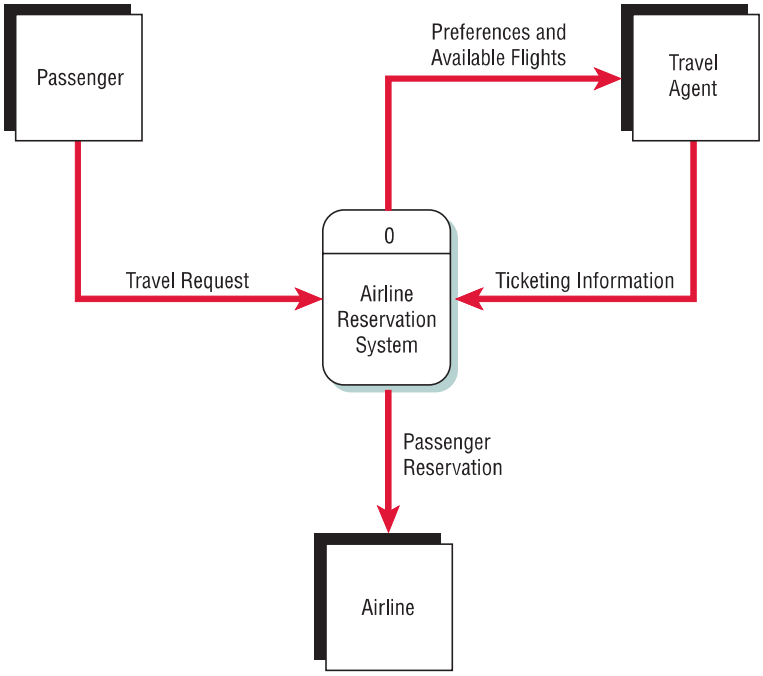


FIGURE 2.5
A context-level data flow diagram for an airline reservation system.

Systems and the Entity-Relationship Model

Another way a systems analyst can show the scope of the system and define proper system boundaries is to use an entity-relationship model. The elements that make up an organizational system can be referred to as entities. An entity may be a person, a place, or a thing, such as a passenger on an airline, a destination, or a plane. Alternatively, an entity may be an event, such as the end of the month, a sales period, or a machine breakdown. A relationship is the association that describes the interaction among the entities.

There are many different conventions for drawing entity-relationship (E-R) diagrams (with names like crow's foot, Arrow, or Bachman notation). In this book, we use crow's foot notation. For now, we assume that an entity is a plain rectangular box.

Figure 2.6 shows a simple entity-relationship diagram. Two entities are linked together by a line. In this example, the end of the line is marked with two short parallel marks (| |), signifying that this relationship is one-to-one. Thus, exactly one employee is assigned to one phone extension. No one shares the same phone extension in this office.

The red arrows are not part of the entity-relationship diagram. They are present to demonstrate how to read the entity-relationship diagram. The phrase on the right side of the line is read from top to bottom as follows: "One EMPLOYEE is assigned to one PHONE EXTENSION." On

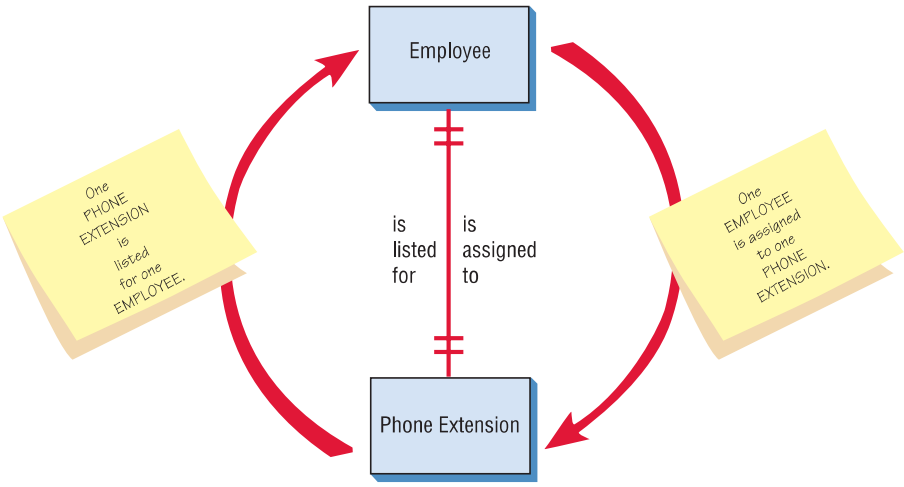


FIGURE 2.6
An entity-relationship diagram showing a one-to-one relationship.

the left side, as you read from bottom to top, the arrow says, "One PHONE EXTENSION is listed for one EMPLOYEE."

Similarly, Figure 2.7 shows another relationship. The crow's foot notation (≻+) is obvious on this diagram, and this particular example is a many-to-one example. As you read from left to right, the arrow signifies, "Many EMPLOYEES are members of a DEPARTMENT." As you read from right to left, it implies, "One DEPARTMENT contains many EMPLOYEES."

Notice that when a many-to-one relationship is present, the grammar changes from "is" to "are" even though the singular "is" is written on the line. The crow's foot and the single mark do not literally mean that this end of the relationship must be a mandatory "many." Instead, they imply that this end could be anything from one to many.

Figure 2.8 elaborates on this scheme. Here we have listed a number of typical entity relationships. The first, "An EMPLOYEE is assigned to an OFFICE," is a one-to-one relationship. The second one is a one-to-many relationship: "One CARGO AIRCRAFT will serve one or more DISTRIBUTION CENTERs." The third one is slightly different because it has a circle at one end. It can be read as "A SYSTEMS ANALYST may be assigned to MANY PROJECTS," meaning that the analyst can be assigned to no projects [that is what the circle (O), for zero, is for], one, or many projects. Likewise, the circle (O) indicates that none is possible in the next relationship. Recall that the short mark means one. Therefore, we can read it as follows: "A MACHINE may or may not be undergoing SCHEDULED MAINTENANCE." Notice that the line is written as "is undergoing," but the end marks on the line indicate that either no maintenance (O) or maintenance (I) is actually going on.

The next relationship states, "One or many SALESPEOPLE (plural of SALESPERSON) are assigned to one or more CUSTOMERs." It is the classic many-to-many relationship. The next relationship can be read as follows: "The HOME OFFICE can have one or many EMPLOYEEs," or "One or more EMPLOYEEs may or may not be assigned to the HOME OFFICE." Once again, the I and O together imply a Boolean situation; in other words, one or zero.

The final relationship shown here can be read as, "Many PASSENGERs are flying to many DESTINATIONs." This symbol [≻+] is preferred by some to indicate a mandatory "many" condition. (Would it ever be possible to have only one passenger or only one destination?) Even so, some CASE tools such as Visible Analyst do not offer this possibility, because the optional one-or-many condition as shown in the SALESPERSON-CUSTOMER relationship will do.

Up to now we have modeled all our relationships using just one simple rectangle and a line. This method works well when we are examining the relationships of real things such as real people, places, and things. Sometimes, though, we create new items in the process of developing an information system. Some examples are invoices, receipts, files, and databases. When we want

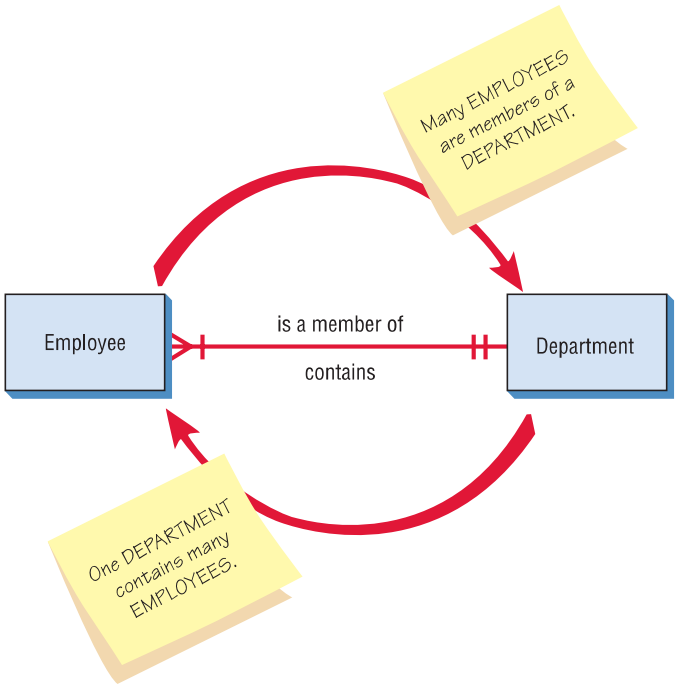
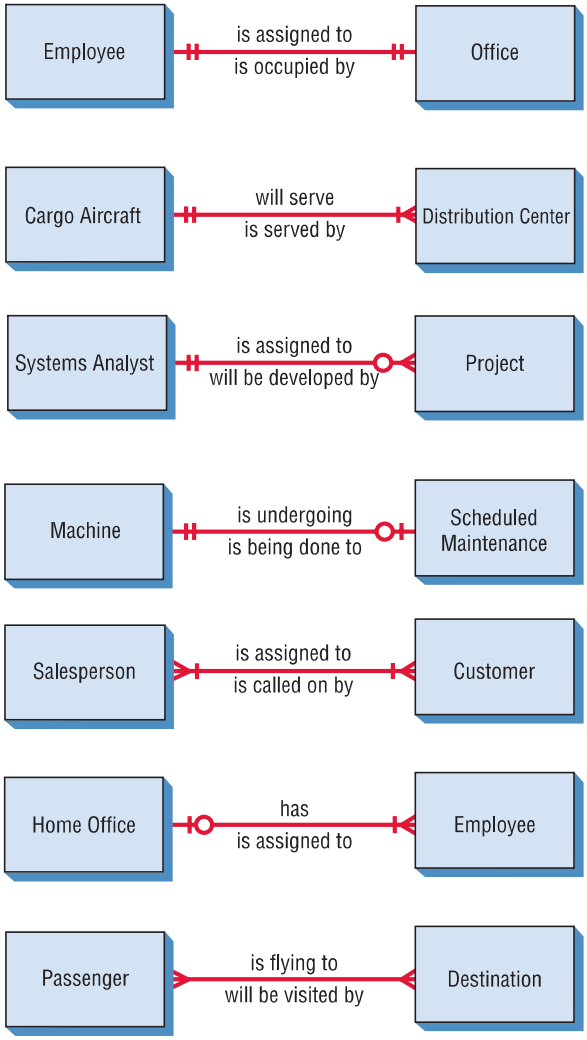


FIGURE 2.7

An entity-relationship diagram showing a many-to-one relationship.

FIGURE 2.8

Examples of different types of relationships in E-R diagrams.



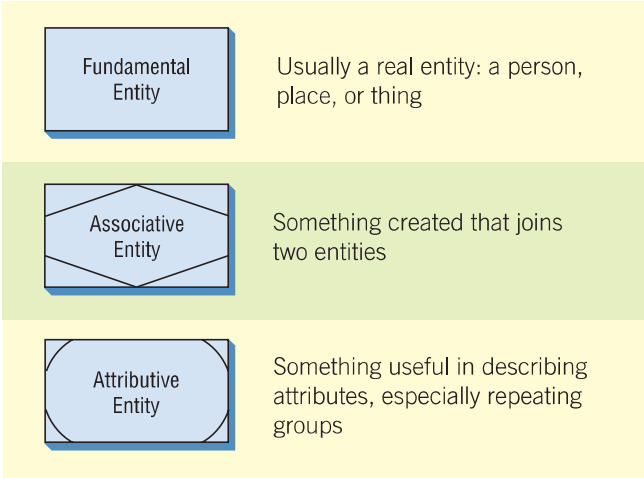
to describe how a person relates to a receipt, for example, it becomes convenient to indicate the receipt in a different way, as shown in Figure 2.9 as an associative entity.

An associative entity can only exist if it is connected to at least two other entities. For that reason, some call it a gerund, a junction, an intersection, or a concatenated entity. This wording makes sense because a receipt wouldn't be necessary unless there were a customer and a salesperson making the transaction.

Another type of entity is the attributive. When an analyst wants to show data that are completely dependent on the existence of a fundamental entity, an attributive entity should be used.

FIGURE 2.9

Three different types of entities used in E-R diagrams.



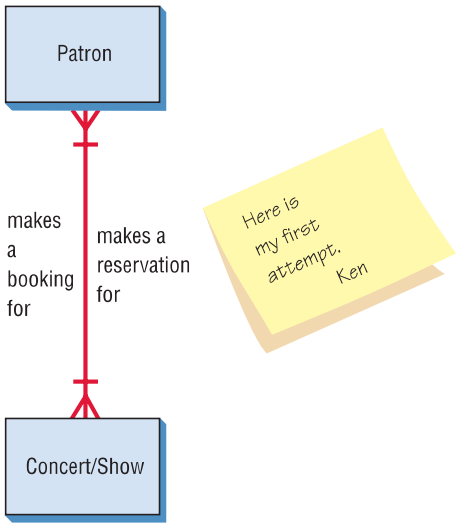


FIGURE 2.10

The first attempt at drawing an E-R diagram.

For example, when a library had multiple copies of the same book, an attributive entity could be used to designate which copy of the book is being checked out. The attributive entity is useful for showing repeating groups of data. For example, suppose we are going to model the relationships that exist when a patron gets tickets to a concert or show. The entities seem obvious at first: "a PATRON and a CONCERT/SHOW," as shown in Figure 2.10. What sort of relationship exists? At first glance the PATRON gets a reservation for a CONCERT/SHOW, and the CONCERT/SHOW can be said to have made a booking for a PATRON.

The process isn't that simple, of course, and the E-R diagram need not be that simple either. The PATRON actually makes a RESERVATION, as shown in Figure 2.11. The RESERVATION is for a CONCERT/SHOW. The CONCERT/SHOW holds the RESERVATION, and the RESERVATION is in the name of the PATRON. We added an associative entity here because a RESERVATION was created due to the information system required to relate the PATRON and the CONCERT/SHOW.

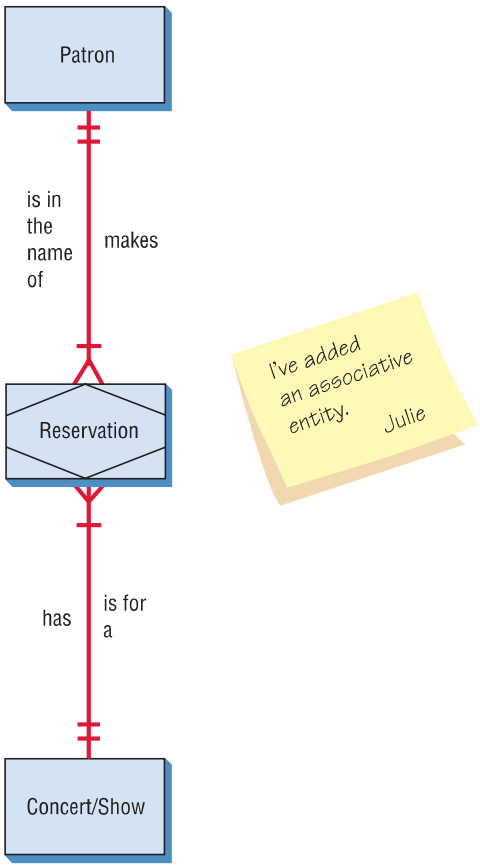


FIGURE 2.11

Improving the E-R diagram by adding an associative entry called RESERVATION.

Again this process is quite simple, but because concerts and shows have many performances, the E-R diagram is drawn once more in Figure 2.12. Here we add an attributive entity to handle the many performances of the CONCERT/SHOW. In this case the RESERVATION is made for a particular PERFORMANCE, and the PERFORMANCE is one of many that belong to a specific CONCERT/SHOW. In turn the CONCERT/SHOW has many performances, and one PERFORMANCE has a RESERVATION that is in the name of a particular PATRON.

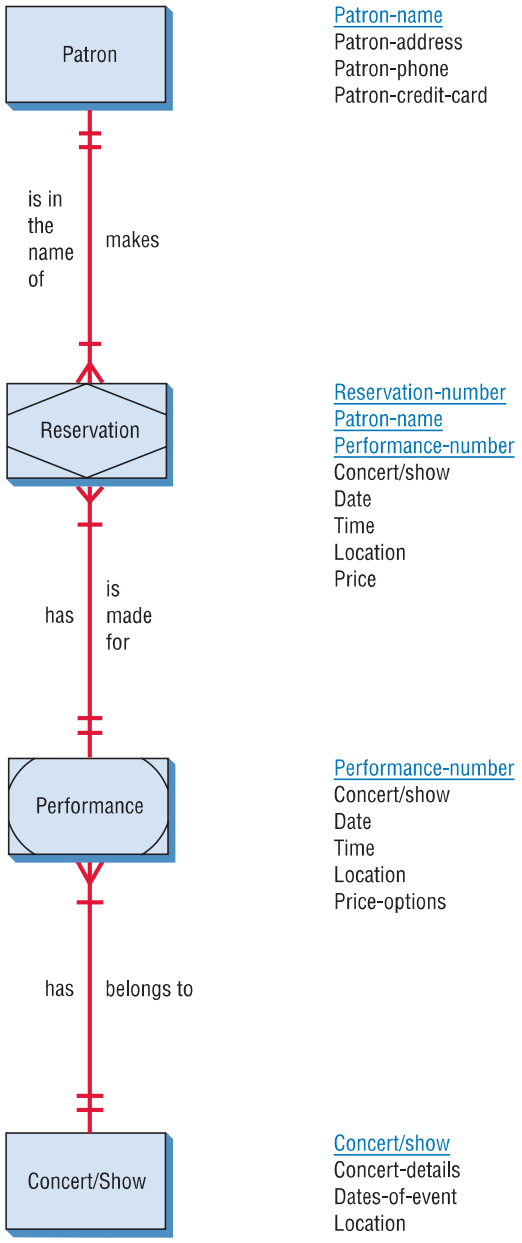
To the right of this E-R diagram is a set of data attributes that make up each of the entities. Some entities may have attributes in common. The attributes that are underlined can be searched for. The attributes are referred to as keys and are discussed in Chapter 13.

E-R diagrams are often used by systems designers to help model the file or database. It is even more important, however, that the systems analyst understand early both the entities and relationships in the organizational system. In sketching out some basic E-R diagrams, the analyst needs to:

1. List the entities in the organization to gain a better understanding of the organization.
2. Choose key entities to narrow the scope of the problem to a manageable and meaningful dimension.

FIGURE 2.12

A more complete E-R diagram showing data attributes of the entities.



MAC APPEAL

Microsoft Visio makes it easy for a systems analyst to draw E-R diagrams as well as most of the diagrams found in this book, but it is available only for PCs. Mac users have an alternative, OmniGraffle Professional. OmniGraffle is easier to use than Microsoft Visio because its drag-and-drop interface is smoother and more intuitive.

It also features a "smart guide" that uses pop-up distance markers to help position the symbols in the correct places. Many symbols like those used in E-R diagrams are built in, but OmniGraffle also lets the user search a third-party library called Graffletopia to find UML and other specialized symbols.

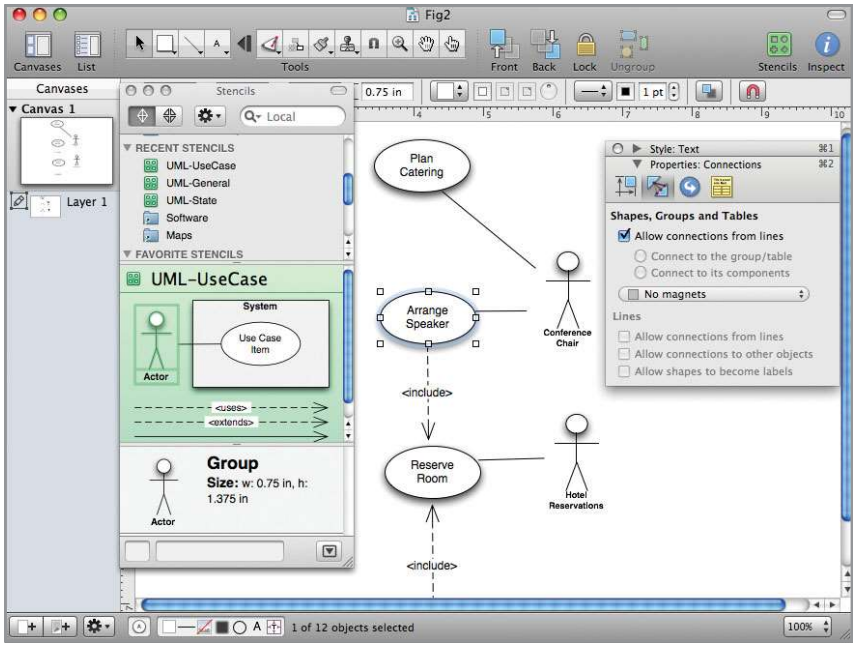


FIGURE 2.MAC

OmniGraffle from The Omni Group is an easy and powerful drawing package.

3. Identify what the primary entity should be.
4. Confirm the results of steps 1 through 3 through other data-gathering methods (investigation, interviewing, administering questionnaires, observation, and prototyping), as discussed in Chapters 4 through 6.

It is critical that the systems analyst begin to draw E-R diagrams upon entering the organization rather than waiting until the database needs to be designed, because E-R diagrams help the analyst understand what business the organization is actually in, determine the size and scope of the problem, and discern whether the right problem is being addressed. The E-R diagrams need to be confirmed or revised as the data-gathering process takes place.

USE CASE MODELING

Originally introduced as a diagram for use in object-oriented UML, use cases are now being used regardless of the approach to systems development. It can be used as part of the SDLC or in agile modeling. The word *use* is pronounced as a noun (yoos) rather than a verb (yooz). A

use case model describes *what* a system does without describing *how* the system does it; that is, it is a logical model of the system. (Logical or conceptual models will be further discussed in Chapter 7.) The use case model reflects the view of the system from the perspective of a user outside of the system (i.e., the system requirements).

An analyst develops use cases in a cooperative effort with the business experts who help define the requirements of the system. The use case model provides an effective means of communication between the business team and the development team. A use case model partitions the way the system works into behaviors, services, and responses (the use cases) that are significant to the users of the system.

From the perspective of an actor (or user), a use case should produce something that is of value. Therefore, the analyst must determine what is important to the user, and remember to include it in the use case diagram. For example, is entering a password something of value to the user? It may be included if the user has a concern about security or if it is critical to the success of the project.

Use Case Symbols

A use case diagram contains the actor and use case symbols, along with connecting lines. Actors are similar to external entities; they exist outside of the system. The term *actor* refers to a particular role of a user of the system. For example, an actor may be an employee, but also may be a customer at the company store. Even though it is the same person in the real world, it is represented as two different symbols on a use case diagram, because the person interacts with the system in different roles. The actor exists outside of the system and interacts with the system in a specific way. An actor can be a human, another system, or a device such as a keyboard or Web connection. Actors can initiate an instance of a use case. An actor may interact with one or more use cases, and a use case may involve one or more actors.

Actors may be divided into two groups. Primary actors supply data or receive information from the system. Some users directly interact with the system (system actors), but primary actors may also be businesspeople who do not directly interact with the system but have a stake in it. Primary actors are important because they are the people who use the system and can provide details on what the use case should do. They can also provide a list of goals and priorities. Supporting actors (also called secondary actors) help to keep the system running or provide other services. These are the people who run the help desk, the analysts, programmers, and so on.

Sometimes it is useful to create an actor profile that lists the actors, their background, and their skills in a simple table format. This may be useful to understand how the actor interacts with the system. An example is an Order Processing Specialist. The profile would be, "A routine user of the software, familiar with minor features, order exceptions, and order customization." It is also useful to list the actors and their goals and priorities. Each goal may become a use case.

A use case provides developers with a view of what the users want. It is free of technical or implementation details. We can think of a use case as a sequence of transactions in a system. The use case model is based on the interactions and relationships of individual use cases.

A use case always describes three things: an actor that initiates an event; the event that triggers a use case; and the use case that performs the actions triggered by the event. In a use case, an actor using the system initiates an event that begins a related series of interactions in the system. Use cases are used to document a single transaction or event. An event is an input to the system that happens at a specific time and place and causes the system to do something.

It is better to create fewer use cases rather than more. Often queries and reports are not included; 20 use cases (and no more than 40 or 50) are sufficient for a large system. Use cases may also be nested, if needed. Some use cases use the verb *manage* to group use cases for adding, deleting, and changing into another, lower-level, use case diagram. You can include a use case on several diagrams, but the actual use case is defined only once in the repository. A use case is named with a verb and a noun.

Use Case Relationships

Active relationships are referred to as behavioral relationships and are used primarily in use case diagrams. There are four basic types of behavioral relationships: communicates, includes, extends, and generalizes. Notice that all these terms are action verbs. Figure 2.13 shows the arrows

Relationship	Symbol	Meaning
Communicates		An actor is connected to a use case using a line with no arrowheads.
Includes	<< include >>	A use case contains a behavior that is common to more than one other use case. The arrow points to the common use case.
Extends	<< extend >>	A different use case handles exceptions from the basic use case. The arrow points from the extended to the basic use case.
Generalizes		One UML "thing" is more general than another "thing." The arrow points to the general "thing."

FIGURE 2.13

Some components of use case diagrams showing actors, use cases, and relationships for a student enrollment example.

and lines used to diagram each of the four types of behavioral relationships. The four relationships are described next.

COMMUNICATES. The behavioral relationship communicates is used to connect an actor to a use case. Remember that the task of the use case is to give some sort of result that is beneficial to the actor in the system. Therefore, it is important to document these relationships between actors and use cases. In our first example, a **Student** communicates with **Enroll in Course**. Examples of some components of a student enrollment example are shown in the use case diagrams in Figure 2.14.

INCLUDES. The includes relationship (also called uses relationship) describes the situation in which a use case contains behavior that is common to more than one use case. In other words, the common use case is included in the other use cases. A dotted arrow that points to the common use case indicates the includes relationship. An example would be a use case **Pay Student Fees** that is included in **Enroll in Course** and **Arrange Housing**, because in both cases students must pay their fees. This may be used by several use cases. The arrow points toward the common use case.

EXTENDS. The extends relationship describes the situation in which one use case possesses the behavior that allows the new use case to handle a variation or exception from the basic use case.

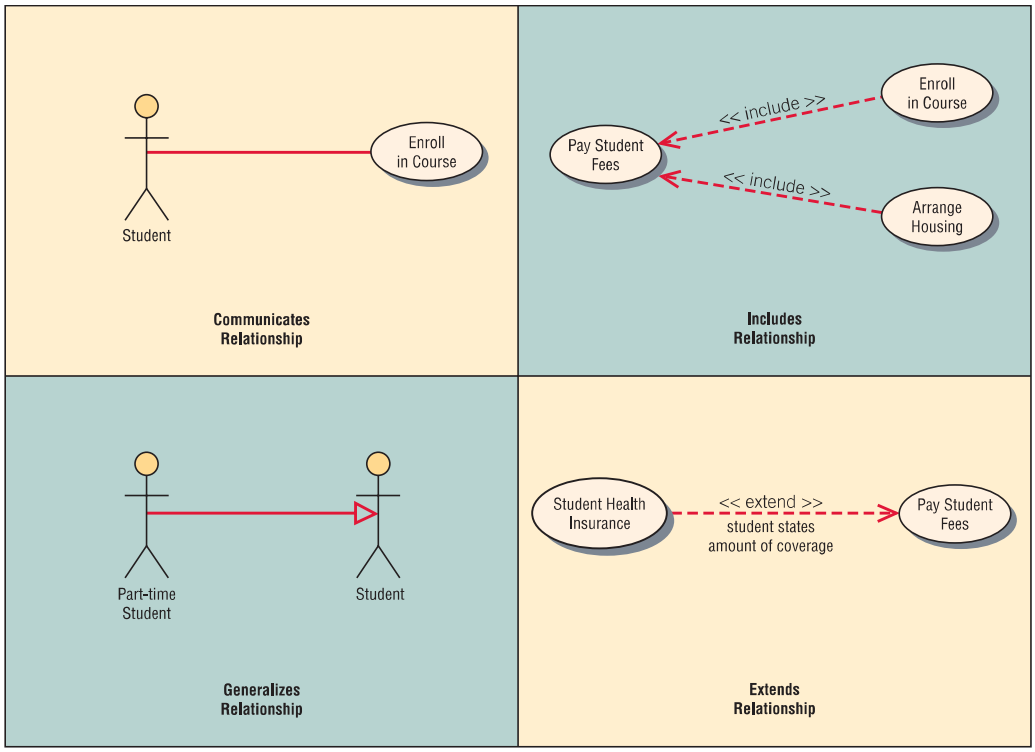


FIGURE 2.14

Examples of use cases and behavioral relationships for student enrollment.

For example, the extended use case **Student Health Insurance** extends the basic use case **Pay Student Fees**. The arrow goes from the extended to the basic use case.

GENERALIZES. The generalizes relationship implies that one thing is more typical than the other thing. This relationship may exist between two actors or two use cases. For example, a **Part-Time Student** generalizes a **Student**. Similarly, some of the university employees are professors. The arrow points to the general thing.

Developing System Scope

The scope of a system defines its boundaries, what is in scope—or inside the system—and what is out of scope. The project usually has a budget that helps to define scope, and a start and end time. Actors are always outside the scope of the system. The communicates lines that connect actors to the use cases are the boundaries, and define the scope. Since a use case diagram is created early in the systems life cycle, the budget, starting time, and ending time may change as the project progresses; as the analyst learns more about the system, the use case diagrams, use case, and scope may change.

Developing Use Case Diagrams

The primary use case consists of a standard flow of events in the system that describes a standard system behavior. The primary use case represents the normal, expected, and successful completion of the use case.

When diagramming a use case, start by asking the users to list everything the system should do for them. This can be done using interviews, in a joint application design session (as described in Chapter 4), or through other facilitated team sessions. The analyst may also use agile stories sessions (described in Chapter 6) to develop use cases. Write down who is involved with each use case, and the responsibilities or services the use case must provide to actors or other systems. In the initial phases, this may be a partial list that is expanded in the later analysis phases. Use the following guidelines:

1. Review the business specifications and identify the actors involved.
2. Identify the high-level events and develop the primary use cases that describe those events and how the actors initiate them. Carefully examine the roles played by the actors to identify all the possible primary use cases initiated by each actor. Use cases with little or no user interaction do not have to be shown.
3. Review each primary use case to determine the possible variations of flow through the use case. From this analysis, establish the alternative paths. Because the flow of events is usually different in each case, look for activities that could succeed or fail. Also look for any branches in the use case logic in which different outcomes are possible.

If a context-level data flow diagram has been created, it can be a starting point for creating a use case. The external entities are potential actors. Then examine the data flow to determine if it would initiate a use case or be produced by a use case.

Figure 2.15 is an example of a use case diagram representing a system used to plan a conference. The actors are the **Conference Chair**, responsible for planning and managing the conference, the conference **Participant, Speakers**, a **Keynote Speaker**, **Hotel Reservations**, and a **Caterer**. Actors represent the *role* the user plays, and the **Caterer** may be either a hotel employee or an external catering service.

Both the **Conference Chair** and the **Caterer** are involved in planning meals and banquets. The **Conference Chair** is also responsible for arranging speakers. The **Participant** registers for the conference. Notice that the **Reserve Room** use case is involved in an *includes* relationship with the **Arrange Speaker** and **Register for Conference** use cases, since both speakers and participants will need lodging. The **Arrange Language Translation** use case extends the **Register for Conference** use case because not all participants will require language translation services. The **Speaker** actor is a generalization of **Keynote Speaker**.

Developing Use Case Scenarios

Each use case has a description. We will refer to the description as a use case scenario. As mentioned, the primary use case represents the standard flow of events in the system, and alternative

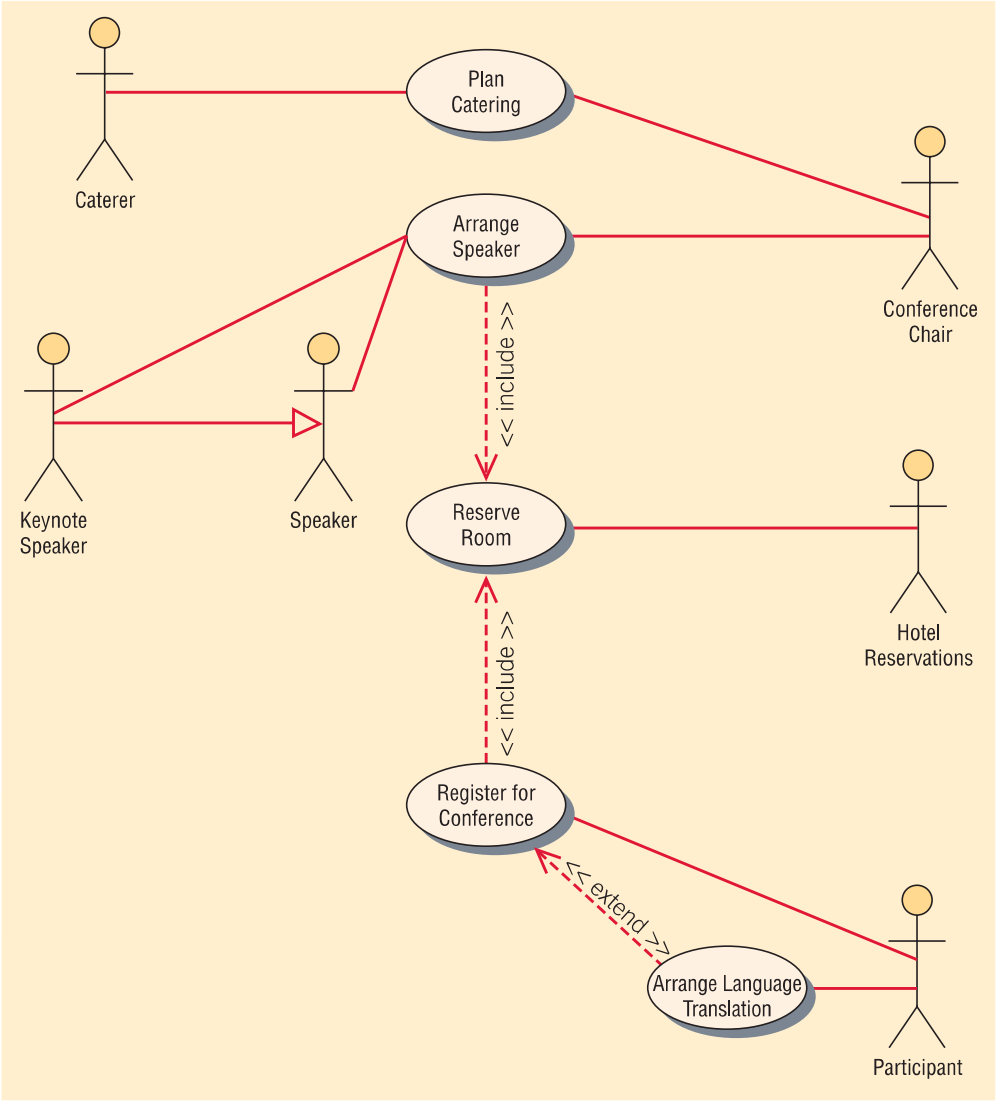


FIGURE 2.15

A use case diagram representing a system used to plan a conference.

paths describe variations to the behavior. Use case scenarios may describe what happens if an item purchased is out of stock, or if a credit card company rejects a customer's requested purchase.

There is no standardized use case scenario format, so each organization is faced with specifying what standards should be included. Often the use cases are documented using a use case document template predetermined by the organization, which makes the use cases easier to read and provides standardized information for each use case in the model.

Use Case Levels

You may want to create use cases for different levels. One method (defined by Alistair Cockburn) uses the following altitude metaphors:

1. White is the highest level, like clouds. This is the enterprise level, and there may only be four to five for the entire organization. Examples might be to advertise goods, sell goods to customers, manage inventory, manage the supply chain, and optimize shipping.
2. Kite is lower than white but still a high level, providing an overview. The kite use case may be at the business unit or department level and is a summary of goals. Examples would be to register students, or if working with a travel company: make an airline, hotel, car, or cruise reservation.
3. Blue is at sea level, and is usually created for user goals. This often has the greatest interest for users and is easiest for a business to understand. It is usually written for a business activity and each person should be able to do one blue level activity in anywhere from 2 to

20 minutes. Examples are register a continuing student, add a new customer, place an item in a shopping cart, and order checkout.

4. Indigo or fish is a use case that shows lots of detail, often at a functional or subfunctional level. Examples are choose a class, pay academic fees, look up the airport code for a given city, and produce a list of customers after entering a name.
5. Black or clam, like the bottom of the ocean, are the most detailed use cases, at a subfunction level. Examples might be a secure logon validation, adding a new field using dynamic HTML, or using Ajax to update a Web page in a small way.

A use case scenario example is shown in Figure 2.16. Some of the areas included are optional, and may not be used by all organizations. The three main areas are:

1. An area header containing case identifiers and initiators.
2. Steps performed.
3. A footer area containing preconditions, assumptions, questions, and other information.

FIGURE 2.16

A use case scenario is divided into three sections: identification and initiation; steps performed; and conditions, assumptions, and questions.

Use case name:	Register for Conference	**UniqueID:** Conf RG 003
Area:	Conference Planning	
Actor(s):	Participant	
Stakeholder:	Conference Sponsor, Conference Speakers	
Level:	Blue	
Description:	Allow conference participant to register online for the conference using a secure Web site.	
Triggering Event:	Participant uses Conference Registration Web site, enters userID and password, and clicks the logon button.	
Trigger type:	External Temporal	
Steps Performed (Main Path)		**Information for Steps**
1. Participant logs in using the secure Web server.		userID, Password
2. Participant record is read and password is verified.		Participant Record, userID, Password
3. Participant and session information is displayed on the Registration Web page.		Participant Record, Session Record
4. Participant enters information on the Registration Web form and clicks **Submit** button.		Registration Web Form
5. Registration information is validated on the Web server.		Registration Web Form
6. Registration Confirmation page is displayed to confirm registration information.		Confirmation Web Page
7. Credit card is charged for registration fees.		Secure Credit Card Web Page
8. Add Registration Journal record is written.		Confirmation Web Page
9. Registration record is updated on the Registration Master.		Confirmation Web Page, Registration Record
10. Session record is updated for each selected session on the Session Master.		Confirmation Web Page, Session Record
11. Participant record is updated for the participant on the Participant Master.		Confirmation Web Page, Participant Record
12. Successful Registration Confirmation Web page is sent to the participant.		Registration Record Confirmation Number
Preconditions:	Participant has already registered and has created a user account.	
Postconditions:	Participant has successfully registered for the conference.	
Assumptions:	Participant has a browser and a valid userID and password.	
Success Guarantee:	Participant has registered for the conference and is enrolled in all selected sessions.	
Minimum Guarantee:	Participant was able to logon.	
Requirements Met:	Allow conference participants to be able to register for the conference using a secure Web site.	
Outstanding Issues:	How should a rejected credit card be handled?	
Priority:	High	
Risk:	Medium	

The first area, use case identifiers and initiators, orients the reader and contains the use case name and a unique ID; the application area or system that this use case belongs to; the actors involved in the use case; and the stakeholders that have a high level of interest in the use case. Some stakeholders never interact directly with the system, such as the stockholders, the board of directors, or the sales manager. Each primary actor is a stakeholder, but not listed in the stakeholder area. Include the level (blue, kite, and so on) and a brief description of what the use case accomplishes.

The header concludes with the initiating (triggering) event, that is, what caused the use case to start, and the type of trigger, either external or temporal. External events are those started by an actor, either a person or another system requesting information, such as an airline reservation system requesting flight information from an airline system. Temporal events are those that are triggered or started by time. Events occur at a specific time, such as sending an email about special offers once a week on a Sunday evening, sending bills on a specific day, or generating government statistics on a specified date every quarter.

The second area of the use case includes the steps performed, and the information required for each of the steps. These statements represent the standard flow of events and the steps taken for the successful completion of the use case. It is desirable to write up a use case for the main path, and then to write up one for each of the alternative paths separately, rather than using IF . . . THEN . . . statements. Steps are numbered with an integer. The steps may come from a detailed interview with users or may be derived from agile modeling stories (as described in Chapter 6). These steps should be reviewed with the users for clarification.

The analyst should examine each of the steps and determine the information required for each step. If the analyst cannot determine the information, he or she should schedule a follow-up interview with the user. Some use case descriptions include extensions or alternative scenarios, with the exceptions as additional sections following the standard flow of events. These are numbered with an integer, decimal point, and another integer, such as 3.1, 3.2, 3.3, and so on. These are steps that may or may not be used. Analysts and users can brainstorm what can go wrong with the main path, and may uncover important details and conditions. It is necessary to work with the users to determine what to do when these conditions occur. This helps to detect errors earlier in the life cycle.

Figure 2.17 illustrates how logic and alternative scenarios can be included in the middle section of a use case. In this airline example, notice that step 1 is made up of smaller steps, many of which are preceded by an "if." These are still on the main path, but only occur if the condition is met. For example, if there are many airports that serve a city, then all the airports will be displayed. Extensions or alternate scenarios can also appear here. For this airline, other scenarios include flight selection, seat selection, and meal selection. Use cases may even include iterative or looping steps.

The third area of the use case includes:

- Preconditions, or the condition of the system before the use case may be performed, which may be another use case. An example might be, "The viewer has successfully logged into the system," or it might be the successful completion of another use case.
- Postconditions, or the state of the system after the use case has finished, including output people have received, transmissions to other systems, and data that have been created or updated. These relate to the goals or user requirements from a problem definition (described in Chapter 3) or to agile stories (described in Chapter 6).
- Assumptions made that would affect the method of the use case and that could stipulate required technology, such as the minimum technology requirements in a browser or even a specific or higher version of a browser. An assumption might be that cookies or JavaScript are enabled. The analyst must determine what to do if the assumptions are not met. When using Google Maps, JavaScript must be enabled. If it is not enabled, the map will not display. Cookies are required by Netflix. Good Web pages will detect that an assumption has not been met and notify the viewer with a message, including information on how to turn on cookies or JavaScript for different browsers.
- Minimal guarantee is the minimum promised to the users. They may not be happy with this result and it may be that nothing happens.
- Success guarantee is what would satisfy the users, and it is usually that the goal of the use case has been met.
- Any outstanding issues or questions must be answered before implementation of the use case.

Steps Performed (Main Path)	Information for Steps
1. Enter departing and arriving airports, dates of travel.	Airport Locations
1.1. If an airport code is entered, display matching name, city, country	
1.2. If a city is entered, find all matching cities	
1.3. Customer selects a city	
1.4. If there is more than one airport for the city, display airports	
1.5. Client selects an airport	
1.6. Insert the airport code (3 characters)	
1.7. Display the matching airport country, city, and airport name	
2. Find all matching flights with available seats	Flight Information
3. Customer selects flight	
4. Customer logs on	Customer Logon
5. Customer selects passenger names	Passenger Records
6. Seating chart is displayed showing all available seats	Plane Number, Seating Chart, Available Seats
7. Customer selects seat(s) for each passenger	
8. Display confirmation and credit card page	
9. Credit card verified	
10. Email confirmation sent	
11. Airline reservation made	
Extensions or Alternative Scenarios	
Flight Selection 1. A list of flights displays 2. Customer selects a flight 3. Request is sent to airline 4. Flight is already full	
Seat Selection 1. A list of flights displays 2. Customer selects a flight 3. Request is sent to airline 4. Seat reservations are retrieved 5. Seating chart is displayed 6. Customer cannot find an acceptable seat	
Meal Selection for International Flights	
1. Customer selects meal from drop-down list	Available Airline Meal List
2. Record is updated with meal selection	Customer Meal Record

FIGURE 2.17

Use cases can include conditional steps as well as extensions or alternative scenarios.

- An optional statement of priority of the use case, which may come from a problem definition or user requirements.
- An optional statement of risk involved in creating the use case.

The "requirements met" area links the use case to user requirements or objectives from a problem definition. Once you develop the use case scenarios, be sure to review your results with the business experts to verify and refine the use cases if needed.

In this particular use case scenario, called **Register for Conference,** the only actor involved is the **Participant.** The overall area is **Conference Planning,** and the use case is triggered by the participant logging on to the **Registration Web page.** The **Steps Performed** area lists the sequence of events that must occur for a successful conference registration. Notice that the information needed to perform each of the steps is listed on the right. This may include Web pages and forms, as well as database tables and records.

The **Preconditions** area in the footer section of the use case scenario lists what must occur before the participant can register for a conference. In this example, the participant must have already signed up as a member of the society and have a valid userID and password. The **Postconditions** area lists what has been accomplished by the use case. The **Assumptions** area lists any basic premises the analyst assumes are fulfilled by the actor beforehand. The **Requirements Met** area shows why this use case is important and necessary for the business area to be successful. **Priority** is an indication of which use cases should be developed first and which may be delayed. **Risk** is a rough assessment of whether there may be problems or difficulties developing the use case. In this case, the risk is medium because the registration use case requires a secure server and is accepting credit card information.

Creating Use Case Descriptions

Use the following four steps to create use case descriptions:

1. Use agile stories, problem definition objectives, user requirements, or a features list as a starting point.
2. Ask about the tasks that must be done to accomplish the transaction. Ask if the use case reads any data or updates any tables.
3. Find out if there are any iterative or looping actions.
4. The use case ends when the customer goal is complete.

Why Use Case Diagrams Are Helpful

No matter what method you use to develop your system (traditional SDLC methods, agile methods, or object-oriented methods), you will find that use cases are very valuable. The use case diagrams identify all the actors in the problem domain, and a systems analyst can concentrate on what humans want and need to use the system, extend their capabilities, and enjoy their interaction with technology.

The actions that need to be completed are also clearly shown on the use case diagram. This not only makes it easy for the analyst to identify processes, but it also aids in communication with other analysts on the team and business executives.

The use case scenario is also worthwhile. Since a lot of the information the users impart to the analyst already takes the form of stories, it is easy to capture the stories on a use case scenario form. The use case scenario always documents the triggering event so that an analyst can always trace the steps that led to other use cases. Since the steps performed are noted, it is possible to employ use case scenarios to write logical processes.

Use case diagrams are becoming popular because of their simplicity and lack of technical detail. They are used to show the scope of a system, along with the major features of the system and the actors who work with those major features. The users see the system and they can react to it and provide feedback. They may also help to determine whether to build or buy the software.

The main reasons for writing use cases are shown in Figure 2.18.

LEVELS OF MANAGEMENT

Management in organizations exists on three broad, horizontal levels: operational control, managerial planning and control (middle management), and strategic management, as shown in

- Use cases effectively communicate systems requirements because the diagrams are kept simple.
- Use cases allow people to tell stories.
- Use case stories make sense to nontechnical people.
- Use cases do not depend on a special language.
- Use cases can describe most functional requirements (such as interactions between actors and applications).
- Use cases can describe nonfunctional requirements (such as performance and maintainability) through the use of stereotypes.
- Use cases help analysts define boundaries.
- Use cases can be traceable, allowing analysts to identify links between use cases and other design and documentation tools.

FIGURE 2.18

The main reasons for writing use cases are their effectiveness in communicating with users and their capturing of user stories.

CONSULTING OPPORTUNITY 2.2

Where There's Carbon, There's a Copy

"I don't know what we do with the pink ones yet," Richard Russell admitted. "They're part of a quadruplicate form that rips apart. All I know is that we keep them for the filing clerk, and he files them when he has time."

Richard is a newly hired junior account executive for Carbon, Carbon & Rippy, a brokerage house. You are walking through the steps he takes in making a stock purchase "official" because his boss has asked you to streamline the process whereby stock purchase information is stored in the computer system and retrieved.

After you leave, Richard continues thinking about the pink forms. He tells his clerk, Harry Schultz, "In my two months here, I haven't seen anyone use those. They take up my time and yours, not to mention all the filing space. Let's pitch them."

Richard and Harry proceed to open all the old files kept by Richard's predecessor and throw out the filed pink forms, along with those accumulated but not yet filed. It takes hours, but they make a lot of room. "Definitely worth the time," Richard reassures Harry.

Three weeks later, an assistant to Richard's boss, Carol Vaness, appears. Richard is happy to see a familiar face, greeting her with, "Hi, Carol. What's new?"

"Same old thing," Carol sighs. "Well, I guess it isn't old to you, because you're the newcomer. But I need all those pesky pink forms."

Almost in shock, Richard exchanges looks with Harry, then mumbles, "You're kidding, of course."

Carol looks more serious than Richard ever thought possible, replying, "No joke. I summarize all the pink forms from all the brokers, and then my totals are compared with computerized stock purchase information. It's part of our routine, three-month audit for transaction accuracy. My work depends on yours. Didn't Ms. McCue explain that to you when you started?"

What systems concept did Richard and Harry ignore when tossing out the pink forms? What are the possible ramifications for systems analysts if general systems concepts are ignored?

Figure 2.19. Each level carries its own responsibilities, and all work toward achieving organizational goals and objectives in their own ways.

Operational control forms the bottom tier of three-tiered management. Operations managers make decisions using predetermined rules that have predictable outcomes when implemented correctly.

They make decisions that affect implementation in work scheduling, inventory control, shipping, receiving, and control of processes such as production. Operations managers oversee the operating details of the organization.

Middle management forms the second, or intermediate, tier of the three-tiered management system. Middle managers make short-term planning and control decisions about how resources may best be allocated to meet organizational objectives.

Their decisions range all the way from forecasting future resource requirements to solving employee problems that threaten productivity. The decision-making domain of middle managers can usefully be characterized as partly operational and partly strategic, with constant fluctuations.

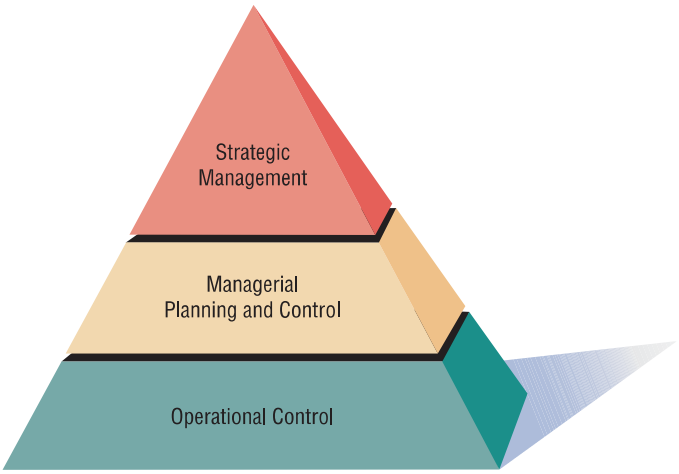


FIGURE 2.19

Management in organizations exists on three horizontal levels: operational control, managerial planning and control, and strategic management.

Strategic management is the third level of three-tiered management control. Strategic managers look outward from the organization to the future, making decisions that will guide middle and operations managers in the months and years ahead.

Strategic managers work in a highly uncertain decision-making environment. Through statements of goals and the determination of strategies and policies to achieve them, strategic managers actually define the organization as a whole. Theirs is the broad picture, wherein the company decides to develop new product lines, divest itself of unprofitable ventures, acquire other compatible companies, or even allow itself to be acquired or merged.

There are sharp contrasts among the decision makers on many dimensions. For instance, strategic managers have multiple decision objectives, whereas operations managers have single ones. It is often difficult for high-level managers to identify problems, but it is easy for operations managers to do so. Strategic managers are faced with semistructured problems, whereas lower-level managers deal mostly with structured problems.

The alternative solutions to a problem facing the strategic managers are often difficult to articulate, but the alternatives that operations managers work with are usually easy to enumerate. Strategic managers most often make one-time decisions, whereas the decisions made by operations managers tend to be repetitive.

Implications for Information Systems Development

Each of the three management levels holds differing implications for developing information systems. Some of the information requirements for managers are clear-cut, whereas others are fuzzy and overlapping.

Operations managers need internal information that is of a repetitive, low-level nature. They are highly dependent on information that captures current performance, and they are large users of online, real-time information resources. The need of operations managers for past performance information and periodic information is only moderate. They have little use for external information that allows future projections.

On the next management level, middle managers are in need of both short- and longer-term information. Due to the troubleshooting nature of their jobs, middle managers experience extremely high needs for information in real time. To control properly, they also need current information on performance as measured against set standards. Middle managers are highly dependent on internal information. In contrast to operations managers, they have a high need for historical information, along with information that allows for the prediction of future events and simulation of numerous possible scenarios.

Strategic managers differ somewhat from both middle and operations managers in their information requirements. They are highly dependent on information from external sources that supply news of market trends and the strategies of competing corporations. Because the task of strategic managing demands projections into the uncertain future, strategic managers have a high need for information of a predictive nature and information that allows creation of many different what-if scenarios. Strategic managers also exhibit strong needs for periodically reported information as they seek to adapt to fast-moving changes.

ORGANIZATIONAL CULTURE

Organizational culture is an established area of research that has grown remarkably in the last generation. Just as it is appropriate to think of organizations as including many technologies, it is similarly appropriate to see them as hosts to multiple, often competing subcultures.

There is still little agreement on what precisely constitutes an organizational subculture. It is agreed, however, that competing subcultures may be in conflict, attempting to gain adherents to their vision of what the organization should be. Research is in progress to determine the effects of virtual organizations and virtual teams on the creation of subcultures when members do not share a physical workspace but share tasks.

Rather than thinking about culture as a whole, it is more useful to think about the researchable determinants of subcultures, such as shared verbal and nonverbal symbolism. Verbal symbolism includes shared language used to construct, convey, and preserve subcultural myths, metaphors, visions, and humor. Nonverbal symbolism includes shared artifacts, rites, and ceremonies; clothing

CONSULTING OPPORTUNITY 2.3

Pyramid Power

"We really look up to you," says Paul LeGon. As a systems analyst, you have been invited to help Pyramid, Inc., a small, independent book-publishing firm that specializes in paperback books outside of the publishing mainstream.

Paul continues, "We deal with what some folks think are fringe topics. You know, pyramid power, end-of-the-world prophecies, and healthier living by thinking of the color pink. Sometimes when people see our books, they just shake their heads and say, 'Tut—uncommon topic.' But we're not slaves to any particular philosophy, and we've been very successful. So much so that because I'm 24, people call me the 'boy king.'" Paul pauses to decipher your reaction.

Paul continues, "I'm at the top as president, and functional areas such as editorial, accounting, production, and marketing are under me."

Paul's assistant, Ceil Toom, who has been listening quietly up to now, barges in with her comments: "The last systems experts that did a project for us recommended the creation of liaison committees of employees between accounting, production, and marketing, so that we could share newly computerized inventory and sales figures across the organization. They claimed that committees such as that would cut down on needless duplication of output, and each functional area would be better integrated with all the rest."

Paul picks up the story, saying, "It was fair—oh, for a while—and the employees shared information, but the reason you're here is that the employees said they didn't have time for committee meetings and were uncomfortable sharing information with people from other departments who were further up the ladder than they were here at Pyramid."

According to Paul and Ceil, what were the effects of installing a management information system at Pyramid, Inc., that required people to share information in ways that were not consistent with their structure? Propose some general ways to resolve this problem so that Pyramid employees can still obtain the sales and inventory figures they need.

of decision makers and workers; the use, placement, and decoration of offices; and rituals for celebrating members' birthdays, promotions, and retirements.

Subcultures coexist within "official" organizational cultures. The officially sanctioned culture may prescribe a dress code, suitable ways to address superiors and coworkers, and proper ways to deal with the outside public. Subcultures may be powerful determinants of information requirements, availability, and use.

Organizational members may belong to one or more subcultures in the organization. Subcultures may exert a powerful influence on member behavior, including sanctions for or against the use of information systems.

Understanding and recognizing predominant organizational subcultures may help the systems analyst overcome the resistance to change that arises when a new information system is installed. For example, the analyst might devise user training to address specific concerns of organizational subcultures. Identifying subcultures may also help in the design of decision support systems that are tailored for interaction with specific user groups.

SUMMARY

There are three broad organizational fundamentals to consider when analyzing and designing information systems: the concept of organizations as systems, the various levels of management, and the overall organizational culture.

Organizations are complex systems composed of interrelated and interdependent subsystems. In addition, systems and subsystems are characterized by their internal environments on a continuum from open to closed. An open system allows free passage of resources (people, information, materials) through its boundaries; closed systems do not permit free flow of input or output. Organizations and teams can also be organized virtually with remote members connected electronically who are not in the same physical workspace. Enterprise resource planning systems are integrated organizational (enterprise) information systems developed with customized, proprietary software that help the flow of information between the functional areas in the organization. They support a systems view of the organization.

HYPERCASE® EXPERIENCE 2

"You seem to have already made a good start at MRE. I'm glad you met Snowden Evans. As you know, you'll be reporting directly to him during your consulting project. As his administrative assistant for the last five years I can tell you a lot about the company, but remember that there are a number of ways to find out more. You will want to interview users, observe their decision-making settings, and look at archival reports, charts, and diagrams. To do so, you can click on the phone directory to get an appointment with an interviewee, click on the building map to view the layout of the building, or click on the corporate Web site to see the functional areas and formal hierarchical relationships at MRE.

"Many of the rules of corporate life apply in the MRE HyperCase. You can walk freely in many public areas. If you want to tour a private office, however, you must first book an appointment with one of our employees. Some secure areas are strictly off limits to you as an outsider since you could pose a security risk.

"I don't think you'll find us excessively secretive, however, because you may assume that any employee who grants you an interview will also grant you access to the archival material in his or her files as well as to current work on their desktops or screens.

"Unfortunately, some people in the company never seem to make themselves available to consultants. I suggest you be persistent. There are lots of ways to find out about the people and the systems of MRE. Creativity pays off. You'll notice that the systems consultants who follow their hunches, sharpen their technical skills, and never stop thinking about piecing together the puzzles at MRE are the ones who get the best results.

"Remember to use multiple methods—interviewing, observation, and investigation—to understand what we at MRE are trying to tell you. Sometimes actions, documents, and offices actually speak louder than words!"

There are many ways to graphically depict the system. The analyst should choose among these tools early on to get an overview of the system. These approaches include drawing context-level data flow diagrams, capturing relationships early on with entity-relationship diagrams; or drawing use case diagrams or writing use case scenarios based on user stories. Using these diagrams and techniques at the beginning of analysis can help the analyst define the boundaries and scope of the system, and can help bring into focus which people and systems are external to the system being developed.

Entity-relationship diagrams help the systems analyst understand the entities and relationships that comprise the organizational system. E-R diagrams can depict a one-to-one relationship, a one-to-many relationship, a many-to-one relationship, and a many-to-many relationship.

The three levels of managerial control are operational, middle management, and strategic. The time horizon of decision making is different for each level.

FIGURE 2.HC1

Click on key words in the HyperCase and find out more detail.

Organizational cultures and subcultures are important determinants of how people use information and information systems. By grounding information systems in the context of the organization as a larger system, it is possible to realize that numerous factors are important and should be taken into account when ascertaining information requirements and designing and implementing information systems.

HYPERCASE Questions

1. What major organizational change recently took place at MRE? What department(s) was (or were) involved? Why was the change made?
2. What are the goals of the Training and Management Systems Department?
3. Would you categorize MRE as a service industry, a manufacturer, or both? What kind of "products" does MRE "produce"? Suggest how the type of industry MRE is in affects the information systems it uses.
4. What type of organizational structure does MRE have? What are the implications of this structure for MIS?
5. Describe in a paragraph the "politics" of the Training and Management Systems Department at MRE. Who is involved, and what are some of the main issues?
6. Draw a use case diagram representing the activities of the Webster Design group at MRE when developing site and facility master plans (use the MRE Web site to obtain your basic information).

KEYWORDS AND PHRASES

actor
associative entity
attributive entity
closedness
context-level data flow diagram
crow's foot notation
enterprise resource planning (ERP)
enterprise systems
entity (fundamental entity)
entity-relationship (E-R) diagrams
environment
feedback
four levels of use cases
interdependent
interrelatedness
middle management
openness
operations management
organizational boundaries
organizational culture
scope of the system
strategic management
systems
use case
use case diagram
use case scenario
virtual enterprise
virtual organization
virtual team

REVIEW QUESTIONS

1. What are the three groups of organizational fundamentals that carry implications for the development of information systems?
2. What is meant by saying that organizational subsystems are interrelated and interdependent?
3. Define the term *organizational boundary.*
4. What are the two main purposes for feedback in organizations?
5. Define openness in an organizational environment.
6. Define closedness in an organizational environment.
7. What is the difference between a traditional organization and a virtual one?
8. What are the potential benefits and a drawback of a virtual organization?
9. Give an example of how systems analysts could work with users as a virtual team.
10. What are enterprise systems?
11. What is ERP, and what is its purpose?
12. What problems do analysts often encounter when they try to implement an ERP package?
13. What are the two symbols on a use case diagram, and what do they represent?
14. What is a use case scenario?
15. What are the three main parts of a use case scenario?
16. What are the four steps in creating use case descriptions?
17. What are the five altitude metaphors for describing use case on different levels? What do they represent?
18. What does a process represent on a context-level data flow diagram?
19. What is an entity on a data flow diagram?
20. What is meant by the term *entity-relationship diagram*?
21. What symbols are used to draw E-R diagrams?
22. List the types of E-R diagrams.
23. How do an entity, an associative entity, and an attributive entity differ?
24. List the three broad, horizontal levels of management in organizations.
25. How can understanding organizational subcultures help in the design of information systems?

PROBLEMS

1. "It's hard to focus on what we want to achieve. I look at what our real competitors, the convenience stores, are doing and think we should copy that. Then a hundred customers come in, and I listen to each of them, and they say we should keep our little store the same, with friendly clerks and old-fashioned cash registers. Then, when I pick up a copy of *SuperMarket News*, they say that the wave of the future is super grocery stores, with no individual prices marked and UPC scanners replacing clerks. I'm pulled in so many directions I can't really settle on a strategy for our grocery store," admits Geoff Walsham, owner and manager of Jiffy Geoff's Grocery Store.

   In a paragraph, apply the concept of permeable organizational boundaries to analyze Geoff's problem in focusing on organizational objectives.
2. Write seven sentences explaining the right-to-left relationships in Figure 2.8.

3. Draw an entity-relationship diagram of a patient–doctor relationship.
   a. Which of the types of E-R diagrams is it?
   b. In a sentence or two, explain why the patient–doctor relationship is diagrammed in this way.
4. You began drawing E-R diagrams soon after your entry into the health maintenance organization for which you're designing a system. Your team member is skeptical about using E-R diagrams before the design of the database is begun. In a paragraph, persuade your team member that early use of E-R diagrams is worthwhile.
5. Neil is a decision maker for Pepe's Atlantic Sausage Company. Because there are several suppliers of ingredients and their prices fluctuate, he has come up with several different formulations for the various sausages that he makes, depending on the availability of particular ingredients from particular suppliers. He then orders ingredients accordingly twice a week. Even though he cannot predict when ingredients will become available at a particular price, his ordering of supplies can be considered routine.
   a. On what level of management is Neil working? Explain in a paragraph.
   b. What attributes of his job would have to change before you would categorize him as working on a different level of management? List them.
6. Many of the people who work at Pepe's (Problem 5) are extremely dedicated to Pepe's and have devoted their lives to the company. Others feel that the company is behind the times and should use more sophisticated production systems, information systems, and supply chain management to make the company more competitive. Members of a third group feel that what they do is unappreciated. Describe the various subculture in words. Assign them a name based on their emotions.
7. Alice in the human resources department at the Cho Manufacturing plant is constantly being asked by employees how much is taken out of their paychecks for insurance, taxes, medical, mandatory retirement, and voluntary retirement. "It takes up to a few hours every day," says Alice.

   She would like a Web system that would allow employees to use a secure logon to view the information. Alice wants the system to interface with health and dental insurance companies to obtain the amount remaining in the employee's account for the year. She would also like to obtain retirement amounts saved along with investment results. Alice has a high regard for privacy and wants the system to have employees register and give permission to obtain financial amounts from the dental insurance and retirement companies. Draw a use case diagram representing the activities of the Employee Benefit system.
8. Write up a use case scenario for the use case diagram you constructed for Cho Manufacturing.
9. What level are you creating your use case at? Choose one of the four altitude metaphors and explain why you chose it.
10. Create a context-level data flow diagram for the Employee Benefit system in Problem 7. Make any assumptions about the data to and from the central process. Do you find this to be better or not as good at explaining the system to Alice than the use case and use case scenarios?
11. Draw a use case and write up a use case scenario for getting two or three email accounts. Think about the steps that are needed to ensure security.

GROUP PROJECTS

1. Break up into groups of five. Assign one person to act as the Web site designer, one to write copy for a company's product, one to keep track of customer payments, one to monitor distribution, and one to satisfy customers who have questions about using the product. Then select a simple product (one that does not have too many versions). Good examples are a digital camera, a DVD player, a GPS, a box of candy, or a specialty travel hat (rainproof or sunblocker). Now spend 20 minutes trying to explain to the Web site designer what to include on the Web site. Describe in about three paragraphs what experience your group had in coordination. Elaborate on the interrelatedness of subsystems in the organization (your group).
2. In a small group, develop a use case and a use case scenario for making air, hotel, and car reservations for domestic travel.
3. Change your answer in Group Project 2 to include foreign travel. How does the use case and use case scenario change?
4. With your group, draw a context-level data flow diagram of your school's or university's registration system. Label each entity and process. Discuss why there appear to be different ways to draw the diagram. Reach consensus as a group about the best way to draw the diagram and defend your choice in a paragraph. Now, working with your group's members, follow the appropriate steps for developing an E-R diagram and create one for your school or university registration system. Make sure your group indicates whether the relationship you depict is one-to-one, one-to-many, many-to-one, or many-to-many.

SELECTED BIBLIOGRAPHY

Bleeker, S. E. "The Virtual Organization." *Futurist,* Vol. 28, No. 2, 1994, pp. 9–14.

Chen, P. "The Entity-Relationship Model—Towards a Unified View of Data." *ACM Transactions on Database Systems,* Vol. 1, March 1976, pp. 9–36.

Ching, C., C. W. Holsapple, and A. B. Whinston. "Toward IT Support for Coordination in Network Organizations." *Information Management,* Vol. 30, No. 4, 1996, pp. 179–199.

Cockburn, A. "Use Case Icons," http://alistair.cockburn.us/Use+case+icons?version=8339&diff=8339&with=6296. Last accessed March 18, 2009.

Davis, G. B., and M. H. Olson. *Management Information Systems, Conceptual Foundations, Structure, and Development,* 2d ed. New York: McGraw-Hill, 1985.

Galbraith, J. R. *Organizational Design.* Reading, MA: Addison-Wesley, 1977.

Kendall, K. E., J. R. Buffington, and J. E. Kendall. "The Relationship of Organizational Subcultures to DSS User Satisfaction." *Human Systems Management,* March 1987, pp. 31–39.

Kulak, D., and E. Guiney. *Use Cases: Requirement in Context,* 2d ed. Boston: Pearson Education, 2004.

PeopleSoft. Available at: www.peoplesoft.com/corplen/public_index.jsp. Accessed June 3, 2003.

Warkentin, M., L. Sayeed, and R. Hightower. "Virtual Teams versus Face-to-Face Teams; An Exploratory Study of a Web-Based Conference System." In *Emerging Information Technologies: Improving Decisions, Cooperation, and Infrastructure.* Edited by K. E. Kendall, pp. 241–262. Thousand Oaks, CA: Sage Publications, 1999.

Yager, S. E. "Everything's Coming Up Virtual." Available at: www.acm.org/crossroads/xrds4-1/organ.html. Accessed June 3, 2003.

EPISODE 2

CPU CASE

ALLEN SCHMIDT, JULIE E. KENDALL, AND KENNETH E. KENDALL

Picturing the Relationships

"So the project involves more than simply performing maintenance work on the current programs," Chip says. "Are we using a formal methodology for analyzing and designing the new system?"

"Yes," replies Anna. "We are also using Microsoft Visio to create and modify the diagrams and some simple repository information. We also have a CASE tool, Visible Analyst, to analyze and design the system.[1] We've recently installed the products on the computer in the office."

With a few easy mouse clicks Anna comes to a context-level data flow diagram (see Figure E2.1). "It's very useful to begin thinking of the system this way," Anna says as they look at the diagram on the screen.

Chip agrees, saying, "I can very easily see what you think is happening with the system. For instance, I see that the external entity Management supplies hardware and software inquiries and receives the corresponding responses in return. It shows the system within the larger organization."

"I've also drawn a preliminary E-R diagram of the system," Anna says as she brings up the entity-relationship diagram on the screen (see Figure E2.2). "It may need modification as we learn more about the system."

"Yes, the many-to-many and one-to-many relationships are very clear when you look at this," Chip says, viewing the screen.

"There is one more view of the system," continues Anna, opening the use case diagram. "This is the preliminary use case diagram for our system (see Figure E2.3). We are going to use it to obtain some valuable

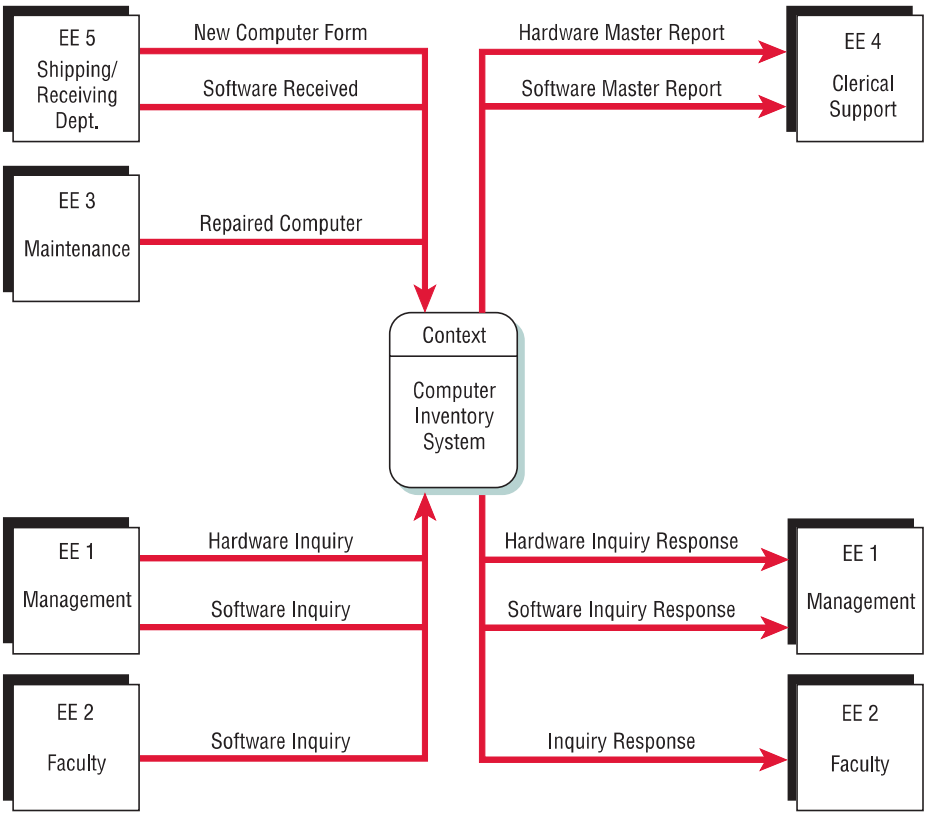


FIGURE E2.1
Context-level data flow diagram, current system.

[1]For more details on how to begin using Visible Analyst, see Allen Schmidt, *Working with Visible Analyst,* 2d ed. (Upper Saddle River, NJ: Prentice Hall, 2004).