response. For example, if the problem involves a mission-critical system, an IT maintenance team would attempt to restore normal operations. When the system is functioning properly, the team conducts a review and prepares a systems request to cover the work that was performed.

Systems Review Committee

Most large companies use a systems review committee to evaluate systems requests. Instead of relying on a single individual, a committee approach provides a variety of experience and knowledge. With a broader viewpoint, a committee can establish priorities more effectively than an individual, and one person's bias is less likely to affect the decisions. A typical committee consists of the IT director and several managers from other departments. The IT director usually serves as a technical consultant to ensure that committee members are aware of crucial issues, problems, and opportunities.

Although a committee offers many advantages, some disadvantages exist. For example, action on requests must wait until the committee meets. To avoid delay, committee members typically use e-mail and teleconferencing to communicate. Another potential disadvantage of a committee is that members might favor projects requested by their own departments, and internal political differences could delay important decisions.

Many smaller companies rely on one person to evaluate system requests instead of a committee. If only one person has the necessary IT skills and experience, that person must consult closely with users and managers throughout the company to ensure that business and operational needs are considered carefully.

Whether one person or a committee is responsible, the goal is to evaluate the requests and set priorities. Suppose four requests must be reviewed: a request from the marketing group to analyze current customer spending habits and forecast future trends; a request from the technical support group for a cellular link so service representatives can download technical data instantly; a request from the accounting department to redesign customer statements and allow Internet access; and a request from the production staff for an inventory control system that can exchange data with major suppliers. Which of those projects should the firm pursue? What criteria should be applied? How should priorities be determined? To answer those questions, the individual or the committee must assess the feasibility of each systems request.

Overview of Feasibility

A systems request must pass several tests, called a **feasibility study**, to see whether it is worthwhile to proceed further. As shown in Figure 2-13, a feasibility study uses four main yardsticks to measure a proposal: operational feasibility, technical feasibility, economic feasibility, and schedule feasibility.

Sometimes a feasibility study is quite simple and can be done in a few hours. If the request involves a new system or a major change, however, extensive fact-finding and investigation is required.

How much effort needs to go into a feasibility study? That depends on the request. For example, if a department wants an existing report sorted in a different order, the analyst can decide quickly whether the request is feasible. On the other hand, a proposal by the marketing department for a new market research system to predict sales trends requires more effort. In both cases, the systems analyst asks these important questions:

- Is the proposal desirable in an operational sense? Is it a practical approach that will solve a problem or take advantage of an opportunity to achieve company goals?
- Is the proposal technically feasible? Are the necessary technical resources and people available for the project?

- Is the proposal economically desirable? What are the projected savings and costs? Are other intangible factors involved, such as customer satisfaction or company image? Is the problem worth solving, and will the request result in a sound business investment?
- Can the proposal be accomplished within an acceptable time frame?

To obtain more information about a systems request, you might perform initial fact-finding by studying organization charts, performing interviews, reviewing current documentation, observing operations, and surveying users. If the systems request is approved, more intensive fact-finding will continue during the systems analysis phase.

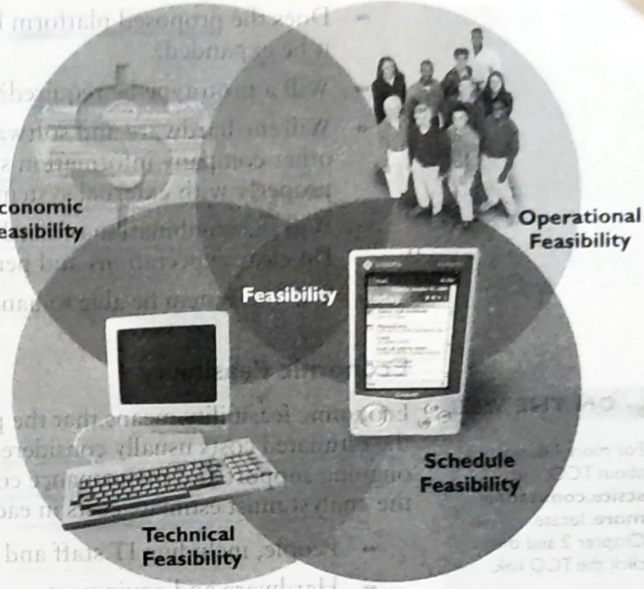


FIGURE 2-13 A feasibility study includes tests for operational, technical, economic, and schedule feasibility.

Operational Feasibility

Operational feasibility means that a proposed system will be used effectively after it has been developed. If users have difficulty with a new system, it will not produce the expected benefits. Operational feasibility depends on several vital issues. For example, consider the following questions:

- Does management support the project? Do users support the project? Is the current system well liked and effectively used? Do users see the need for change?
- Will the new system result in a workforce reduction? If so, what will happen to affected employees?
- Will the new system require training for users? If so, is the company prepared to provide the necessary resources for training current employees?
- Will users be involved in planning the new system right from the start?
- Will the new system place any new demands on users or require any operating changes? For example, will any information be less accessible or produced less frequently? Will performance decline in any way? If so, will an overall gain to the organization outweigh individual losses?
- Will customers experience adverse effects in any way, either temporarily or permanently?
- Will any risk to the company's image or goodwill result?
- Does the development schedule conflict with other company priorities?
- Do legal or ethical issues need to be considered?

Technical Feasibility

Technical feasibility refers to the technical resources needed to develop, purchase, install, or operate the system. When assessing technical feasibility, an analyst must consider the following points:

- Does the company have the necessary hardware, software, and network resources? If not, can those resources be acquired without difficulty?
- Does the company have the needed technical expertise? If not, can it be acquired?

- Does the proposed platform have sufficient capacity for future needs? If not, can it be expanded?
- Will a prototype be required?
- Will the hardware and software environment be reliable? Will it integrate with other company information systems, both now and in the future? Will it interface properly with external systems operated by customers and suppliers?
- Will the combination of hardware and software supply adequate performance? Do clear expectations and performance specifications exist?
- Will the system be able to handle future transaction volume and company growth?

Economic Feasibility

ON THE WEB

For more information about TCO, visit **scsite.com/sad6e/more**, locate Chapter 2 and then click the TCO link.

Economic feasibility means that the projected benefits of the proposed system outweigh the estimated costs usually considered the **total cost of ownership** (TCO), which includes ongoing support and maintenance costs, as well as acquisition costs. To determine TCO, the analyst must estimate costs in each of the following areas:

- People, including IT staff and users
- Hardware and equipment
- Software, including in-house development as well as purchases from vendors
- Formal and informal training
- Licenses and fees
- Consulting expenses
- Facility costs
- The estimated cost of not developing the system or postponing the project

In addition to costs, you need to assess tangible and intangible benefits to the company. The systems review committee will use those figures, along with your cost estimates, to decide whether to pursue the project beyond the preliminary investigation phase.

Tangible benefits are benefits that can be measured in dollars. Tangible benefits result from a decrease in expenses, an increase in revenues, or both. Examples of tangible benefits include the following:

- A new scheduling system that reduces overtime
- An online package tracking system that improves service and decreases the need for clerical staff
- A sophisticated inventory control system that cuts excess inventory and eliminates production delays

TOOLKIT TIME

The Financial Analysis tools in Part 2 of the Systems Analyst's Toolkit can help you analyze project costs, benefits, and economic feasibility. To learn more about these tools, turn to Part 2 of the six-part Toolkit that follows Chapter 10.

Intangible benefits are advantages that are difficult to measure in dollars but are important to the company. Examples of intangible benefits include the following:

- A user-friendly system that improves employee job satisfaction
- A sales tracking system that supplies better information for marketing decisions
- A new Web site that enhances the company's image

You also must consider the development timetable, because some benefits might occur as soon as the system is operational, but others might not take place until later.

Schedule Feasibility

Schedule feasibility means that a project can be implemented in an acceptable time frame. When assessing schedule feasibility, a systems analyst must consider the interaction between time and costs. For example, speeding up a project schedule might make a project feasible, but much more expensive.

Other issues that relate to schedule feasibility include the following:

- Can the company or the IT team control the factors that affect schedule feasibility?
- Has management established a firm timetable for the project?
- What conditions must be satisfied during the development of the system?
- Will an accelerated schedule pose any risks? If so, are the risks acceptable?
- Will project management techniques be available to coordinate and control the project?
- Will a project manager be appointed?

Part 4 of the Systems Analyst's Toolkit describes various project management tools and techniques.

ON THE WEB

For more information about Project Management Tools, visit **scsite.com/sad6e/more**, locate Chapter 2 and then click the Project Management Tools link.

Evaluating Feasibility

The first step in evaluating feasibility is to identify and weed out systems requests that are not feasible. For example, a request would not be feasible if it required hardware or software that the company already had rejected.

Even if the request is feasible, it might not be necessary. For example, a request for multiple versions of a report could require considerable design and programming effort. A better alternative might be to download the server data to a personal computer-based software package and show users how to produce their own reports. In this case, training users would be a better investment than producing reports for them.

Also keep in mind that systems requests that are not currently feasible can be resubmitted as new hardware, software, or expertise becomes available. Development costs might decrease, or the value of benefits might increase enough that a systems request eventually becomes feasible. Conversely, an initially feasible project can be rejected later. As the project progresses, conditions often change. Acquisition costs might increase, and the project might become more expensive than anticipated. In addition, managers and users sometimes lose confidence in a project. For all those reasons, feasibility analysis is an ongoing task that must be performed throughout the systems development process.

Setting Priorities

After rejecting systems requests that are not feasible, the systems review committee must establish priorities for the remaining items. The highest priority goes to projects that provide the greatest benefit, at the lowest cost, in the shortest period of time. Many factors, however, influence project evaluation.

INTERVIEWS

Systems analysts spend a great deal of time talking with people, both inside and outside the IT department. Much of that time is spent conducting interviews, which are the most common fact-finding technique. An **interview** is a planned meeting during which you obtain information from another person. You must have the skills needed to plan, conduct, and document interviews successfully.

After you identify the information you need, as described earlier in the chapter, you can begin the interviewing process, which consists of seven steps for each interview:

1. Determine the people to interview.
2. Establish objectives for the interview.
3. Develop interview questions.
4. Prepare for the interview.
5. Conduct the interview.
6. Document the interview.
7. Evaluate the interview.

Step 1: Determine the People to Interview

To get an accurate picture, you must select the right people to interview and ask them the right questions. During the preliminary investigation, you talked mainly to middle managers or department heads. Now, during the systems analysis phase, you might need to interview people from all levels of the organization.

Although you can select your interview candidates from the formal organization charts that you reviewed earlier, you also must consider any informal structures that exist in the organization. Informal structures usually are based on interpersonal relationships and can develop from previous work assignments, physical proximity, unofficial procedures, or personal relationships such as the informal gathering shown in Figure 3-17. In an **informal structure**, some people have more influence or knowledge than appears on an organization chart. Your knowledge of the company's formal and informal structures helps you determine the people to interview during the systems analysis phase.

Should you interview several people at the same time? Group interviews can save time and provide an opportunity to observe interaction among the participants. Group interviews also can present problems. One person might dominate the conversation, even when questions are addressed specifically to others. Organization level also can present a problem, as the presence of senior managers in an interview might prevent lower-level employees from expressing themselves candidly.

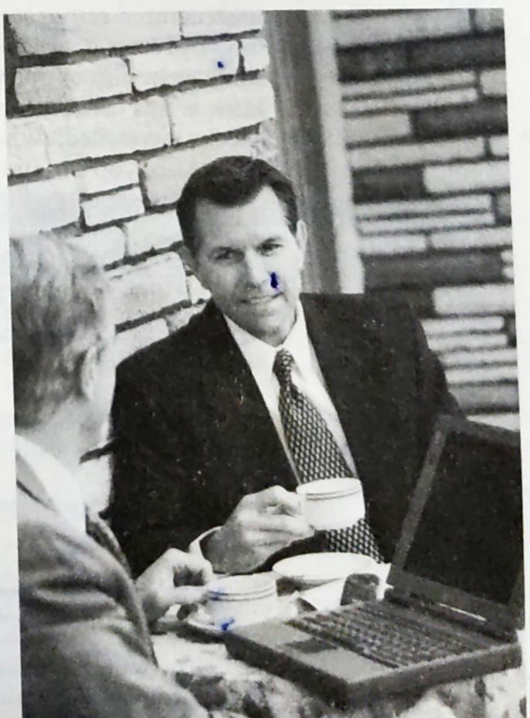


FIGURE 3-17 An analyst must consider informal structures in the organization when selecting interview candidates.

Step 2: Establish Objectives for the Interview

After deciding on the people to interview, you must establish objectives for the session. First, you should determine the general areas to be discussed, and then list the facts you want to gather. You also should try to solicit ideas, suggestions, and opinions during the interview.

The objectives of an interview depend on the role of the person being interviewed. Upper-level managers can provide the big picture and help you to understand the system as a whole. Specific details about operations and business processes are best learned from people who actually work with the system on a daily basis.

In the early stages of systems analysis, interviews usually are general. As the fact-finding process continues, however, the interviews focus more on specific topics. Interview objectives also vary at different stages of the investigation. By setting specific objectives, you create a framework that helps you decide what questions to ask and how to phrase the questions.

Step 3: Develop Interview Questions

Creating a standard list of interview questions helps to keep you on track and avoid unnecessary tangents. Also, if you interview several people who perform the same job, a standard question list allows you to compare their answers. Although you have a list of specific questions, you might decide to depart from it because an answer to one question leads to another topic that you want to pursue. That question or topic then should be included in a revised set of questions used to conduct future interviews. If the question proves to be extremely important, you may need to return to a previous interviewee to query him or her on the topic.

The interview should consist of several different kinds of questions: open-ended, closed-ended, or questions with a range of responses. When you phrase your questions, you should avoid **leading questions** that suggest or favor a particular reply. For example, rather than asking, "What advantages do you see in the proposed system?" you might ask, "Do you see any advantages in the proposed system?"

OPEN-ENDED QUESTIONS **Open-ended questions** encourage spontaneous and unstructured responses. Such questions are useful when you want to understand a larger process or draw out the interviewee's opinions, attitudes, or suggestions. Here are some examples of open-ended questions: What are users saying about the new system? How is this task performed? Why do you perform the task that way? How are the checks reconciled? What added features would you like to have in the new billing system? Also, you can use an open-ended question to probe further by asking: Is there anything else you can tell me about this topic?

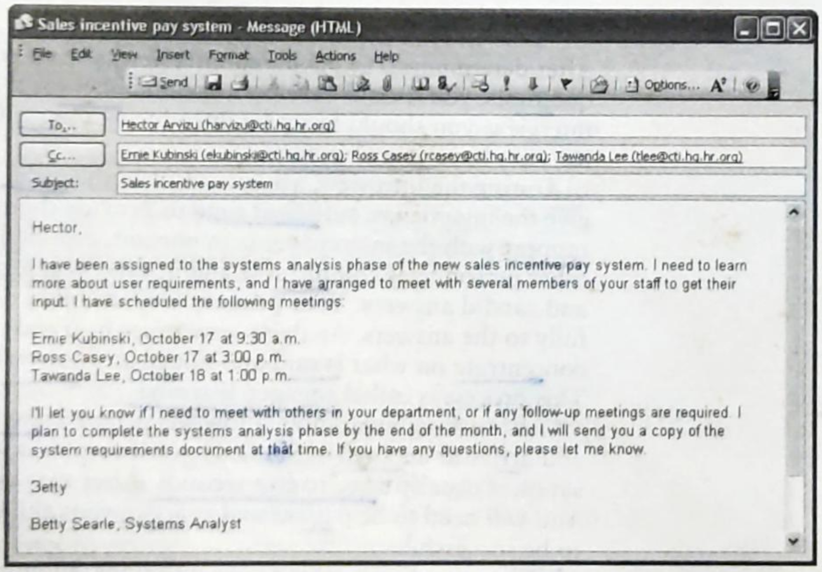
CLOSED-ENDED QUESTIONS **Closed-ended questions** limit or restrict the response. You use closed-ended questions when you want information that is more specific or when you need to verify facts. Examples of closed-ended questions include the following: How many personal computers do you have in this department? Do you review the reports before they are sent out? How many hours of training does a clerk receive? Is the calculation procedure described in the manual? How many customers ordered products from the Web site last month?

RANGE-OF-RESPONSE QUESTIONS **Range-of-response questions** are closed-ended questions that ask the person to evaluate something by providing limited answers to specific responses or on a numeric scale. This method makes it easier to tabulate the answers and interpret the results. Range-of-response questions might include these: On a scale of 1 to 10, with 1 the lowest and 10 the highest, how effective was your training? How would you rate the severity of the problem: low, medium, or high? Is the system shutdown something that occurs never, sometimes, often, usually, or always?

Step 4: Prepare for the Interview

After setting the objectives and developing the questions, you must prepare for the interview. Careful preparation is essential because an interview is an important meeting and not just a casual chat. When you schedule the interview, suggest a specific day and time and let the interviewee know how long you expect the meeting to last. It is also a good idea to send an e-mail or place a reminder call the day before the interview.

Remember that the interview is an interruption of the other person's routine, so you should limit the interview to no more than one hour. If business pressures force a postponement of the meeting, you should schedule another appointment as soon as it is convenient. Remember to keep department managers informed of your meetings with their staff members. Sending a message to each department manager listing your planned appointments is a good way to keep them informed. Figure 3-18 is an example of such a message.



Sales incentive pay system - Message (HTML)

To... Hector Arvizu (harvizu@cti.hq.hr.org)

Cc... Ernie Kubinski (ekubinski@cti.hq.hr.org); Ross Casey (rcasey@cti.hq.hr.org); Tawanda Lee (tlee@cti.hq.hr.org)

Subject: Sales incentive pay system

Hector,

I have been assigned to the systems analysis phase of the new sales incentive pay system. I need to learn more about user requirements, and I have arranged to meet with several members of your staff to get their input. I have scheduled the following meetings:

Ernie Kubinski, October 17 at 9:30 a.m.
Ross Casey, October 17 at 3:00 p.m.
Tawanda Lee, October 18 at 1:00 p.m.

I'll let you know if I need to meet with others in your department, or if any follow-up meetings are required. I plan to complete the systems analysis phase by the end of the month, and I will send you a copy of the system requirements document at that time. If you have any questions, please let me know.

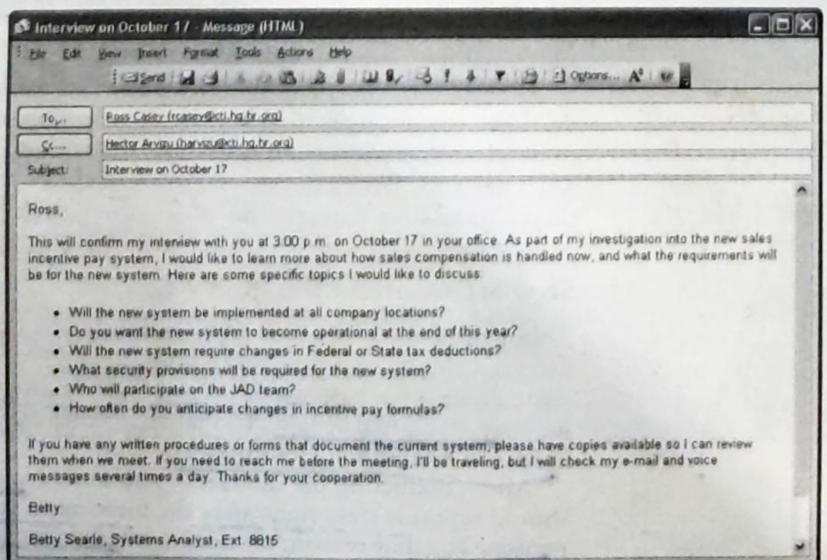
Betty

Betty Searle, Systems Analyst

FIGURE 3-18 Sample message to a department head about interviews with people in his group.

You should send a list of topics to an interviewee several days before the meeting, especially when detailed information is needed, so the person can prepare for the interview and minimize the need for a follow-up meeting. Figure 3-19 shows a sample message that lists specific questions and confirms the date, time, location, purpose, and anticipated duration of the interview.

If you have questions about documents, ask the interviewee to have samples available at the meeting. Your advance memo should include a list of the documents you want to discuss, if you know what they are. Otherwise, you can make a general request for documents, as the analyst did in her e-mail shown in Figure 3-19.



Interview on October 17 - Message (HTML)

To... Ross Casey (rcasey@cti.hq.hr.org)

Cc... Hector Arvizu (harvizu@cti.hq.hr.org)

Subject: Interview on October 17

Ross,

This will confirm my interview with you at 3:00 p.m. on October 17 in your office. As part of my investigation into the new sales incentive pay system, I would like to learn more about how sales compensation is handled now, and what the requirements will be for the new system. Here are some specific topics I would like to discuss:

- Will the new system be implemented at all company locations?
- Do you want the new system to become operational at the end of this year?
- Will the new system require changes in Federal or State tax deductions?
- What security provisions will be required for the new system?
- Who will participate on the JAD team?
- How often do you anticipate changes in incentive pay formulas?

If you have any written procedures or forms that document the current system, please have copies available so I can review them when we meet. If you need to reach me before the meeting, I'll be traveling, but I will check my e-mail and voice messages several times a day. Thanks for your cooperation.

Betty

Betty Searle, Systems Analyst, Ext. 8815

FIGURE 3-19 Sample message to confirm a planned meeting and provide advance information about topics to be discussed.

Two schools of thought exist about the best location for an interview. Some analysts believe that interviews should take place in the interviewee's office, whereas other analysts feel that a neutral location such as a conference room is better.

Supporters of interviews in the interviewee's office believe that is the best location because it makes the interviewee feel comfortable during the meeting. A second argument in favor of the interviewee's office is that the office is where he or she has the easiest access to supporting material that might be needed during the discussion. If you provide a complete list of topics in advance, however, the interviewee can bring the necessary items to a conference room or other location.

Supporters of neutral locations stress the importance of keeping interruptions to a minimum so both people can concentrate fully. In addition, an interview that is free of interruptions takes less time. If the meeting does take place in the interviewee's office, you should suggest tactfully that all calls be held until the conclusion of the interview.

Step 5: Conduct the Interview

After determining the people to interview, setting your objectives, and preparing the questions, you should develop a specific plan for the meeting. When conducting an interview, you should begin by introducing yourself, describing the project, and explaining your interview objectives.

During the interview, ask questions in the order in which you prepared them, and give the interviewee sufficient time to provide thoughtful answers. Establishing a good rapport with the interviewee is important, especially if this is your first meeting. If the other person feels comfortable and at ease, you probably will receive more complete and candid answers. Your primary responsibility during an interview is to listen carefully to the answers. Analysts sometimes hear only what they expect to hear. You must concentrate on what is said and notice any nonverbal communication that takes place. This process is called **engaged listening.**

After asking a question, allow the person enough time to think about the question and arrive at an answer. Studies have shown that the maximum pause during a conversation is usually three to five seconds. After that interval, one person will begin talking. You will need to be patient and practice your skills in many actual interview situations to be successful.

When you finish asking your questions, summarize the main points covered in the interview and explain the next course of action. For example, mention that you will send a follow-up memo or that the interviewee should get back to you with certain information. When you conclude the interview, thank the person and encourage him or her to contact you with any questions or additional comments. Also, when the interview ends, it is a good idea to ask the interviewee whether he or she can suggest any additional topics that should be discussed.

After an interview, you should summarize the session and seek a confirmation from the other person. By stating your understanding of the discussion, the interviewee can respond and correct you, if necessary. One good approach is to rephrase the interviewee's answers. For example, you can say, "If I understand you correctly, you are saying that ..." and then reiterate the information given by the interviewee.

Step 6: Document the Interview

Although taking notes during an interview has both advantages and disadvantages, the accepted view is that note taking should be kept to a minimum. Although you should write down a few notes to jog your memory after the interview, you should avoid writing everything that is said. Too much writing distracts the other person and makes it harder to establish a good rapport.

After conducting the interview, you must record the information quickly. You should set aside time right after the meeting to record the facts and evaluate the information. For that reason, try not to schedule back-to-back interviews. Studies have shown that 50 percent of a conversation is forgotten within 30 minutes. You, therefore, should use your notes to record the facts immediately so you will not forget them. You can summarize the facts by preparing a narrative describing what took place or by recording the answers you received next to each question on your prepared question list.

Tape recorders are effective tools for an interview; however, many people feel uncomfortable when recorders are present. Before using a recorder, you should discuss its use with the interviewee. Assure the interviewee that you will erase the tape after you transcribe your notes and that you will stop and rewind the tape anytime during the interview at his or her request. If you ask sensitive questions or the interviewee wants to answer a question without being recorded, explain that you will turn off the tape for a period of time during the interview.

Even with a tape recorder in use, you should listen carefully to the interviewee's responses so you can ask good follow-up questions. Otherwise, you might have to return for a second visit to ask the questions you missed the first time. Also, remember that each recorded interview takes twice the amount of time, because you must listen to or view the recorded meeting again after conducting the interview itself.

After the interview, send a memo to the interviewee expressing your appreciation for his or her time and cooperation. In the memo, you should note the date, time, location, purpose of the interview, and the main points you discussed so the interviewee has a written summary and can offer additions or corrections.

Step 7: Evaluate the Interview

In addition to recording the facts obtained in an interview, try to identify any possible biases. For example, an interviewee who tries to protect his or her own area or function might give incomplete answers or refrain from volunteering information. Or, an interviewee with strong opinions about the current or future system might distort the facts. Some interviewees might answer your questions in an attempt to be helpful even though they do not have the necessary experience to provide accurate information.

CASE IN POINT 3.2: DEEP RIVER COLLEGE

Deep River College is a two-year school in Southern California. Twice a year, the fund-raising office at Deep River mails requests for donations to the alumni. The staff uses a word processing program and a personal information database to create personalized letters. Data on past contributions and other alumni information, however, is stored manually. The dean, Alexandra Ali, recently submitted a systems request asking the college's IT department to develop a computerized alumni information system. The school does not have a formal systems review committee, and each department has an individual budget for information services.

Eddie Bateman, a systems analyst, performed a preliminary investigation and he concluded that the system met all the feasibility tests. After reading his report, Alexandra asked him to proceed with the systems analysis phase. Eddie has scheduled an interview with her, and he has asked you to help him prepare for the meeting. Specifically, he wants you to list all the topics he should cover during the interview. Eddie also wants you to prepare a list of specific questions that he should ask. Be sure to include open-ended, closed-ended, and range-of-response questions.

Unsuccessful Interviews

No matter how well you prepare for interviews, some are not successful. One of the main reasons could be that you and the interviewee did not get along well. Such a situation can be caused by several factors. For example, a misunderstanding or personality conflict them. You can summarize the facts by preparing a narrative describing what took place or by recording the answers you received next to each question on your prepared question list.

Tape recorders are effective tools for an interview; however, many people feel uncomfortable when recorders are present. Before using a recorder, you should discuss its use with the interviewee. Assure the interviewee that you will erase the tape after you transcribe your notes and that you will stop and rewind the tape anytime during the interview at his or her request. If you ask sensitive questions or the interviewee wants to answer a question without being recorded, explain that you will turn off the tape for a period of time during the interview.

Even with a tape recorder in use, you should listen carefully to the interviewee's responses so you can ask good follow-up questions. Otherwise, you might have to return for a second visit to ask the questions you missed the first time. Also, remember that each recorded interview takes twice the amount of time, because you must listen to or view the recorded meeting again after conducting the interview itself.

After the interview, send a memo to the interviewee expressing your appreciation for his or her time and cooperation. In the memo, you should note the date, time, location, purpose of the interview, and the main points you discussed so the interviewee has a written summary and can offer additions or corrections.

Step 7: Evaluate the Interview

In addition to recording the facts obtained in an interview, try to identify any possible biases. For example, an interviewee who tries to protect his or her own area or function might give incomplete answers or refrain from volunteering information. Or, an interviewee with strong opinions about the current or future system might distort the facts. Some interviewees might answer your questions in an attempt to be helpful even though they do not have the necessary experience to provide accurate information.

CASE IN POINT 3.2: DEEP RIVER COLLEGE

Deep River College is a two-year school in Southern California. Twice a year, the fund-raising office at Deep River mails requests for donations to the alumni. The staff uses a word processing program and a personal information database to create personalized letters. Data on past contributions and other alumni information, however, is stored manually. The dean, Alexandra Ali, recently submitted a systems request asking the college's IT department to develop a computerized alumni information system. The school does not have a formal systems review committee, and each department has an individual budget for information services.

Eddie Bateman, a systems analyst, performed a preliminary investigation and he concluded that the system met all the feasibility tests. After reading his report, Alexandra asked him to proceed with the systems analysis phase. Eddie has scheduled an interview with her, and he has asked you to help him prepare for the meeting. Specifically, he wants you to list all the topics he should cover during the interview. Eddie also wants you to prepare a list of specific questions that he should ask. Be sure to include open-ended, closed-ended, and range-of-response questions.

Unsuccessful Interviews

No matter how well you prepare for interviews, some are not successful. One of the main reasons could be that you and the interviewee did not get along well. Such a situation can be caused by several factors. For example, a misunderstanding or personality conflict could affect the interview negatively, or the interviewee might be afraid that the new system will eliminate or change his or her job.

In other cases, the interviewee might give only short or incomplete responses to your open-ended questions. If so, you should switch to closed-ended questions or questions with a range of responses, or try rephrasing your open-ended questions into those types of questions. If that still does not help, you should find a tactful way to conclude the meeting.

Continuing an unproductive interview is difficult. The interviewee could be more cooperative later, or you might find the information you seek elsewhere. If failure to obtain specific information will jeopardize the success of the project, inform your supervisor, who can help you decide what action to take. Your supervisor might contact the interviewee's supervisor, ask another systems analyst to interview the person, or find some other way to get the needed information.

CASE IN POINT 3.3: FASTPAK OVERNIGHT PACKAGE SYSTEM

FastPak, the nation's fourth-largest overnight package system, is headquartered in Los Angeles, California. Jesse Evans is a systems analyst on an IT team that is studying ways to update FastPak's package tracking system. Jesse prepared well for her interview with Jason Tanya, FastPak's executive vice president. Mr. Tanya did not ask his assistant to hold his calls during the meeting, however. After several interruptions, Jesse tactfully suggested that she could come back another time, or perhaps that Mr. Tanya might ask his assistant to hold his calls. "No way," he replied. "I'm a very busy man and we'll just have to fit this in as we can, even if it takes all day." Jesse was unprepared for his response. What are her options? Is an analyst always in control of this kind of situation? Why or why not?

OTHER FACT-FINDING TECHNIQUES

In addition to interviewing, systems analysts use other fact-finding techniques, including document review, observation, questionnaires and surveys, sampling, and research. Such techniques are used before interviewing begins to obtain a good overview and to help develop better interview questions.

Document Review

Document review can help you understand how the current system is supposed to work. Remember that system documentation sometimes is out of date. Forms can change or be discontinued, and documented procedures often are modified or eliminated. You should obtain copies of actual forms and operating documents currently in use. You also should review blank copies of forms, as well as samples of actual completed forms. You usually can obtain document samples during interviews with the people who perform that procedure. If the system uses a software package, you should review the documentation for that software.

Observation

The **observation** of current operating procedures is another fact-finding technique. Seeing the system in action gives you additional perspective and a better understanding of system procedures. Personal observation also allows you to verify statements made in interviews and determine whether procedures really operate as they are described.

Through observation, you might discover that neither the system documentation nor the interview statements are accurate.

Personal observation also can provide important advantages as the development process continues. For example, recommendations often are better accepted when they are based on personal observation of actual operations. Observation also can provide the knowledge needed to test or install future changes and can help build relationships with the users who will work with the new system.

Plan your observations in advance by preparing a checklist of specific tasks you want to observe and questions you want to ask. Consider the following issues when you prepare your list:

1. Ask sufficient questions to ensure that you have a complete understanding of the present system operation. A primary goal is to identify the methods of handling situations that are not covered by standard operating procedures. For example, what happens in a payroll system if an employee loses a time card? What is the procedure if an employee starts a shift 10 minutes late but then works 20 minutes overtime? Often, the rules for exceptions such as these are not written or formalized; therefore, you must try to document any procedures for handling exceptions.
2. Observe all the steps in a transaction and note the documents, inputs, outputs, and processes involved.
3. Examine each form, record, and report. Determine the purpose each item of information serves.
4. Consider each user who works with the system and the following questions: What information does that person receive from other people? What information does this person generate? How is the information communicated? How often do interruptions occur? How much downtime occurs? How much support does the user require, and who provides it?
5. Talk to the people who receive current reports to see whether the reports are complete, timely, accurate, and in a useful form. Ask whether information can be eliminated or improved and whether people would like to receive additional information.

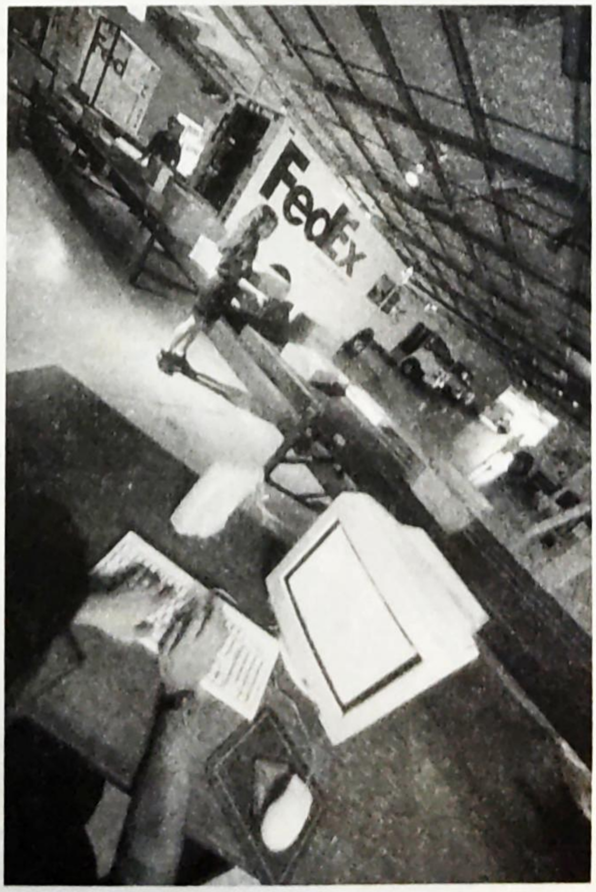


FIGURE 3-20 The Hawthorne study suggested that worker productivity improves during observation. Always consider the Hawthorne Effect when observing the operation of an existing system.

As you observe people at work, as shown in Figure 3-20, consider a factor called the **Hawthorne Effect**. The name comes from a well-known study performed in the Hawthorne plant of the Western Electric Company in the 1920s. The purpose of the study was to determine how various changes in the work environment would affect employee productivity. The surprising result was that productivity improved during observation whether the conditions were made better or worse. Researchers concluded that productivity seemed to improve whenever the workers knew they were being observed.

Thus, as you observe users, remember that normal operations might not always run as smoothly as your observations indicate. Operations also might run less smoothly because workers might be nervous during the observation. If possible, meet with workers and their supervisors to discuss your plans and objectives to help establish a good working relationship. In some situations, you might even participate in the work yourself to gain a personal understanding of the task or the environment.

Questionnaires and Surveys

In projects where it is desirable to obtain input from a large number of people, a questionnaire can be a valuable tool. A **questionnaire**, also called a **survey**, is a document containing a number of standard questions that can be sent to many individuals.

Questionnaires can be used to obtain information about a wide range of topics, including workloads, reports received, volumes of transactions handled, job duties, difficulties, and opinions of how the job could be performed better or more efficiently. Figure 3-21 shows a sample questionnaire that includes several different question and response formats.

PURCHASE REQUISITION QUESTIONNAIRE

Pat Kline, Vice President, Finance, has asked us to investigate the purchase requisition process to see if it can be improved. Your input concerning this requisition process will be very valuable. We would greatly appreciate it if you could complete the following questionnaire and return it by March 10 to Dana Juarez in information technology. If you have any questions, please call Dana at x2561.

A. **YOUR OBSERVATIONS**
Please answer each question by checking one box.

1. How many purchase requisitions did you process in the past five working days? ______

2. What percentage of your time is spent processing requisitions?
[ ] under 20% [ ] 60–79%
[ ] 21–39% [ ] 80% or more
[ ] 40–59%

3. Do you believe too many errors exist on requisitions?
[ ] yes
[ ] no

4. Out of every 100 requisitions you process, how many contain errors?
[ ] fewer than 5 [ ] 20 to 29
[ ] 5 to 9 [ ] 30 to 39
[ ] 10 to 14 [ ] 40 to 49
[ ] 15 to 19 [ ] 50 or more

5. What errors do you see most often on requisitions? (Place a 1 next to the most common error, place a 2 next to the second, etc.)
[ ] incorrect charge number [ ] missing authorization
[ ] missing charge information [ ] other (please explain) ______
[ ] arithmetic errors
[ ] incorrect discount percent used

B. **YOUR SUGGESTIONS**
Please be specific, and give examples if possible.

1. If the currently used purchase requisition form were to be redesigned, what changes to the form would you recommend?

______
______
______
______
______
______

(If necessary, please attach another sheet)

2. Would you be interested in meeting with an information technology representative to discuss your ideas further? If so, please complete the following information:

Name ______ Department ______
Telephone ______ E-mail address ______

FIGURE 3-21 Sample questionnaire.

A typical questionnaire starts with a heading, which includes a title, a brief statement of purpose, the name and telephone number of the contact person, the deadline date for completion, and how and where to return the form. The heading usually is followed by general instructions that provide clear guidance on how to answer the questions. Headings also are used to introduce each main section or portion of the survey and include instructions when the type of question or response changes. A long questionnaire might end with a conclusion that thanks the participants and reminds them how to return the form.

What about the issue of anonymity? Should people be asked to sign the questionnaire, or is it better to allow anonymous responses? The answer depends on two questions. First, does an analyst really need to know who the respondents are in order to match or correlate information? For example, it might be important to know what percentage of users need a certain software feature, but specific user names might not be relevant. Second, does the questionnaire include any sensitive or controversial topics? Many people do not want to be identified when answering a question such as "How well has your supervisor explained the system to you?" In such cases, anonymous responses might provide better information.

When designing a questionnaire, the most important rule of all is to make sure that your questions collect the right data in a form that you can use to further your fact-finding. Here are some additional ideas to keep in mind when designing your questionnaire:

- Keep the questionnaire brief and user-friendly.
- Provide clear instructions that will answer all anticipated questions.
- Arrange the questions in a logical order, going from simple to more complex topics.
- Phrase questions to avoid misunderstandings; use simple terms and wording.
- Try not to lead the response or use questions that give clues to expected answers.
- Limit the use of open-ended questions that are difficult to tabulate.
- Limit the use of questions that can raise concerns about job security or other negative issues.
- Include a section at the end of the questionnaire for general comments.
- Test the questionnaire whenever possible on a small test group before finalizing it and distributing to a large group.

A questionnaire can be a traditional paper form, or you can create a **fill-in form** and collect data on the Internet or a company intranet. For example, you can use Microsoft Word, as shown in Figure 3-22 on the next page, to create form fields, including text boxes, check boxes, and drop-down lists where users can click selections. Before you publish the form, you should protect it so users can fill it in but cannot change the layout or design. Forms also can be automated, so if a user answers no to question three, he or she goes directly to question eight, where the form-filling resumes.

Sampling

When studying an information system, you should collect examples of actual documents using a process called **sampling**. The samples might include records, reports, operational logs, data entry documents, complaint summaries, work requests, and various types of forms. Sampling techniques include systematic sampling, stratified sampling, and random sampling.

Suppose you have a list of 200 customers who complained about errors in their statements, and you want to review a representative sample of 20 customers. A **systematic sample** would select every tenth customer for review. If you want to ensure that the sample is balanced geographically, however, you could use a **stratified sample** to select five

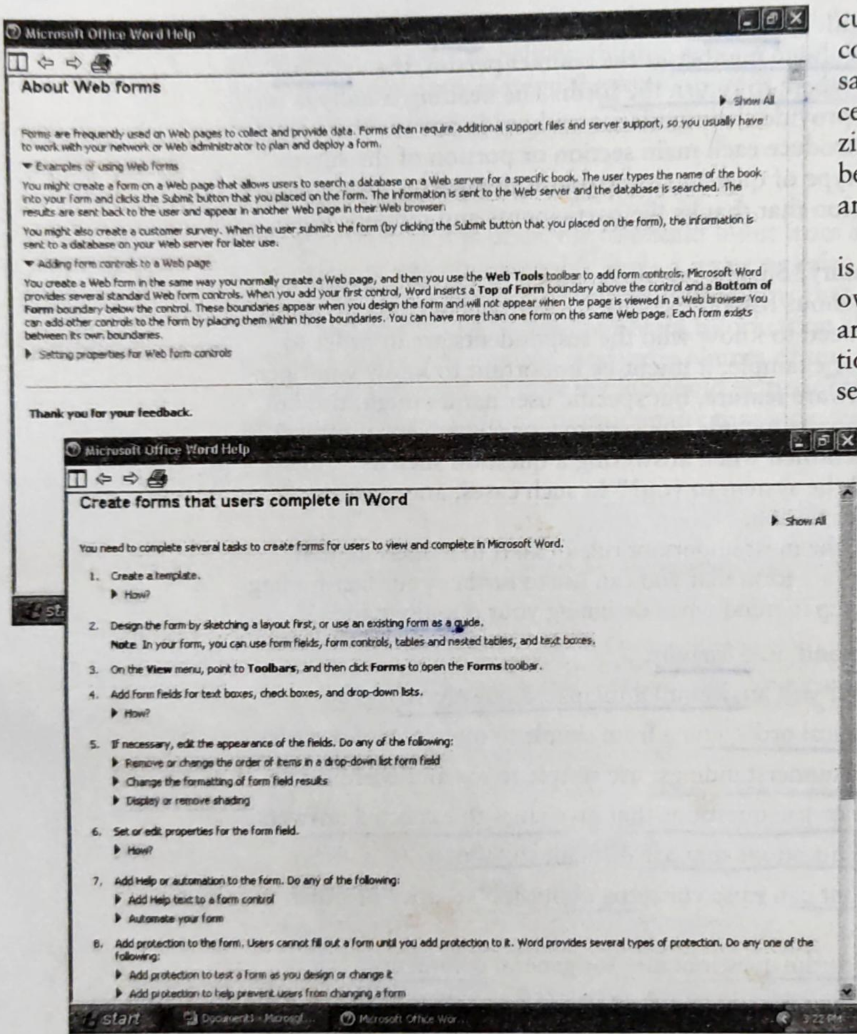


FIGURE 3-22 Microsoft Word enables you to create a fill-in form and collect data on the Internet or a company intranet.

customers from each of four ZIP codes. Another example of stratified sampling is to select a certain percentage of transactions from each zip code, rather than a fixed number. Finally, a **random sample** selects any 20 customers.

The main objective of a sample is to ensure that it represents the overall population accurately. If you are analyzing inventory transactions, for example, you should select a sample of transactions that are typical of actual inventory operations and do not include unusual or unrelated examples. For instance, if a company performs special processing on the last business day of the month, that day is not a good time to sample *typical* daily operations. To be useful, a sample must be large enough to provide a fair representation of the overall data.

You also should consider sampling when using interviews or questionnaires. Rather than interviewing everyone or sending a questionnaire to the entire group, you can use a sample of participants. You must use sound sampling techniques to reflect the overall population and obtain an accurate picture.

ON THE WEB

For more information about Sampling, visit **scsite.com/sad6e/more**, locate Chapter 3 and then click the Sampling link.

Research

Research is another important fact-finding technique. Your research can include the Internet, IT magazines, and books to obtain background information, technical material, and news about industry trends and developments. In addition, you can attend professional meetings, seminars, and discussions with other IT professionals, which can be very helpful in problem solving.

The Internet is an extremely valuable resource. Part 6 of the Systems Analyst's Toolkit describes a variety of Internet resource tools. Using the Internet, you also can access information from federal and state governments, as well as from publishers, universities, and libraries around the world. Internet **newsgroups** are good resources for exchanging information with other professionals, seeking answers to questions, and monitoring discussions that are of interest to you.

All major hardware and software vendors maintain sites on the Web where you can obtain information about products and services offered by the company and send e-mail with specific questions to company representatives. In addition to contacting specific firms, you can access Web sites maintained by publishers and independent firms that provide links to hundreds of hardware and software vendors, as shown in Figure 3-23. Such sites are one-stop information centers where IT professionals can find information, share ideas, and keep posted on developments in technology.

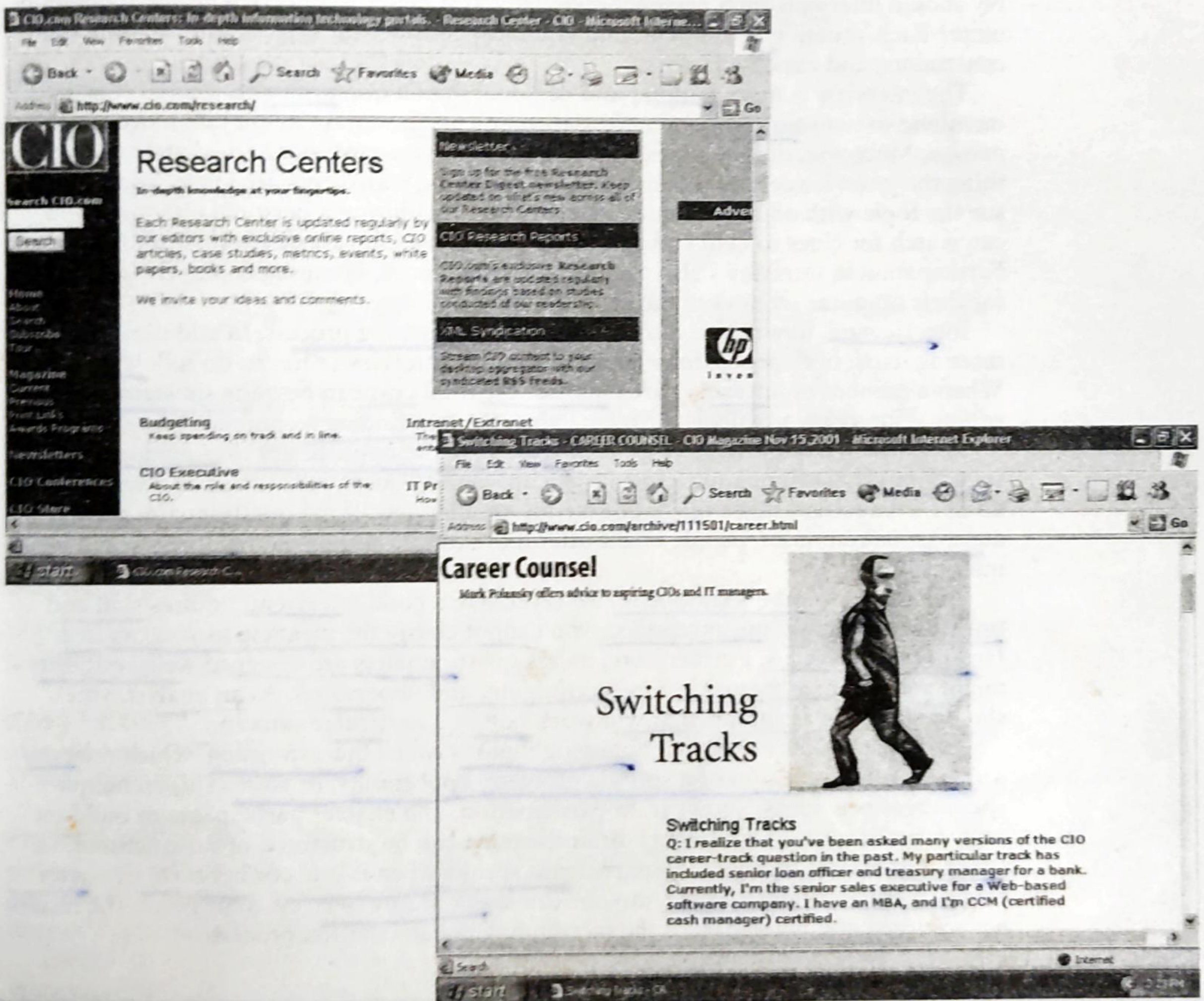


FIGURE 3-23 CIO Magazine maintains a Web site with various features for information technology professionals.

Research also can involve a visit to a physical location, called a site visit, where the objective is to observe a system in use at another location. If you are studying your firm's human resources information system, for example, you might want to see how another company's system works. Site visits also are important when considering the purchase of a software package. If the software vendor suggests possible sites to visit, be aware that such sites might constitute a biased sample. A single site visit seldom gives you true pictures, so you should try to visit more than one installation.

Before a site visit, prepare just as you would for an interview. Contact the appropriate manager and explain the purpose of your visit. Decide what questions you will ask and what processes you will observe. During your visit, observe how the system works and note any problems or limitations. You also will want to learn about the support provided by the vendor, the quality of the system documentation, and so on.

Interviews versus Questionnaires

When you seek input from a large group, a questionnaire is a very useful tool. On the other hand, if you require detailed information from only a few people, then you probably should interview each person individually. Is it better to interview or use a questionnaire? Each situation is different, and you must consider the type of information, time constraints, and expense factors.

The interview is more familiar and personal than a questionnaire. People who are unwilling to put critical or controversial comments in writing might talk more freely in person. Moreover, during a face-to-face interview, you can react immediately to anything the interviewee says. If surprising or confusing statements are made, you can pursue the topic with additional questions. In addition, during a personal interview, you can watch for clues to help you determine if responses are knowledgeable and unbiased. Participation in interviews also can affect user attitudes, because people who are asked for their opinions often view the project more favorably.

Interviewing, however, is a costly and time-consuming process. In addition to the meeting itself, both people must prepare, and the interviewer has to do follow-up work. When a number of interviews are planned, the total cost can be quite substantial. The personal interview usually is the most expensive fact-finding technique.

In contrast, a questionnaire gives many people the opportunity to provide input and suggestions. Questionnaire recipients can answer the questions at their convenience and do not have to set aside a block of time for an interview. If the questionnaire allows anonymous responses, people might offer more candid responses than they would in an interview.

Preparing a good questionnaire, however, like a good interview, requires skill and time. If a question is misinterpreted, you cannot clarify the meaning as you can in a face-to-face interview. Furthermore, unless questionnaires are designed well, recipients might view them as intrusive, time-consuming, and impersonal. As an analyst, you should select the technique that will work best in a particular situation.

Another popular method of obtaining input is called **brainstorming**, which refers to a small group discussion of a specific problem, opportunity, or issue. This technique encourages new ideas, allows team participation, and enables participants to build on each other's inputs and thoughts. Brainstorming can be structured or unstructured. In **structured brainstorming**, each participant speaks when it is his or her turn, or passes. In **unstructured brainstorming**, anyone can speak at any time. At some point, the results are recorded and made part of the fact-finding documentation process.

CASE IN POINT 3.4: CYBERSTUFF

Ann Ellis is a systems analyst at CyberStuff, a large company that sells computer hardware and software via telephone, mail order, and the Internet. CyberStuff processes several thousand transactions per week on a three-shift operation and employs 50 full-time and 125 part-time employees. Lately, the billing department has experienced an increase in the number of customer complaints about incorrect bills. During the preliminary investigation, Ann learned that some CyberStuff representatives did not follow established order entry procedures. She feels that with more information, she might find a pattern and identify a solution for the problem.

Ann is not sure how to proceed. She came to you, her supervisor, with two separate questions. First, is a questionnaire the best approach, or would interviews be better? Second, whether she uses interviews, a questionnaire, or both techniques, should she select the participants at random, include an equal number of people from each shift, or use some other approach? As Ann's supervisor, what would you suggest, and why?

Overview of Data and Process Modeling Tools

TOOLKIT TIME

The CASE Tools in Part 2 of the Systems Analyst's Toolkit can help you document business functions and processes, develop graphical models, and provide an overall framework for information system development. To learn more about these tools, turn to Part 2 of the six-part Toolkit that follows Chapter 10.

Systems analysts use many graphical techniques to describe an information system. One popular method is to draw a set of data flow diagrams. A **data flow diagram** (DFD) uses various symbols to show how the system transforms input data into useful information. Other graphical tools include entity-relationship diagrams, which are described in Chapter 7 (Data Design); and object models, which are described in Part 5 of the Systems Analyst's Toolkit (Object-Oriented Tools).

Data Flow Diagrams

ON THE WEB

For more information about Yourdon Symbols, visit **scsite.com/sad6e/more**, locate Chapter 4 and then click the Yourdon Symbols link.

In Part 1 of the Systems Analyst's Toolkit, you learn how to use visual aids to help explain a concept, as shown in Figure 4-2. Similarly, during the systems analysis phase, you learn how to create a visual model of the information system using a set of data flow diagrams.

A data flow diagram (DFD) shows how data moves through an information system but does not show program logic or processing steps. A set of DFDs provides a logical model that shows *what* the system does, not *how* it does it. That distinction is important because focusing on implementation issues at this point would restrict your search for the most effective system design.

DFD Symbols

DFDs use four basic symbols that represent processes, data flows, data stores, and entities. Several different versions of DFD symbols exist, but they all serve the same purpose. DFD examples in this textbook use the **Gane and Sarson** symbol set. Another popular symbol set is the **Yourdon** symbol set. Figure 4-3 shows examples of both versions. Symbols are referenced by using all capital letters for the symbol name.

PROCESS SYMBOL A **process** receives input data and produces output that has a different content, form, or both. For instance, the process for calculating pay uses two inputs (pay rate and hours worked) to produce one output (total pay). Processes can be very simple or quite complex. In a typical company, processes might include calculating sales trends, filing online insurance claims, ordering inventory from a supplier's system, or verifying e-mail addresses for Web customers. Processes contain the **business logic**, also called **business rules**, that transform the data and produce the required results.

The symbol for a process is a rectangle with rounded corners. The name of the process appears inside the rectangle. The process name identifies a specific function and consists of a verb (and an adjective, if necessary) followed by a singular noun. Examples of process names are APPLY RENT PAYMENT, CALCULATE COMMISSION, ASSIGN FINAL GRADE, VERIFY ORDER, and FILL ORDER.

Processing details are not shown in a DFD. For example, you might have a process named DEPOSIT PAYMENT. The process symbol

FIGURE 4-2 Systems analysts often use visual aids during presentations.

does not reveal the business logic for the DEPOSIT PAYMENT process. To document the logic, you create a process description, which is explained later in this chapter.

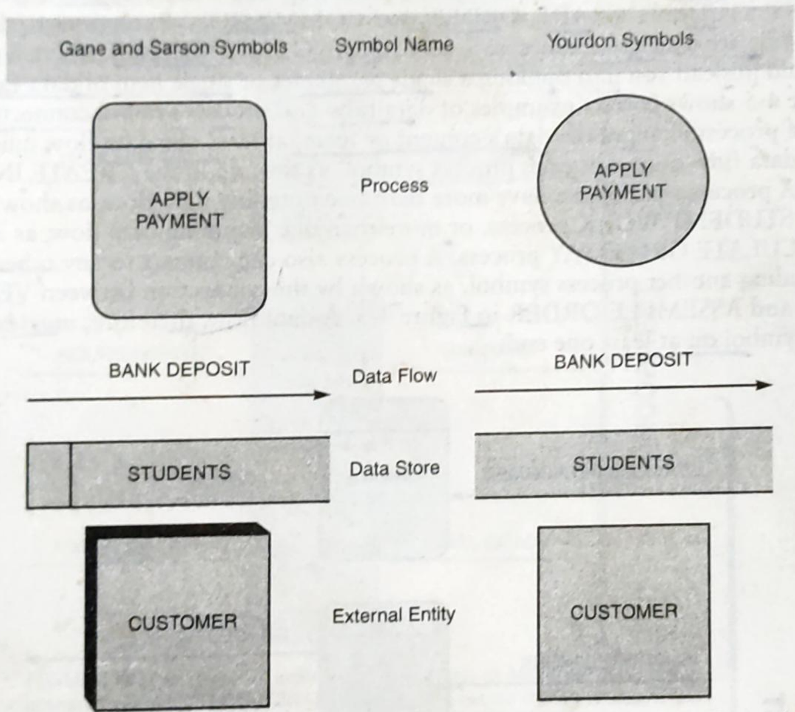


FIGURE 4-3 Data flow diagram symbols, symbol names, and examples of the Gane and Sarson and Yourdon symbol sets.

In DFDs, a process symbol can be referred to as a black box, because the inputs, outputs, and general functions of the process are known, but the underlying details and logic of the process are hidden. By showing processes as black boxes, an analyst can create DFDs that show how the system functions, but avoid unnecessary detail and clutter. When the analyst wishes to show additional levels of detail, he or she can zoom in on a process symbol and create a more in-depth DFD that shows the process's internal workings — which might reveal even more processes, data flows, and data stores. In this manner, the information system can be modeled as a series of increasingly detailed pictures.

The network router shown in Figure 4-4 is an example of a black box. An observer can see cables that carry data into and out of the router, but the router's internal operations are not revealed — only the results are apparent.

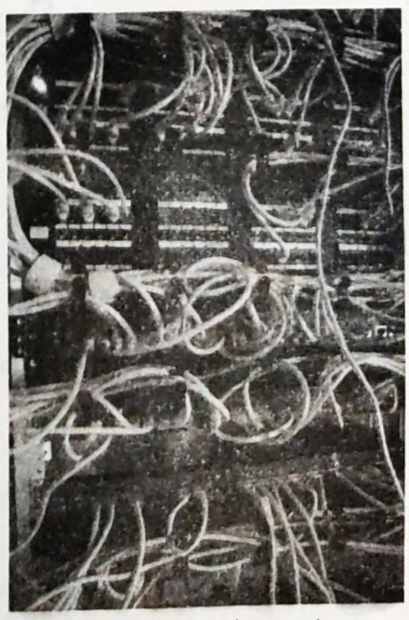


FIGURE 4-4 A network router is an example of a black box. The cables carry data into and out of the router, but the router's internal operations are hidden inside the case.

DATA FLOW SYMBOL A data flow is a path for data to move from one part of the information system to another. A data flow in a DFD represents one or more data items. For example, a data flow could consist of a single data item (such as a student ID number) or it could include a set of data (such as a class roster with student ID numbers, names, and registration dates for a specific class). Although the DFD does not show the detailed contents of a data flow, that information is included in the data dictionary, which is described later in this chapter.

The symbol for a data flow is a line with a single or double arrowhead. The data flow name appears above, below, or alongside the line. A data flow name consists of a singular noun and an adjective, if needed. Examples of data flow names are DEPOSIT, INVOICE PAYMENT, STUDENT GRADE, ORDER, and COMMISSION. Exceptions to the singular name rule are data flow names, such as GRADING PARAMETERS, where a singular name could mislead you into thinking a single parameter or single item of data exists.

Figure 4-5 shows correct examples of data flow and process symbol connections. Because a process changes the data's content or form, at least one data flow must enter and one data flow must exit each process symbol, as they do in the CREATE INVOICE process. A process symbol can have more than one outgoing data flow, as shown in the GRADE STUDENT WORK process, or more than one incoming data flow, as shown in the CALCULATE GROSS PAY process. A process also can connect to any other symbol, including another process symbol, as shown by the connection between VERIFY ORDER and ASSEMBLE ORDER in Figure 4-5. A data flow, therefore, *must* have a process symbol on at least one end.

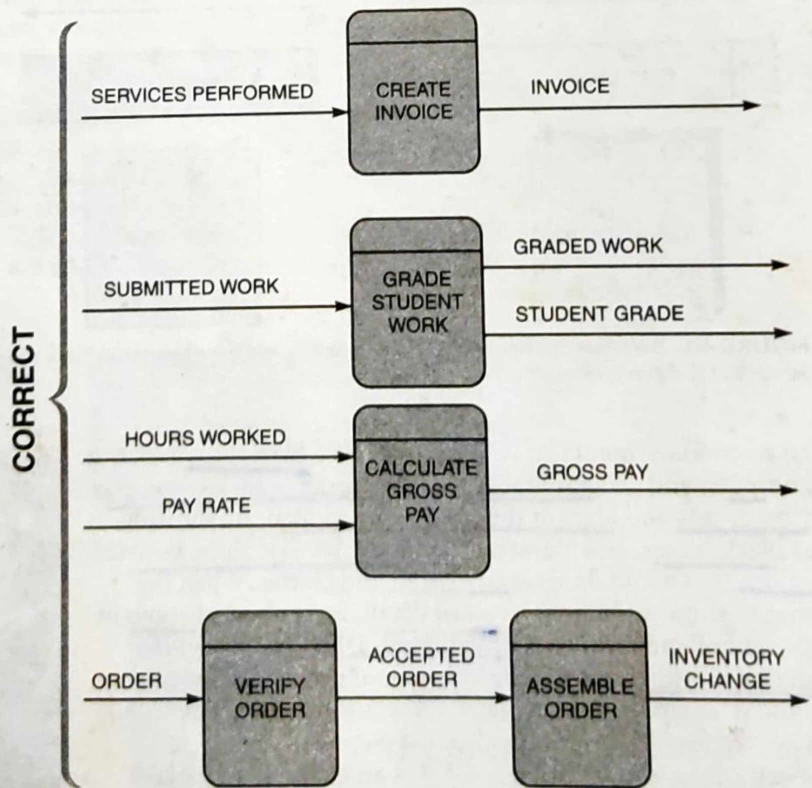


FIGURE 4-5 Examples of correct combinations of data flow and process symbols.

Figure 4-6 shows three data flow and process combinations that you must avoid:

- **Spontaneous generation.** The APPLY INSURANCE PREMIUM process, for instance, produces output, but has no input data flow. Because it has no input, the process is called a spontaneous generation process.
- **Black hole.** The CALCULATE GROSS PAY is called a black hole process, which is a process that has input, but produces no output.

- **Gray hole.** A gray hole is a process that has at least one input and one output, but the input obviously is insufficient to generate the output shown. For example, a date of birth input is not sufficient to produce a final grade output in the CALCULATE GRADE process.

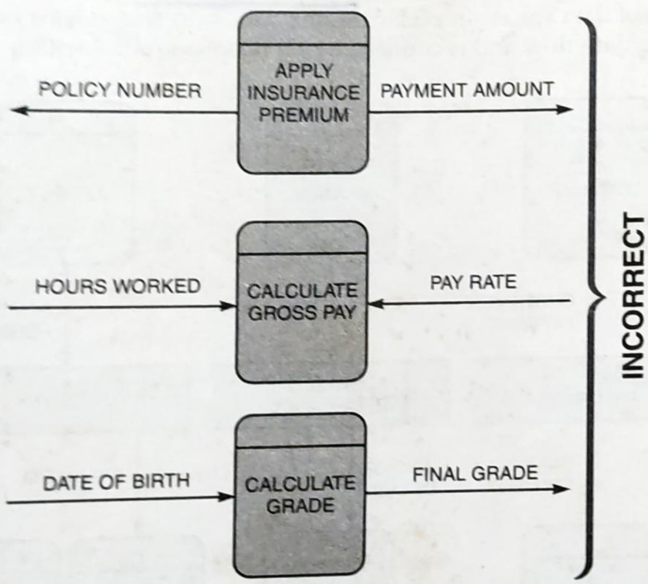


FIGURE 4-6 Examples of incorrect combinations of data flow and process symbols. APPLY INSURANCE PREMIUM has no input and is called a spontaneous generation process. CALCULATE GROSS PAY has no outputs and is called a black hole process. CALCULATE GRADE has an input that is obviously unable to produce the output. This process is called a gray hole.

Spontaneous generation, black holes, and gray holes are impossible logically in a DFD because a process must act on input, shown by an incoming data flow, and produce output, represented by an outgoing data flow.

DATA STORE SYMBOL A data store is used in a DFD to represent data that the system stores because one or more processes need to use the data at a later time. For instance, instructors need to store student scores on tests and assignments during the semester so they can assign final grades at the end of the term. Similarly, a company stores employee salary and deduction data during the year in order to print W-2 forms with total earnings and deductions at the end of the year. A DFD does not show the detailed contents of a data store — the specific structure and data elements are defined in the data dictionary, which is discussed later in this chapter.

The physical characteristics of a data store are unimportant because you are concerned only with a logical model. Also, the length of time that the data is stored is unimportant — it can be a matter of seconds while a transaction is processed or a period of months while data is accumulated for year-end processing. What is important is that a process needs access to the data at some later time.

In a DFD, the Gane and Sarson symbol for a data store is a flat rectangle that is open on the right side and closed on the left side. The name of the data store appears between the lines and identifies the data it contains. A data store name is a plural name consisting of a noun and adjectives, if needed. Examples of data store names are STUDENTS, ACCOUNTS RECEIVABLE, PRODUCTS, DAILY PAYMENTS,

PURCHASE ORDERS, OUTSTANDING CHECKS, INSURANCE POLICIES, and EMPLOYEES. Exceptions to the plural name rule are collective nouns that represent multiple occurrences of objects. For example, GRADEBOOK represents a group of students and their scores.

A data store must be connected to a process with a data flow. Figure 4-7 illustrates typical examples of data stores. In each case, the data store has at least one incoming and one outgoing data flow and is connected to a process symbol with a data flow.

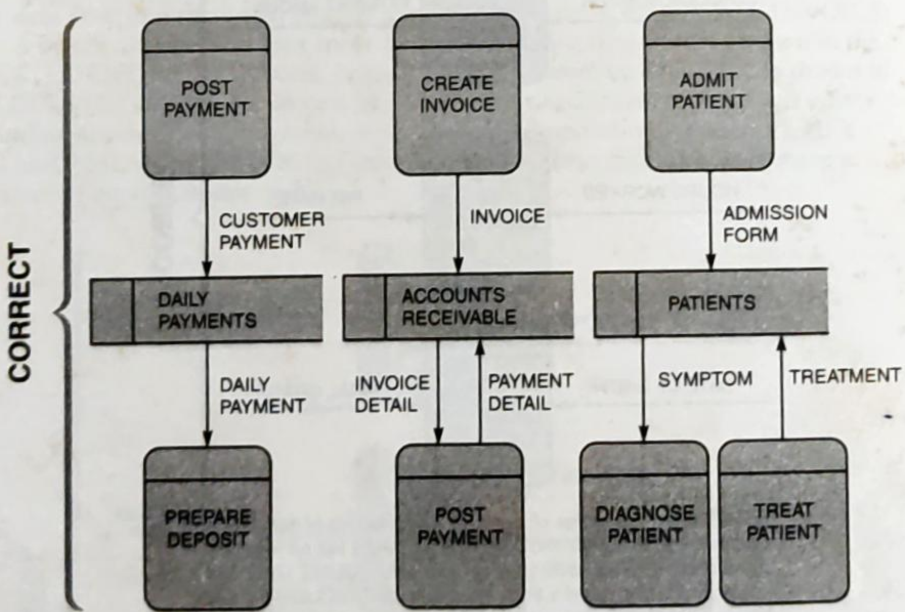


FIGURE 4-7 Examples of correct uses of data store symbols in a data flow diagram.

Violations of the rule that a data store must have at least one incoming and one outgoing data flow are shown in Figure 4-8. In the first example, two data stores are connected incorrectly because no process is between them. Also, COURSES has no incoming data flow and STUDENTS has no outgoing data flow. In the second and third examples, the data stores lack either an outgoing or incoming data flow.

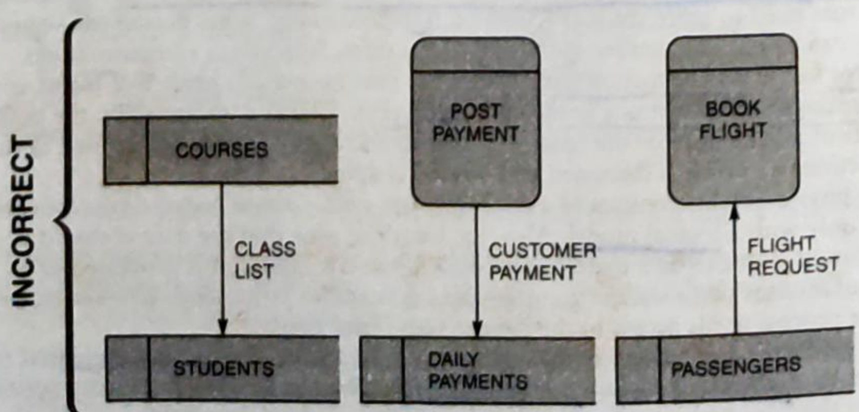


FIGURE 4-8 Examples of incorrect uses of data store symbols: two data stores cannot be connected by a data flow without an intervening process, and each data store should have an outgoing and incoming data flow.

There is an exception to the requirement that a data store must have at least one incoming and one outgoing data flow. In some situations, a data store has no input data flow because it contains fixed reference data that is not updated by the system. For example, consider a data store called TAX TABLE, which contains withholding tax data that a company downloads from the Internal Revenue Service. When the company runs its payroll, the CALCULATE WITHHOLDING process accesses data from this data store. On a DFD, this would be represented as a one-way outgoing data flow from the TAX TABLE data store into the CALCULATE WITHHOLDING process.

ENTITY SYMBOL The symbol for an **entity** is a rectangle, which may be shaded to make it look three-dimensional. The name of the entity appears inside the symbol.

A DFD shows only external entities that provide data to the system or receive output from the system. A DFD shows the boundaries of the system and how the system interfaces with the outside world. For example, a customer entity submits an order to an order processing system. Other examples of entities include a patient who supplies data to a medical records system, a homeowner who receives a bill from a city property tax system, or an accounts payable system that receives data from the company's purchasing system.

DFD entities also are called **terminators**, because they are data origins or final destinations. Systems analysts call an entity that supplies data to the system a **source**, and an entity that receives data from the system a **sink**. An entity name is the singular form of a department, outside organization, other information system, or person. An external entity can be a source or a sink or both, but each entity must be connected to a process by a data flow. Figures 4-9 and 4-10 on the next page show correct and incorrect examples of this rule.

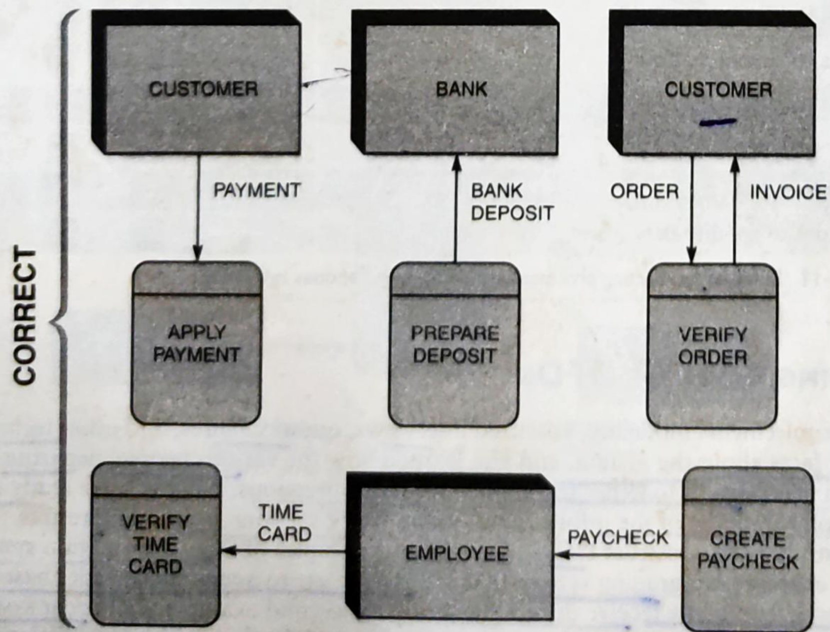


FIGURE 4-9 Examples of correct uses of external entities in a data flow diagram.

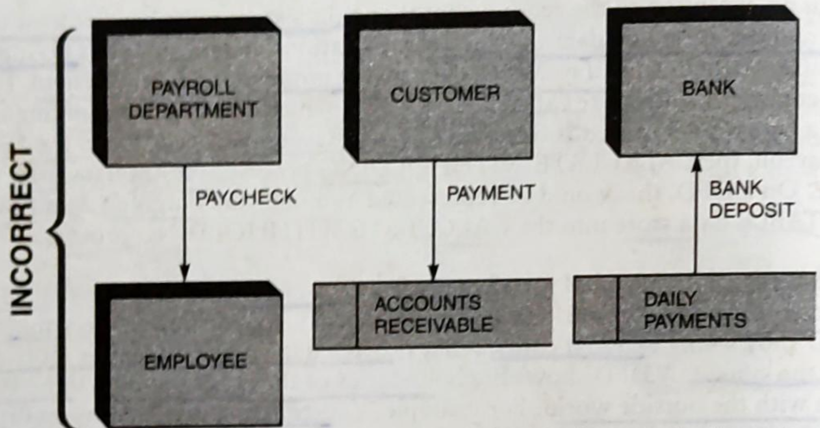


FIGURE 4-10 Examples of incorrect uses of external entities. An external entity must be connected by a data flow to a process, and not directly to a data store or to another external entity.

With an understanding of the proper use of DFD symbols, you are ready to construct diagrams that use these symbols. Figure 4-11 shows a summary of the rules for using DFD symbols.

Data Flow That Connects	*Okay to Use?* YES	NO
A process to another process	✓	
A process to an external entity	✓	
A process to a data store	✓	
An entity to another entity		✓
An entity to a data store		✓
A data store to another data store		✓

FIGURE 4-11 Rules for connecting processes, data stores, and entities in a DFD.

Creating a Set of DFDs

During requirements modeling, you used interviews, questionnaires, and other techniques to gather facts about the system, and you learned how the various people, departments, data, and processes fit together to support business operations. Now you are ready to create a graphical model of the information system based on your fact-finding results.

To learn how to construct DFDs, you will use examples of three information systems. The first example is a grading system that instructors use to assign final grades based on the scores that students receive during the term. The second example is an order system that a company uses to enter orders and apply payments against a customer's balance. The third example is a manufacturing system that handles a company's production.

You will learn how to create a set of DFDs by performing three main tasks:

Step 1: Draw a context diagram
Step 2: Draw a diagram 0 DFD
Step 3: Draw the lower-level diagrams

Draw a Context Diagram

The first step in constructing a set of DFDs is to draw a context diagram. A **context diagram** is a top-level view of an information system that shows the system's boundaries and scope. To draw a context diagram, you start by placing a single process symbol in the center of the page. The symbol represents the entire information system, and you identify it as process 0. Then you place the entities around the perimeter of the page and use data flows to connect the entities to the central process. You do not show any data stores in a context diagram because data stores are internal to the system.

How do you know which entities and data flows to place in the context diagram? You begin by reviewing the system requirements to identify all external data sources and destinations. During that process, you record the name of the entities, the name and content of the data flows, and the direction of the data flows. If you do that carefully, and you did a good job of fact-finding in the previous stage, you should have no difficulty drawing the context diagram.

CONTEXT DIAGRAM EXAMPLE Figure 4-12 shows the context diagram for a grading system. The GRADING SYSTEM process is at the center of the diagram. The three entities (STUDENT RECORDS SYSTEM, STUDENT, and INSTRUCTOR) are placed around the central process. Interaction among the central process and the entities involves six different data flows. The STUDENT RECORDS SYSTEM entity supplies data through the CLASS ROSTER data flow and receives data through the FINAL GRADE data flow. The STUDENT entity supplies data through the SUBMITTED WORK data flow and receives data through the GRADED WORK data flow. Finally, the INSTRUCTOR entity supplies data through the GRADING PARAMETERS data flow and receives data through the GRADE REPORT data flow.

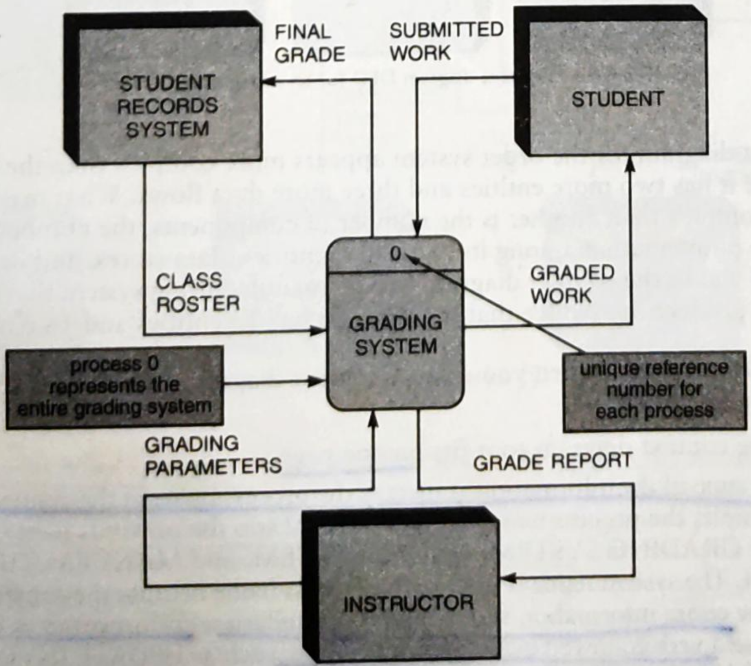


FIGURE 4-12 Context diagram DFD for a grading system.

The context diagram for an order system is shown in Figure 4-13 on the next page. Notice that the ORDER SYSTEM process is at the center of the diagram and five entities surround the process. Three of the entities, SALES REP, BANK, and ACCOUNTING, have single incoming data flows for COMMISSION, BANK DEPOSIT, and

CASH RECEIPTS ENTRY, respectively. The WAREHOUSE entity has one incoming data flow — PICKING LIST — that is, a report that shows the items ordered and their quantity, location, and sequence to pick from the warehouse. The WAREHOUSE entity has one outgoing data flow, COMPLETED ORDER. Finally, the CUSTOMER entity has two outgoing data flows, ORDER and PAYMENT, and two incoming data flows, ORDER REJECT NOTICE and INVOICE.

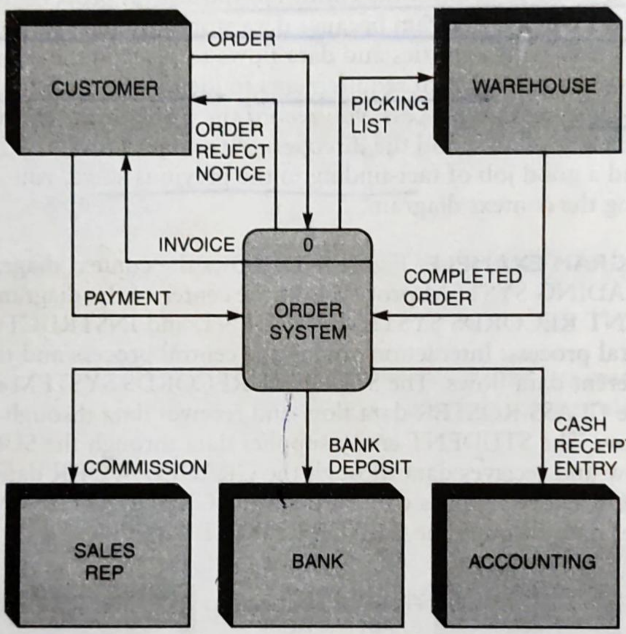


FIGURE 4-13 Context diagram DFD for an order system.

The context diagram for the order system appears more complex than the grading system because it has two more entities and three more data flows. What makes one system more complex than another is the number of components, the number of levels, and the degree of interaction among its processes, entities, data stores, and data flows.

Figure 4-14 shows the context diagram for the manufacturing system that supports the company's production. Notice that the diagram has 13 entities and 18 data flows.

DRAWING GUIDELINES When you draw a context diagram, and other DFDs, you follow certain guidelines:

1. Draw the context diagram so it fits on one page.
2. Use the name of the information system as the process name in the context diagram. For example, the process names in Figures 4-12 (on the previous page), 4-13, and 4-14 are GRADING SYSTEM, ORDER SYSTEM, and MANUFACTURING SYSTEM. The system name is used as the process name because the context diagram shows the entire information system and its boundaries. For processes in lower-level DFDs, use a verb followed by a descriptive noun, such as UPDATE INVENTORY, CALCULATE OVERTIME, or PRODUCE REPORT.
3. Use unique names within each set of symbols. For instance, the diagram in Figure 4-14 shows only one entity named CUSTOMER and only one data flow named ORDER. Whenever you see the entity CUSTOMER on one of the other manufacturing system DFDs, you know that you are dealing with the same entity. Whenever the ORDER data flow appears, you know that you are dealing with the same data flow. The naming convention also applies to data stores.

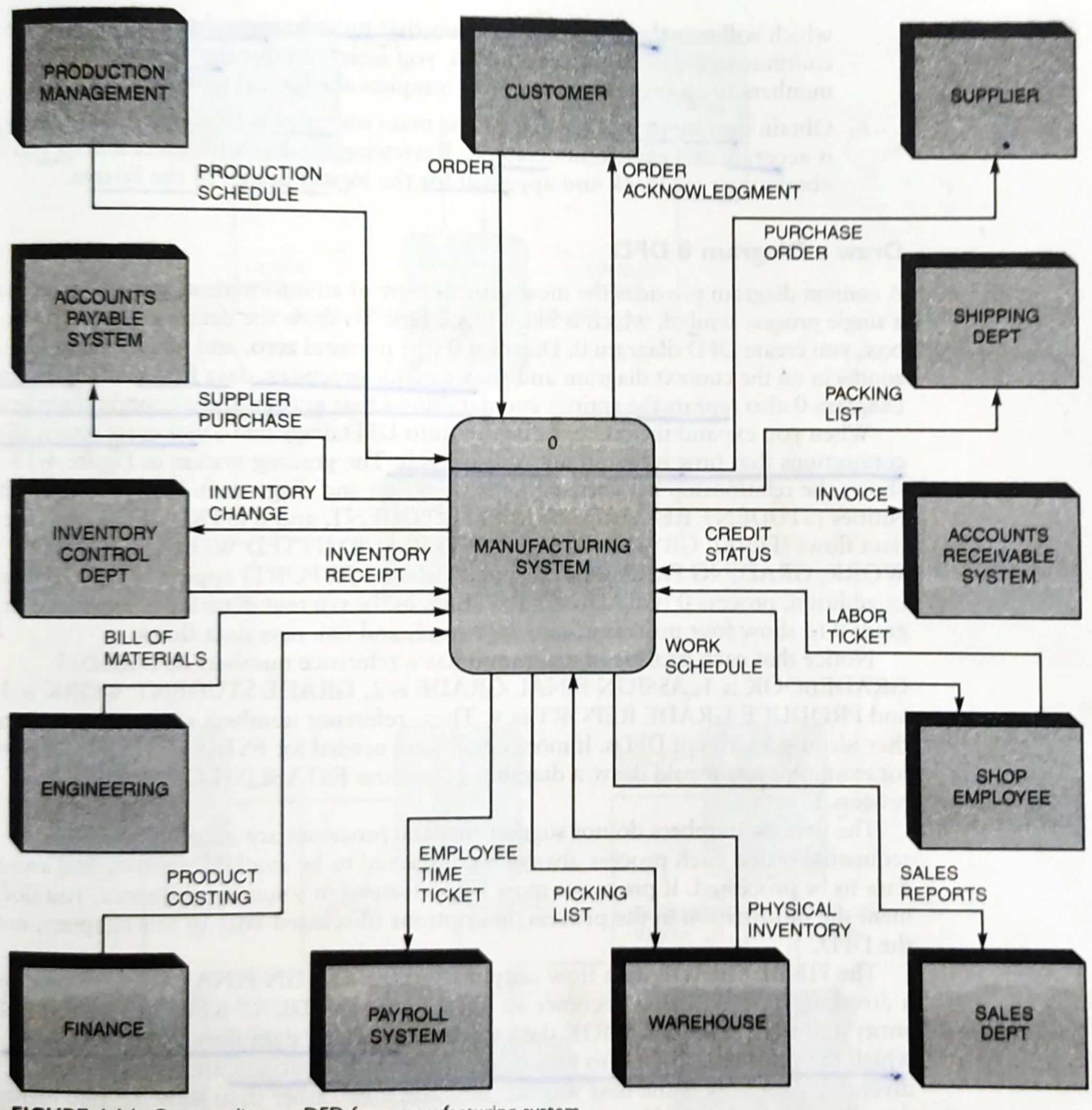


FIGURE 4-14 Context diagram DFD for a manufacturing system.

4. Do not cross lines. One way to achieve that goal is to restrict the number of symbols in any DFD. On lower-level diagrams with multiple processes, you should not have more than nine process symbols. Including more than nine symbols usually is a signal that your diagram is too complex and that you should reconsider your analysis. Another way to avoid crossing lines is to duplicate an entity or data store. When duplicating a symbol on a diagram, make sure to document the duplication to avoid possible confusion. A special notation, such as an asterisk, next to the symbol name and inside the duplicated symbols signifies that they are duplicated on the diagram.
5. Provide a unique name and reference number for each process. Because it is the highest-level DFD, the context diagram contains process 0, which represents the entire information system, but does not show the internal workings. To describe the next level of detail inside process 0, you must create a DFD named diagram 0,

which will reveal additional processes that must be named and numbered. As you continue to create lower-level DFDs, you assign unique names and reference numbers to all processes, until you complete the logical model.

6. Obtain user input and feedback. The main objective is to ensure that your model is accurate and easy to understand. Reviewing models with users allows you to obtain their feedback and approval for the logical design of the system.

Draw a Diagram 0 DFD

A context diagram provides the most general view of an information system and contains a single process symbol, which is like a black box. To show the detail inside the black box, you create DFD diagram 0. **Diagram 0** (the numeral zero, and not the letter O) zooms in on the context diagram and shows major processes, data flows, and data stores. Diagram 0 also repeats the entities and data flows that appear in the context diagram.

When you expand the context diagram into DFD diagram 0, you must retain all the connections that flow into and out of process 0. The grading system in Figure 4-15 shows the relationship between a context diagram and diagram 0. Notice that the three entities (STUDENT RECORDS SYSTEM, STUDENT, and INSTRUCTOR) and the six data flows (FINAL GRADE, CLASS ROSTER, SUBMITTED WORK, GRADED WORK, GRADING PARAMETERS, and GRADE REPORT) appear in both diagrams. In addition, process 0 (GRADING SYSTEM) in the context diagram is expanded in diagram 0 to show four processes, one data store, and five new data flows.

Notice that each process in diagram 0 has a reference number: ESTABLISH GRADEBOOK is 1, ASSIGN FINAL GRADE is 2, GRADE STUDENT WORK is 3, and PRODUCE GRADE REPORT is 4. These reference numbers are important because they identify a series of DFDs. If more detail were needed for ESTABLISH GRADEBOOK, for example, you would draw a diagram 1, because ESTABLISH GRADEBOOK is process 1.

The process numbers do not suggest that the processes are accomplished in a sequential order. Each process always is considered to be available, active, and awaiting data to be processed. If processes must be performed in a specific sequence, you document the information in the process descriptions (discussed later in this chapter), not in the DFD.

The FINAL GRADE data flow output from the ASSIGN FINAL GRADE process is a diverging data flow that becomes an input to the STUDENT RECORDS SYSTEM entity and to the GRADEBOOK data store. A **diverging data flow** is a data flow in which the same data travels to two or more different locations. In that situation, a diverging data flow is the best way to show the flow rather than showing two identical data flows, which could be misleading.

If the same data flows in both directions, you can use a double-headed arrow to connect the symbols. To identify specific data flows into and out of a symbol, however, you use separate data flow symbols with single arrowheads. For example, in Figure 4-15 the separate data flows (SUBMITTED WORK and GRADED WORK) go into and out of the GRADE STUDENT WORK process.

Because diagram 0 is a more detailed or expanded version of process 0 on the context diagram, diagram 0 represents an exploded view of process 0. You also can refer to diagram 0 as a partitioned or decomposed view of process 0. When you explode a DFD, the higher-level diagram is called the **parent diagram**, and the lower-level diagram is referred to as the **child diagram**. The grading system is simple enough that you do not need any additional DFDs to model the system. At that point, the four processes, the one data store, and the 10 data flows can be documented in the data dictionary.

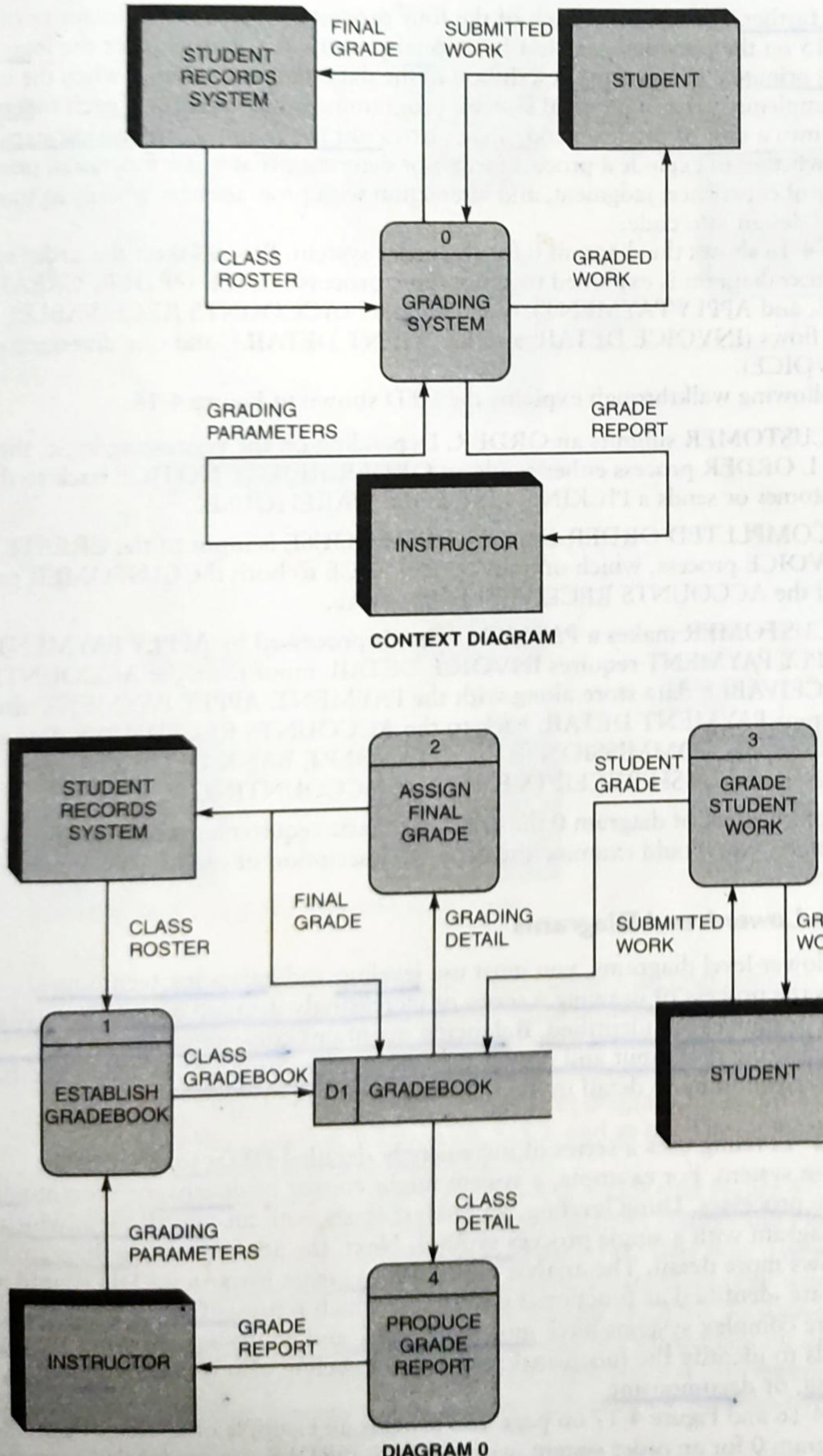


FIGURE 4-15 Context diagram and diagram 0 for the grading system.

When you create a set of DFDs for a system, you break the processing logic down into smaller units, called functional primitives, that programmers will use to develop code. A functional primitive is a process that consists of a single function that is not exploded further. For example, each of the four processes shown in the lower portion of Figure 4-15 on the previous page is a functional primitive. You document the logic for a functional primitive in a process description in the data dictionary. Later, when the logical design is implemented as a physical system, programmers will transform each functional primitive into a unit of program code that carries out the required processing steps. Deciding whether to explode a process further or determine that it is a functional primitive is a matter of experience, judgment, and interaction with programmers who must translate the logical design into code.

Figure 4-16 shows the diagram 0 for the order system. Process 0 on the order system's context diagram is exploded to show three processes (FILL ORDER, CREATE INVOICE, and APPLY PAYMENT), one data store (ACCOUNTS RECEIVABLE), two new data flows (INVOICE DETAIL and PAYMENT DETAIL), and one diverging data flow (INVOICE).

The following walkthrough explains the DFD shown in Figure 4-16:

1. A CUSTOMER submits an ORDER. Depending on the processing logic, the FILL ORDER process either sends an ORDER REJECT NOTICE back to the customer or sends a PICKING LIST to the WAREHOUSE.
2. A COMPLETED ORDER from the WAREHOUSE is input to the CREATE INVOICE process, which outputs an INVOICE to both the CUSTOMER process and the ACCOUNTS RECEIVABLE data store.
3. A CUSTOMER makes a PAYMENT that is processed by APPLY PAYMENT. APPLY PAYMENT requires INVOICE DETAIL input from the ACCOUNTS RECEIVABLE data store along with the PAYMENT. APPLY PAYMENT also outputs PAYMENT DETAIL back to the ACCOUNTS RECEIVABLE data store and outputs COMMISSION to the SALES DEPT, BANK DEPOSIT to the BANK, and CASH RECEIPTS ENTRY to ACCOUNTING.

The walkthrough of diagram 0 illustrates the basic requirements of the order system. To learn more, you would examine the detailed description of each separate process.

Draw the Lower-Level Diagrams

To create lower-level diagrams, you must use leveling and balancing techniques. **Leveling** is the process of drawing a series of increasingly detailed diagrams, until all functional primitives are identified. **Balancing** maintains consistency among a set of DFDs by ensuring that input and output data flows align properly. Leveling and balancing are described in more detail in the following sections.

LEVELING Leveling uses a series of increasingly detailed DFDs to describe an information system. For example, a system might consist of dozens, or even hundreds, of separate processes. Using leveling, an analyst starts with an overall view, which is a context diagram with a single process symbol. Next, the analyst creates diagram 0, which shows more detail. The analyst continues to create lower-level DFDs until all processes are identified as functional primitives, which represent single processing functions. More complex systems have more processes, and analysts must work through many levels to identify the functional primitives. Leveling also is called **exploding**, **partitioning**, or **decomposing**.

Figure 4-16 and Figure 4-17 on page 162 provide an example of leveling. Figure 4-16 shows diagram 0 for an order system, with the FILL ORDER process labeled as process 1. Now consider Figure 4-17, which is an exploded view of the FILL ORDER process. Notice that FILL ORDER (process 1) actually consists of three processes: VERIFY ORDER (process 1.1), PREPARE REJECT NOTICE (process 1.2), and ASSEMBLE ORDER (process 1.3).