Chapter 3

Documenting Accounting Information Systems

After reading this chapter, you will:
1. *Understand* why documenting an AIS is important.
2. *Be able to draw* simple document flowcharts and explain how they describe the flow of data in AISs.
3. *Be able to draw* simple document flowcharts, system flowcharts, process maps, and data flow diagrams.
4. *Know* how program flowcharts and decision tables help document AISs.
5. *Be able to explain* the importance of end-user documentation.
6. *Be aware of* software available for documenting AISs and helping companies comply with the Sarbanes-Oxley Act.

"Although documentation of [business] processes needs to be detailed, it also must be clear and easy to follow . . . .If people cannot easily explain it, or someone can't pick up the documentation and understand it, you have not successfully completed the task."

Regina Baraban, ''Look Out for SOX'' *Meetingsnet.com*
Vol. 24, No. 2, (February 2005), pp. 23–24.

INTRODUCTION

Documentation explains how AISs operate and is therefore a vital part of any accounting system. For example, documentation describes the tasks for recording accounting data, the procedures that users must perform to operate computer applications, the processing steps that AISs follow, and the logical and physical flows of accounting data through the system. This chapter explains in greater detail why accountants need to understand documentation and describes some tools for diagramming complex systems.

Accountants can use many different types of logic charts to trace the flow of accounting data through an AIS. For example, document flowcharts describe the physical flow of order forms, requisition slips, and similar hard-copy documents through an AIS. These flowcharts pictorially represent data paths in compact formats and therefore save pages of narrative description. System flowcharts are similar to document flowcharts, except that system flowcharts usually focus on the electronic flows of data in computerized AISs. Other examples of documentation aids include process maps, data flow diagrams, program flowcharts, and decision tables. This chapter describes all of these documentation aids, as well as some computerized tools for creating them.

Today, many end users develop computer applications for themselves. This end-user programming is very helpful to managers, who consequently do not require IT professionals to develop simple word processing, spreadsheet, or database applications. But end-user programming can also be a problem because many employees do not know how to document their work properly or simply don't do so. The final section of this chapter examines the topic of end-user programming and documentation in greater detail.

WHY DOCUMENTATION IS IMPORTANT

Accountants do not need to understand exactly how computers process the data of a particular accounting application, but it is important for them to understand the documentation that describes how this processing takes place. In fact a recent survey of practitioners found that system documentation has become increasingly important as organizations seek to better understand their own business processes and also comply with legislation that requires this understanding, such as the Sarbanes-Oxley Act.[1] **Documentation** includes all the flowcharts, narratives, and other written communications that describe the inputs,

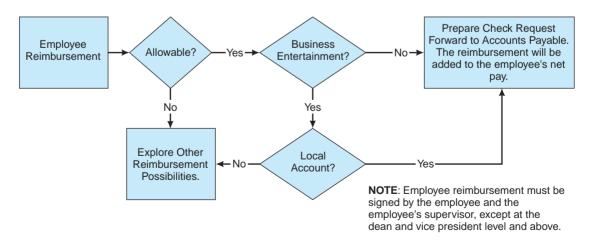---

[1]Bradford, Marianne, Sandra B. Richtermeyer, and Douglas F. Roberts, ''System Diagramming Techniques: An Analysis of Methods Used in Accounting Education and Practice,'' *Journal of Information Systems* Vol. 21, Iss. 1 (Spring 2007), p. 173.

processing, and outputs of an AIS. Documentation also describes the logical flow of data within a computer system and the procedures that employees must follow to accomplish application tasks. Here are nine reasons why documentation is important to AISs.

1. Depicting how the system works.   Simply observing large AISs in action is an impractical way to learn about them, even if they are completely manual. In computerized systems, this task is impossible because the processing is electronic and therefore invisible. On the other hand, studying written descriptions of the inputs, processing steps, and outputs of the system make the job easier, and a few graphs or diagrams of these processing functions makes things easier still. This is one purpose of documentation—to help explain how an AIS operates. Documentation helps employees understand how a system works, assists accountants in designing controls for it, and gives managers confidence that it will meet their information needs.

The Internet contains many examples of flowcharts or logic diagrams that help individuals understand unfamiliar tasks or processes. For example, some universities use them to show students what classes to take and when they should take them to complete their majors in a timely manner. The University of Washington has flowcharts that show how to obtain grants and other types of funding. The University of Illinois at Urbana-Champaign uses elaborate diagrams to depict what happens when a faculty member's employment terminates. Figure 3-1 is a logic diagram from the University of Arizona website that shows employees how to file a claim for reimbursement. If the employee would like additional information for any step in the process, a click of the mouse on the appropriate flowchart symbol reveals additional information.

2. Training users.   Documentation also includes the user guides, manuals, and similar operating instructions that help people learn how an AIS operates. Employees usually do not like to read the user manuals that typically accompany application software, but these instructional materials are invaluable reference aids when they are needed. Whether distributed manually in hard-copy format or electronically in the familiar Help files or ''get-started tours'' of microcomputer applications, these documentation aids help train users to operate AIS hardware and software, solve operational problems, and perform their jobs better.

FIGURE 3-1   Example of a flowchart used at the University of Arizona to help employees file a reimbursement claim. For additional information, individuals simply click on the appropriate symbol.

3. Designing new systems.   Documentation helps system designers develop new systems in much the same way that blueprints help architects design buildings. For example, professional IT personnel commonly hold structured walkthroughs in which they review system documentation to ensure the integrity and completeness of their designs, and to identify design flaws. Well-written documentation, along with other systems-design methodologies, often plays a key role in reducing systems failures and decreasing the time spent correcting ''emergency errors'' in computer systems. Conversely, poorly-designed systems usually lead to critical mistakes and expensive write-offs.

4. Controlling system development and maintenance costs.   Personal computer applications typically employ prewritten, off-the-shelf software that is relatively reliable and inexpensive. In contrast, custom-developed business systems often cost millions of dollars and can be less reliable. Good documentation helps system designers develop **object-oriented software**, that is, programs that contain modular, reusable code. This object-orientation helps programmers avoid writing duplicate programs and facilitates changes when programs must be modified later. If you have ever replaced a specialized part in your car, you have some idea of how frustrating, time-consuming, and expensive ''non-standardization'' can be, and therefore how useful object-oriented programming might be to business organizations.

5. Standardizing communications with others.   The usefulness of narrative descriptions can vary significantly, and a reader can interpret such descriptions differently from what the writer intended. Documentation aids such as system flowcharts or data flow diagrams are standard industry tools, and they are more likely to be interpreted the same way by all parties viewing them. Thus, documentation tools are important because they help describe an existing or proposed system in a ''common language'' and help users communicate with one another about these systems.

Case-in-Point 3.1   The fourth largest public healthcare system in the United States, Carolinas HealthCare System (CHS), uses business process management software from Staffware to automate and streamline its revenue cycle processes and improve customer service for their patients and doctors. As a result of using this software, CHS can now service patient accounts much faster and has increased the speed and accuracy of their billing office operations.[2]

6. Auditing AISs.   Documentation helps depict audit trails. When investigating an AIS, for example, the auditors typically focus on internal controls. In such circumstances, documentation helps auditors determine the strengths and weaknesses of a system's controls, and therefore the scope and complexity of the audit. Similarly, the auditors will want to trace sample outputs to the original transactions that created them (e.g., tracing inventory assets back to original purchases). System documentation helps auditors perform these tasks.

7. Documenting business processes.   Accounting systems automatically create a record of some organization's processes because they capture financial data as they occur. A study of these processes can lead to better systems. Thus, in mapping these processes, documentation can help managers better understand the ways in which their businesses

---

[2] ''Carolinas HealthCare System Uses Staffware Software to Streamline Business Processes,'' *Business Wire* February 23, 2004, p. 53.

operate, what controls are involved or missing from critical organizational activities, and how to improve core business processes.

8. Complying with the Sarbanes-Oxley Act.   Section 404 of the Sarbanes-Oxley Act of 2002 (SOX) requires publicly-traded companies to identify the major sources of business risks, *document their internal control procedures*, and hire external auditors to evaluate the validity and effectiveness of such procedures. Documentation is therefore crucial for analyzing the risks of errors, frauds, omissions, and similar mistakes in important business processes, as well as helping auditors evaluate the controls used to mitigate such risks—i.e., some of the major tasks required by SOX.

Almost everyone acknowledges that the costs of complying with SOX are enormous, and many also believe that SOX gave documentation ''a new life.'' To save money, many companies now use software packages to help them automate SOX documentation tasks. We describe some examples of such software in a later section of this chapter.

Case-in-Point 3.2   FirstEnergy Corporation in Akron, Ohio, generates and distributes electrical energy to customers in a 36,100-square-mile area of Ohio, Pennsylvania, and New Jersey. Certus Governance Suite, a software package, helps company managers comply with many SOX documentation requirements, better understand the company's internal controls, and saves ''looking at hundreds of spreadsheets.''[3]

9. Establishing accountability.   Manual signatures on business and government documents allow employees and government agents to execute their responsibilities, create audit trails, and establish accountability for their actions. An example is a **signed checklist** that outlines the month-end journal entries an accountant must perform. Such checklists verify that the accountant performed these tasks, that a reviewer approved them, and that both individuals are accountable for the accuracy of the work. Similar comments apply to the checklists for preparing financial statements, tax returns, auditing papers, budgets, and similar accounting documents. Including such checklists with the statements themselves both documents the work that the employees performed as well as the procedures and controls *involved* in the work.

Case-in-Point 3.3   In a parallel universe, the master accountant for the great warlord took on an apprentice. One day, the apprentice approached his mentor and asked ''Master, is it always necessary to document accounting systems?''

The master accountant answered ''No. Such documentation is only needed by those who cannot feel the complete tao and beauty of such systems.''

The apprentice replied ''Who *can* feel such beauty?''

The master accountant responded ''I do not know anyone with such abilities.''

DOCUMENT AND SYSTEM FLOWCHARTS

Despite the many reasons why documentation is important, most organizations find that they document less than they should. One explanation for this deficiency is that organizations often create or implement large AISs under tight deadlines. In such cases, the urgency to develop ''a system that works'' overrides the need for ''a system that

---

[3]Thomas Hoffman, ''Calibrating Toward Compliance'' *Computerworld* Vol. 40, No. 6 (February 6, 2006), pp. 21–24.

is well-documented.'' Another reason is that most IT professionals much prefer creating systems to documenting them. Thus, many developers actively resist it, arguing that they will ''get around to it later'' or that documenting is a job for non-existent assistants.

The record suggests that insufficient or deficient documentation costs organizations time and money and that good documentation is as important as the good software it describes. What tools are available to document AISs? Two examples are document flowcharts and system flowcharts.

Document Flowcharts

A **document flowchart** traces the physical flow of documents through an organization—i.e., from the departments, groups, or individuals who first create them to their final dispositions. Figure 3-2 illustrates common document flowcharting symbols, and the examples below illustrate how to use them to create simple document flowcharts.

Constructing a document flowchart begins by identifying the different departments or groups that handle the documents of a particular system. The flowcharter then uses the symbols in Figure 3-2 to illustrate the document flows. Let us first examine two simple cases and then discuss some general flowcharting guidelines.
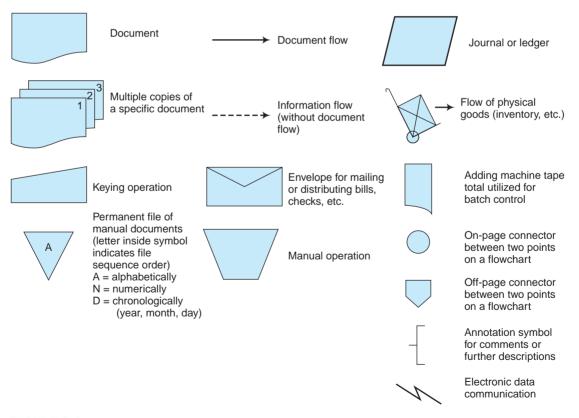

FIGURE 3-2 Common document flowcharting symbols.

Example 1.   Your boss asks you to document the paperwork involved in acquiring office supplies from your company's Central Supplies Department. Your administrative assistant explains the process as follows:

Reordering supplies requires a requisition request. When I need more stationery, for example, I fill out two copies of a goods requisition form (GRF). I send the first copy to central supplies and file the second copy here in the office.

There are two departments involved in this example—your department (which we shall call the Requesting Department) and the Central Supplies Department. Thus, you should begin by naming these departments in headings on your document flowchart (Figure 3-3). Next, you draw two copies of the GRF under the heading for the Requesting Department because this is the department that creates this form. You number these copies 1 and 2 to indicate two copies.

Finally, you indicate where each document goes: copy 1 to the Central Supplies Department and copy 2 to a file in the Requesting Department. A document's first appearance should be in the department that creates it. A solid line or the on-page connectors shown here indicates its physical transmittal from one place to another. You should then redraw the transmitted document to indicate its arrival at the department that receives it. Figure 3-3 illustrates the completed flowchart for this narrative.

Example 2.   Let us now consider a slightly more complex example—the task of hiring a new employee at your company. The process begins when a department develops a vacancy. The Human Resources (HR) director explains the process as follows:

The department that develops a vacancy must first complete a job vacancy form, which it forwards to my department. We then advertise for the position and, with the help of the requesting department, interview applicants. When the vacancy is filled, the HR Department prepares a position hiring form (PHF) in triplicate. We file the first copy in a manual file, which is organized by employee Social Security
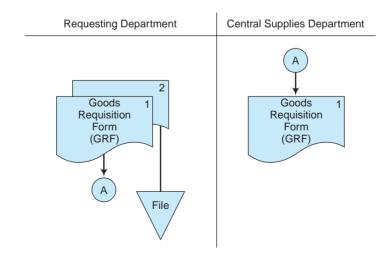
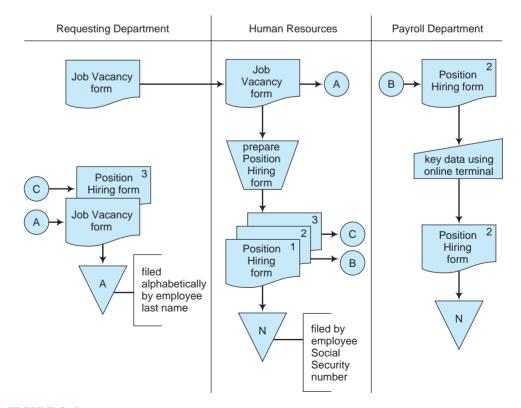**FIGURE 3-3**   A simple document flowchart.

*number. We staple the third copy to the job vacancy form and return it to the
Requesting Department, where clerks file it alphabetically by employee last name.*

*The HR Department forwards the second copy of the PHF to the Payroll
Department. The Payroll Department uses the form as an authorization document
to create a payroll record for the new employee. Thus, the information on the form
is keyed directly into the company's computer system using an online terminal
located in the payroll office. This copy of the PHF is then filed numerically for
reference and also as evidence that the form has been processed.*

Figure 3-4 is a document flowchart for this example. To draw it, your first step is the
same as before—to identify the participants. In this case there are three of them: (1) the
department with the job vacancy (i.e., the Requesting Department in Figure 3-4), (2) the
Human Resources Department, and (3) the Payroll Department. You identify each of these
departments in separate columns at the top of the document flowchart.

Your next step is to identify the documents involved. There are two major ones:
(1) the Job Vacancy form, which we presume is prepared as a single copy, and (2) the
Position Hiring form, which we are told is prepared in triplicate. In practice, multiple-copy
forms are usually color-coded. However, in document flowcharts, usually these are simply
numbered and a separate page is attached to explain the color-number equivalencies.

Your third step is to indicate where the documents are created, processed, and
used. This is probably the most difficult task, and a document flowchart designer
must often use considerable ingenuity to represent data flows and processing activities

FIGURE 3-4 A document flowchart illustrating the flow of documents involved in the hiring of
a new employee.

accurately. Figure 3-4 illustrates these flows for the hiring procedures just described. Where there are a large number of document transmittals, you can use on-page connectors (circles) to connect document flows from one place on a page to another and avoid complicated flow lines. Thus, Figure 3-4 uses several on-page connectors (with letters A, B, and C) to avoid cluttering the drawing and shows the completed document flowchart. You should use a unique identifier in each connector (such as a letter) for identification purposes. You can also use off-page connectors (to connect data flows to other pages) if necessary.

Guidelines for Drawing Document Flowcharts

Document flowcharts concentrate on the physical flow of reports and similar documents. When constructing them, some analysts also include any movement of physical goods in their document flowcharts—e.g., moving inventory from a receiving department to an inventory storeroom. (Document flowcharts typically use hand-truck symbols for this task.) Some document flowcharts also illustrate information flows that do not involve documents (for example, a sales clerk telephoning to check a customer's account balance before approving a credit sale). Thus, the term ''document'' broadly includes all types of organizational communications and data flows.
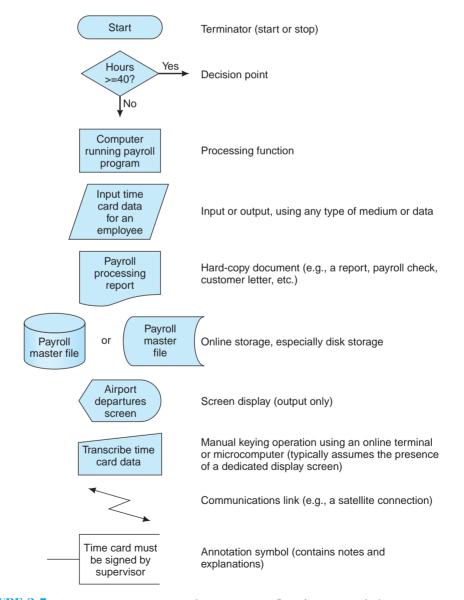
Unlike other types of symbols—for example, the system and program flowcharting symbols discussed later in this chapter—document flowcharting symbols are not standardized. But even though creating document flowcharts is more an art than a science, you can use the following guidelines to make these flowcharts clearer.

1. Identify all the departments that create or receive the documents involved in the system. Use vertical lines to create ''swim lanes'' to separate each department from the others.
2. Carefully classify the documents and activities of each department, and draw them under their corresponding department headings.
3. Identify each copy of an accounting document with a number. If multiple-copy documents are color-coded, use a table to identify the number-color associations.
4. Account for the distribution of each copy of a document. In general, it is better to over-document a complicated process than to under-document it.
5. Use on-page and off-page connectors to avoid diagrams with lines that cross one another.
6. Each pair of connectors (a ''from'' and a ''to'' connector in each pair) should use the same letter or number.
7. Use annotations if necessary to explain activities or symbols that may be unclear. These are little notes to the reader that help clarify your documentation.
8. If the sequence of records in a file is important, include the letter ''A'' for alphabetical, ''N'' for numeric, or ''C'' for chronological in the file symbol. As indicated in guideline 7, you can also include a note in the flowchart to make things clearer.
9. Most employees reference forms with acronyms (e.g., GRF or PHF in the preceding examples). To avoid confusion, use full names (possibly with acronyms in parentheses) or create a table of equivalents to ensure accuracy in identifying such forms.
10. Consider using automated flowcharting tools. See the section of this chapter on CASE tools.

Case-in-Point 3.4 Accountants disagree about the usefulness of document flowcharts relative to other documenting tools, but one manuscript reviewer of this book wrote: "Flowcharting is one of the most essential skills, in my opinion, for a student to learn in a systems course. During my tenure at a CPA firm, I had the opportunity to document several accounting information systems, and document flowcharting was the key skill. When word got around the office that I was a good flowcharter, I got placed on more important clients, furthering my career."

System Flowcharts

Whereas document flowcharts focus on tangible documents, **system flowcharts** concentrate on the computerized data flows of AISs. Thus, a system flowchart typically depicts the electronic flow of data and processing steps in an AIS. Figure 3-5 illustrates some common

FIGURE 3-5 Some common system and programming flowcharting symbols.

system flowcharting symbols. Most of these symbols are industry conventions that have been standardized by the National Bureau of Standards (Standard $\times$ 3.5), although additional symbols are now necessary to represent newer data transmission technologies—for example, wireless data flows.

Some system flowcharts are general in nature and merely provide an overview of the system. These *are high-level system flowcharts*. Figure 3-6 is an example. The inputs and outputs of the system are specified by the general input and output symbol, a parallelogram. In more detailed system flowcharts, the specific form of these inputs and outputs would be indicated—for example, by magnetic disk symbols.

Figure 3-6 refers to only one process—preparing a payroll. A more detailed system flowchart would describe all the processes performed by the payroll program and the specific inputs and outputs of each process. At the lowest, most-detailed level of such documentation are program flowcharts that describe the processing logic of each application program. We will examine program flowcharts later in this chapter.

Like document flowcharts, the process of drawing system flowcharts is probably best understood by studying an illustration. Figure 3-7 is a system flowchart for the following example.

The Sarah Stanton Company is a magazine distributor that maintains a file of magazine subscribers for creating monthly mailing labels. Magazine subscribers mail change-of-address forms or new-subscription forms directly to the company, where input personnel key the information into the system through online terminals. The computer system temporarily stores this information as a file of address-change or new-subscription requests. Clerical staff keys these data into computer files continuously, so we may characterize it as "daily processing."

Once a week, the system uses the information in the daily processing file to update the subscriber master file. At this time, new subscriber names and addresses are added to the file, and the addresses of existing subscribers who have moved are changed. The system also prepares a Master File Maintenance Processing Report to indicate what additions and modifications were made to the file. Once a month, the company prepares postal labels for the magazine's mailing. The subscriber master file serves as the chief input for this computer program. The two major outputs are the labels themselves and a Mailing Labels Processing Report that documents this run and indicates any problems.

FIGURE 3-6   A high-level system flowchart for payroll processing.

FIGURE 3-7 A system flowchart illustrating the computer steps involved in maintaining a subscriber master file and creating monthly mailing labels.

The system flowchart in Figure 3-7 documents the flow of data through the company's computerized system. Thus, it identifies sources of data, the places where data are temporarily stored, and the outputs on which processed data appear. In Figure 3-7, for example, the system flowchart begins with the subscriber request forms and documents the flow of data on these forms through the keying phase, master-file-maintenance phase, and finally, the monthly mailing phase.

Indirectly, system flowcharts also indicate processing cycles (daily, weekly, or monthly), hardware needs (e.g., disk drives and printers), areas of weak or missing application controls, and potential bottlenecks in processing (e.g., manual data entry). In Figure 3-7, we can also identify the major files of the system (a temporary log file of change-request records and a subscriber master file) and the major reports of the system. Finally, note that each processing phase of a system flowchart usually involves preparing one or more control reports. These reports provide processing-control information (e.g., counts of transactions processed) for control purposes and exceptions information (e.g., the identity of unprocessed transactions) that helps employees correct the errors detected by the system.

Guidelines for Drawing System Flowcharts

System flowcharts depict an electronic **job stream** of data through the various processing phases of an AIS, and therefore also illustrate audit trails. Each time the records of a file are sorted or updated, for example, a system flowchart should show this in a separate processing step. Generally speaking, this is the way processing proceeds in almost all AISs, one step at a time, and is therefore the way system flowcharts must portray processing phases. In recognizing the usefulness of system flowcharts, both the American Institute of Certified Public Accountants (AICPA) and the Institute of Management Accountants (IMA) consistently include test questions in their professional examinations, which require a working knowledge of system flowcharts.

Although no strict rules govern exactly how to construct a system flowchart, the following list provides some guidelines.

1. System flowcharts should read from top to bottom and from left to right. In drawing or reading such flowcharts, you should begin in the upper-left corner.
2. Because system flowcharting symbols are standardized, you should use these symbols when drawing your flowcharts—do not make up your own.
3. A processing symbol should always be found between an input symbol and an output symbol. This is called the **sandwich rule**.
4. Use on-page and off-page connectors to avoid crossed lines and cluttered flowcharts.
5. Sketch a flowchart before designing the final draft. Graphical documentation software tools (discussed shortly) make this job easier.
6. Add descriptions and comments in flowcharts to clarify processing elements. You can place these inside the processing symbols themselves, include them in annotation symbols attached to process or file symbols, or add them as separate notes on your system's documentation.

PROCESS MAPS AND DATA FLOW DIAGRAMS

Like system and document flowcharts, process maps and data flow diagrams (DFDs) document the flow of data through an AIS. We examine both types of diagrams in this section of the chapter.

Process Maps

A *business process* is a natural group of business activities that create value for an organization. **Process maps** document business processes in easy-to-follow diagrams. Did you understand the logic diagram in Figure 3-1 at the beginning of the chapter? It's an example of a process map. Studies suggest that process maps are among the easiest to draw and are also among the easiest for novices to follow.

In businesses, a major process is usually the sales or order fulfillment process. A process map for this process (Figure 3-8) shows such activities as customers placing orders, warehouse personnel picking goods, and clerks shipping goods. Managers can create similar maps that show just about any other process—for example, depict how an organization processes time cards for a payroll application, how a business responds to customer returns, or how a manager deals with defective merchandise.

Case-in-Point 3.5 Increased competition and tighter profit margins have forced companies to look for places where they might be able to save money. One large accounting firm has used process mapping software to assist clients in evaluating and redesigning their business processes. For example, the firm's business reengineering practice helped a financial services company cut its costs and become more efficient. The company was able to cut in half the time it took to approve a loan—and it needed 40% fewer staff to do it.

Internal and external auditors can use process maps to help them learn how a department or division operates, assist them in documenting what they have learned, and help them identify internal control weaknesses or problems in existing operations. An additional benefit is to use such maps as training aids. Consultants frequently use process maps to help them study business processes and redesign them for greater productivity. Accountants and managers can also use this tool to help them describe current processes to others.

Like most other types of documentation, you can draw process maps in multi-level versions called *hierarchical process maps* that show successively finer levels of detail. Such maps are especially popular on the web because viewers can click on individual symbols to
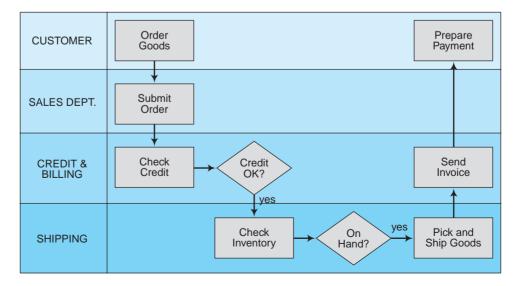

FIGURE 3-8 A process map for the order fulfillment process (created with Microsoft Word).
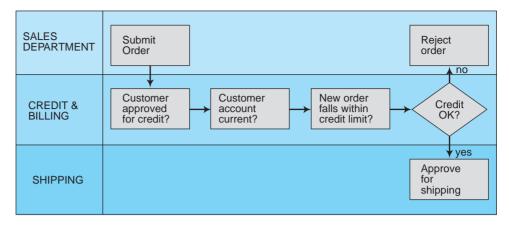
FIGURE 3-9   A second-level process map for the credit approval process of Figure 3-8.

see more information for any given process or decision. Figure 3-9, for example, illustrates a secondary-level process map for checking credit that might link to the ''Check Credit'' box in Figure 3-8.

Guidelines for Drawing Process Maps

Process maps vary considerably and the symbols found in web versions are remarkably inconsistent. Nonetheless, it is possible to use the symbols that you already know from drawing flowcharts to create process maps, including: a rectangle (to represent a process), a diamond (to represent decisions), an oval (to depict the starting and ending points for a process), an off-page connector, and a document symbol. Creating a good process map requires a blend of art, science, and craftsmanship, all of which mostly comes with practice. Here are some guidelines to use when drawing process maps.[4]

1. Identify and define the process of interest. The goal is to stay focused on the scope of the process you are trying to map.
2. Understand the purpose for the process map. Is it to identify bottlenecks? To discover redundancies?
3. Meet with employees to get their ideas, suggestions, and comments. Don't hesitate to ask challenging or probing questions.
4. Remember that processes have inputs, outputs, and enablers. An input could be an invoice; an output could be a payment check for a supplier, and an enabler helps a process achieve results. In AISs, information technology is itself a common enabler.
5. Show key decision points. A process map will not be an effective analytical tool without decision points (the intellectual or mental steps in a process).
6. Pay attention to the level of detail you capture. Did you capture enough detail to truly represent the process and explain it to others?
7. Avoid mapping the ''should-be'' or ''could-be''. Map what is.
8. Practice, practice, practice.

[4]Joe Paradiso, ''The Essential Process: The Quick and Painless Path to Successful Process Mapping,'' *Industrial Engineer*, (April 2003), Vol. 35, p. 46–49.
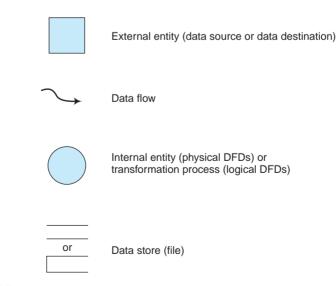
Data Flow Diagrams

Data Flow Diagrams (DFDs) are used primarily in the systems development process—for example, as a tool for analyzing an existing system or as a planning aid for creating a new system. Because documented data flows are important for understanding an AIS, many of the remaining chapters of this book use DFDs to illustrate the flow of data in the AISs under discussion.

Data Flow Diagram Symbols. Figure 3-10 illustrates the four basic symbols used in DFDs. A rectangle or square represents an external data source or data destination—for example, a customer. To show this, a DFD would include the word ''customer'' inside a data source or destination symbol. In Figure 3-10, the term ''external entity'' means ''an entity outside the system under study,'' not necessarily an entity that is external to the company. Thus, for example, a ''customer'' might be another division of the same company under study.

Data flow lines are lines with arrows that indicate the direction of data flow. Thus, data flow lines indicate the paths that data follow into, out of, or through the system under study. For this reason, every data source symbol will have one or more data flow lines leading away from it, and every data destination symbol will have one or more data flow lines leading into it. For clarity, you should label each data flow line to indicate exactly what data are flowing along it.

A circle or ''bubble'' in a DFD indicates a system entity or process that changes or transforms data. (Some authors prefer to use squares with rounded corners for this symbol.) In physical DFDs (discussed shortly), the label inside a bubble typically contains the title of the person performing a task—for example, ''cashier.'' In logical DFDs (also discussed shortly), the label inside the bubble describes a transformation process—for example, ''process cash receipts.''

Finally, DFDs use a set of parallel lines or an open rectangle to represent a store or repository of data. This is usually a file of some sort. If data are permanently stored, a data store symbol is mandatory. If data are collected over time and stored in some temporary

External entity (data source or data destination)

Data flow

Internal entity (physical DFDs) or transformation process (logical DFDs)

or Data store (file)

FIGURE 3-10 Symbols for data flow diagrams.

place, you are not required to use a file symbol for this (although experts recommend including one for clarity).

Context Diagrams.   As with system flowcharts and process maps, we typically draw DFDs in levels that show increasing amounts of detail. Designers first prepare a high-level DFD called a **context diagram** to provide an overview of a system. Figure 3-11 is an example of a context diagram for the payroll processing of Figure 3-6.

The DFD in Figure 3-11 shows the inputs and outputs of the application (payroll processing) as well as the data sources and destinations external to the application. Thus, this context diagram uses rectangles to identify ''Timekeeping'' and ''Human Resources'' as external entities, despite the fact that these departments are internal to the company. This is because these entities are external to the payroll processing system under study. The data flow lines connecting these entities to and from the system (e.g., time card data) are called system interfaces.

Physical Data Flow Diagrams.   A context diagram shows very little detail. For this reason, system designers usually elaborate on the elements in context DFDs by **decomposing** them into successively more detailed levels. These subsequent DFDs show more particulars, such as the detailed processes of the application, and the inputs and outputs associated with each processing step.

The first level of detail is commonly called a **physical data flow diagram**. Figure 3-12 is an example for our payroll illustration. A physical DFD closely resembles the document flowcharts discussed earlier in this chapter; that is, it focuses on physical entities such as the employees involved in the system under study, as well as the tangible documents, reports, and similar hard-copy inputs and outputs that flow through the system. Thus, for example, the bubbles in the physical DFD of Figure 3-12 identify the data-entry clerk who enters payroll information into the computer, the payroll cashier who distributes paychecks to employees, and the tax accountant who sends tax information to the Internal Revenue Service of the federal government.

Figure 3-12 illustrates several important characteristics of physical DFDs. First, we observe that each bubble contains a number as well as a title. Including a number in each bubble makes it easier to reference it later. This also assists designers in the decomposition tasks discussed shortly. Second, we notice that a physical DFD includes the same inputs and outputs as its predecessor context diagram in Figure 3-11—i.e., the context DFD and the physical DFD are balanced. This *balancing* is important because unbalanced DFDs are
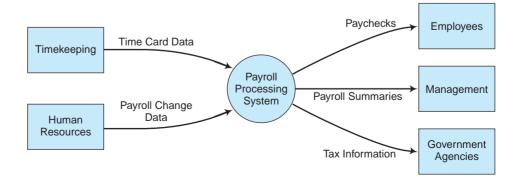

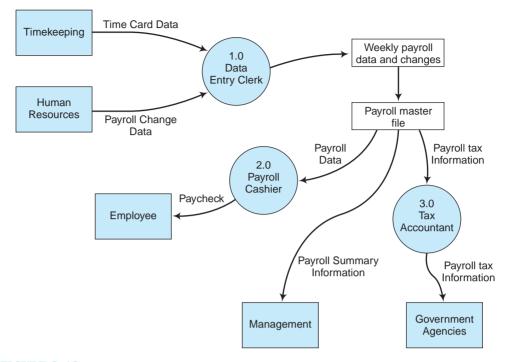**FIGURE 3-11**   A context diagram for a payroll processing system.

FIGURE 3-12 A physical data flow diagram.

inconsistent and therefore probably contain errors. Third, we find that all the bubbles in the physical DFD contain the names of system entities (i.e., the titles of employees). These titles should correspond to the titles in an official organization chart.

Finally, we see that a physical DFD lists the job title of only one typical employee in an entity symbol, despite the fact that several employees may perform the same task—for example, several data-entry clerks or payroll cashiers. This last characteristic also applies when several employees perform the same task at different locations—for example, a company has several payroll cashiers who distribute paychecks at each of its manufacturing facilities. This keeps the DFD simple, more readable, and therefore more easily understood.

Logical Data Flow Diagrams. A physical DFD illustrates which internal and external entities participate in a given system but does not give the reader a good idea of what these participants do. For this task, we need one or more **logical data flow diagrams** that address this requirement.

Figure 3-13 is a logical DFD for the payroll illustration in Figure 3-12. In Figure 3-13, note that each bubble no longer contains the name of a system entity, but instead contains a verb that indicates a task the system performs. For example, instead of a single bubble with the title ''data-entry clerk,'' as in Figure 3-12, the logical DFD in Figure 3-13 shows two bubbles with the titles ''process employee hours worked'' and ''process payroll change data'' because these are separate data processing tasks such clerks perform.

From the standpoint of good system design and control, describing system processes is important because how a system performs its tasks is often more important than what tasks it performs. For example, all payroll systems prepare paychecks, but not all payroll systems do this exactly the same way. The differences may require different hardware, software,
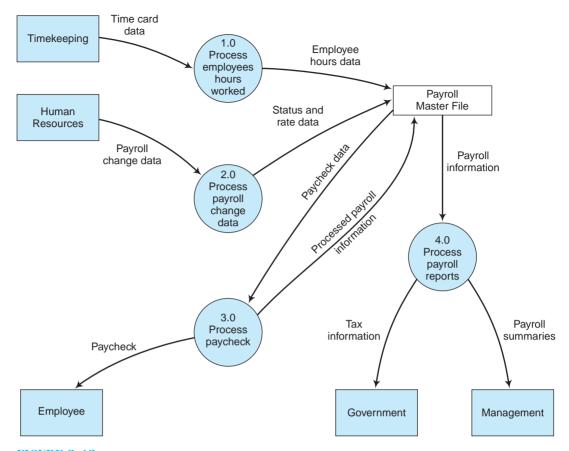
FIGURE 3-13   A logical data flow diagram for a payroll processing system.

procedures, or controls. Logical DFDs help designers decide what system resources to acquire, what activities employees must perform to run these systems, and how to protect and control these systems after they are installed.

Figure 3-13 is a **level 0 data flow diagram** because it shows only in broad terms what tasks a system performs. Most systems are more complex than this and therefore require more detail to describe them completely. The task of creating such detail is called **decomposition**, which becomes necessary because DFD designers try to limit each level diagram to between five and seven processing symbols (bubbles).

Figure 3-14 shows an example of a **level 1 data flow diagram**— an ''explosion'' of symbol 3.0 (in Figure 3-13) with caption ''process paycheck.'' Here, we see that ''processing paychecks'' entails computing gross pay, determining payroll deductions, and calculating net pay. If necessary, you can also show ancillary computer files at this level.

To fully document the system, you would continue to perform these decomposition tasks in still further DFDs. For example, you might decompose the procedure ''compute payroll deductions'' in bubble 3.2 of Figure 3-13 into several additional processes in lower-level DFDs—for example, separate DFDs for ''compute medical deductions,'' ''compute savings plan deductions,'' ''compute tax deductions,'' and so forth. In this way, a set of DFDs become linked together in a hierarchy.
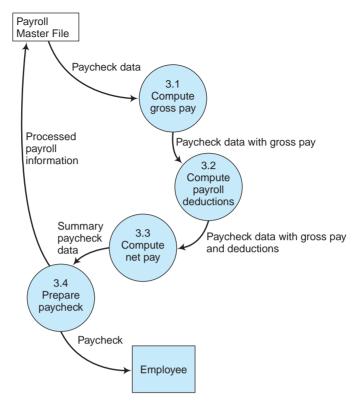
FIGURE 3-14 An exploded view of the ''process paycheck'' bubble of Figure 2-11.

Guidelines for Drawing Data Flow Diagrams

Data flow diagrams use fewer symbols than system flowcharts, and to some people are therefore easier to prepare and understand. Many companies use both types of documentation, with the choice often hinging on the preference of the designer. But like creating other types of system documentation, creating DFDs is as much art as science. The following rules can help you design them better, make them clearer, and assist you in avoiding simple errors.

1. Avoid detail in high-level DFDs (i.e., in levels 0 and 1). Where appropriate, combine activities that are performed at the same place, at the same time, or that are logically related.
2. As a general rule, each logical DFD should contain between five and seven processing bubbles. This guideline helps keep things simple, and again helps you avoid showing too much detail in high-level DFDs.
3. Different data flows should have different names. This avoids confusion about what data are flowing where.
4. Unless they are outside the system or used for archiving, all data stores should have data flows both into them and out of them. Thus, an internal file symbol that lacks both of these data flow lines is usually in error.
5. Even if a file is temporary, it is usually desirable to include it in a DFD.

6. Classify most of the final recipients of system information as external entities.

7. Classify all personnel or departments that process the data of the current system as internal entities.

8. Display only normal processing routines in high-level DFDs. Avoid showing error routines or similar exception tasks in them.

9. Where several system entities perform the same task, show only one to represent them all. This rule also applies when system personnel perform the same task at different locations of the organization—for example, at different plants.

OTHER DOCUMENTATION TOOLS

There are many other tools for documenting AISs besides document flowcharts, system flowcharts, process maps, and data flow diagrams. Two of them are (1) program flowcharts, and (2) decision tables. Because these tools are used mostly by consultants and IT professionals rather than accountants, we will describe them only briefly. Accountants should have some familiarity with these tools, however, because they may see them—for example, when reviewing the design for a revised accounting system.

Program Flowcharts

Because large computer programs today involve millions of instructions, they require careful planning and the coordinated work of hundreds of systems analysts and programmers. Typically, organizations use **structured programming** techniques to create these large programs in a hierarchical fashion—that is, from the top down. This means that the developers design the main routines first and then design subroutines for subsidiary processing as major processing tasks become clear.

To help them plan the logic for each processing routine, IT professionals often create one or more **program flowcharts** (Figure 3-15). Program flowcharts outline the processing logic for each part of a computer program and indicate the order in which processing steps take place. After designing such program flowcharts, the developer typically presents them to colleagues in a **structured walkthrough** or formal review of the logic. This process helps the reviewers assess the soundness of the logic, detect and correct design flaws, and make improvements. Upon approval, the program flowchart then becomes a ''blueprint'' for writing the instructions of the computer program itself, and of course serves to document the program as well.

Program flowcharts use many of the same symbols as system flowcharts (refer back to Figure 3-5). A few specialized symbols for program flowcharts are the diamond symbol (which indicates a decision point in the processing logic) and the oval symbol (which indicates a starting or stopping point).

Like system flowcharts and data flow diagrams, program flowcharts can be designed at different levels of detail. The highest-level program flowchart is sometimes called a **macro program flowchart** and provides an overview of the data processing logic. A lower-level program flowchart would indicate the detailed programming logic necessary to carry out a processing task. Figure 3-15 is a detailed (lower-level) program flowchart for a sales report application.
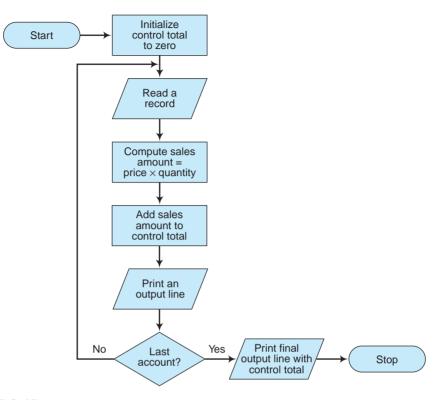
FIGURE 3-15 A program flowchart for a sales application.

Decision Tables

When a computer program involves a large number of conditions and subsequent courses of action, its program flowchart tends to be large and complex. A **decision table** (Figure 3-16) is a table of conditions and processing tasks that indicates what action to take for each possibility. Sometimes, decision tables are used as an alternative to program flowcharts. More commonly, they are used in addition to these flowcharts. To illustrate decision tables, consider the following scenario:

A credit union pays interest to its depositors at the rate of 5% per year. Accounts of less than $5 are not paid interest. Accounts of $1,000 or more that have been with the credit union for more than one year get paid the normal 5 percent, plus a bonus of. 5 percent.

Figure 3-16 illustrates a decision table to help the credit union decide how much interest to pay each account. Note that the decision table consists of four parts: (1) the condition stub outlines the potential conditions of the application, (2) the action stub outlines the available actions that can be taken, (3) the condition entries depict the possible combinations of conditions likely to occur, and (4) the action entries outline the action to be taken for each combination of conditions.

The rules at the top of the decision table set forth the combination of conditions that may occur, and the action entries show what to do for each of them. For the illustration at hand, three conditions affect the data processing of each account: (1) an account balance less than $5, (2) an account balance less than $1,000, and (3) an account one year old or less. As defined, each of these conditions can now be answered ''yes'' or ''no.'' Figure 3-15

		Rules			
		1	**2**	**3**	**4**
Condition stub	**Conditions**				
	Account balance less than $5	Y	N	N	N
	Account balance less than $1,000	*	Y	*	N
	Account 1 year old or less	*	*	Y	N
Action stub	**Actions**				
	Pay no interest	X			
	Pay 5 percent interest		X	X	
	Pay 5.5 percent interest				X

Condition entries (columns 1–4 of Conditions); Action entries (columns 1–4 of Actions)

FIGURE 3-16 This is a decision table to help a credit union decide how much interest to pay each account. An asterisk (*) means that the condition does not affect the course of action.

is a decision table for the illustration at hand, in which Y stands for ''yes'' and N stands for ''no.'' The combination of Ys and Ns in each column of the table illustrates each possible condition the system might encounter. Using Xs, the decision table also shows what course of action should be taken for each condition (i.e., how much interest should be paid to each account).

The major advantage of decision tables is that they summarize the processing tasks for a large number of conditions in a compact, easily understood format. This increases system understanding, resulting in fewer omissions of important processing possibilities. Decision tables also serve as useful documentation aids when new data processing conditions arise or when changes in organizational policy result in new actions for existing conditions. This advantage is particularly important to AISs because of organizational concern for accuracy and completeness in processing financial data.

One drawback of decision tables is that they do not show the order in which a program tests data conditions or takes processing actions, as do program flowcharts. This is a major deficiency because the order in which accounting data are tested or processed is often as important as the tests or processing themselves. A second drawback is that decision tables require an understanding of documentation techniques beyond flowcharting. Finally, decision tables require extra work to prepare, and this work may not be cost effective if program flowcharts must be prepared anyway.

Software Tools for Graphical Documentation and SOX Compliance

Accountants, consultants, and system developers can use a variety of software tools to create **graphical documentation** of existing or proposed AISs. The simplest tools include presentation software, such as Microsoft PowerPoint, as well as word processing and spreadsheet software such as Microsoft Word and Excel. The advantages of using such tools closely parallel those of using word processing software instead of typewriters (e.g., easily revised documents, advanced formatting capabilities and coloring options, and a variety of reproduction capabilities). For example, the authors used Microsoft Word to create the process maps in Figures 3-8 and 3-9.

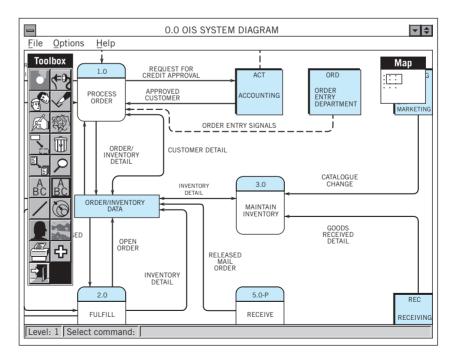**Microsoft Word, Excel, and PowerPoint.** Using the ''AutoShapes'' option in the Drawing Toolbar of Microsoft Word, Excel or PowerPoint, you can reproduce most of the graphics symbols and logic diagrams in this chapter. (The connectors in Excel are different from, as well as better than, simple lines because they adjust automatically when you reposition symbols in your charts.) Two additional advantages of using Excel to create

graphical documentation are the ability to create large drawings (that exceed the margins of word-processing documents) and the option to embed computed values in flowcharting symbols. Problem 3–21 at the end of the chapter describes how to use Excel to create such graphical documentation.

CASE Tools. The capabilities of specialized graphical documentation software exceed those of word-processing or spreadsheet packages. These **CASE tools** (an acronym for computer-assisted software engineering) automate such documentation tasks as drawing or modifying flowcharts, drawing graphics and screen designs, developing reports, and even generating code from documentation. Thus, CASE tools are to flowcharts what word processors are to text documents. Figure 3-17 is an example of a CASE package in use, drawing a data flow diagram.

Most CASE products run on personal computers. Examples include *iGrafx* (Micrografx, Inc.), *allCLEAR* (SPSS, Inc.), *SmartDraw* (SmartDraw Software, Inc.), and *Visio* (Microsoft Corp.). These products are especially popular with auditors and consultants, who use them to document AISs using the techniques discussed above, as well as to analyze the results. Graphical documentation software enables its users to create a wide array of outputs, including data flow diagrams, entity-relationship diagrams (described in Chapter 7), system flowcharts, program flowcharts, process maps, and even computer network designs.

More complex CASE products enable their users to do even more. Examples include *Application Factory* (Cortex Corporation), *Excelerator II* (Intersolv), and *Pacbase* (CGI Systems, Inc.). These CASE tools enable system designers to create process models, data-entry screens, report formats, menu screens, structure charts, and customized user interfaces. Most CASE packages also include modules for creating data dictionaries and word

FIGURE 3-17 This CASE tool is a software program called Excelerator™, which is used here to create a data flow diagram. The toolbox on the left contains symbols that the user can select for his or her diagram.

processors for creating written documentation. Top-end packages include project management modules, and also support client/server applications, encourage object-oriented programming, and provide visual tools for workflow analyses and process redesigns.

Front-end CASE tools focus on the early (''front end'') tasks of systems design—for example, requirements-design activities. *Backend CASE tools* automate the detailed design tasks required in the later stages of a project—for example, developing detailed program flowcharts. Integrated CASE (I-CASE) packages enable users to perform both types of tasks and perhaps even generate computer code directly from logic diagrams. As a result, these tools help support **rapid application development (RAD)** and help organizations save money.

Graphical documentation software tools enable their users to generate documentation quickly and consistently, as well as to automate modifications to this documentation later as changes are required. They include templates and models that allow users to document almost any business and system environment. But these packages only create what they are told to create. Like word processors, they lack imagination and creativity, and they also require training to use them efficiently.

SOX Compliance.   Many businesses now use specialized software packages to automate the tasks required by Section 404 of the Sarbanes-Oxley Act of 2002. Just as word processing software makes document revisions easier, these ''compliance software packages'' enable businesses to reduce the time and costs required to satisfy SOX requirements. *BlindView* (Symatec Corporation), for example, monitors servers to determine which applications each employee can access, identifies financial information stored on these servers, determines whether adequate segregation of duties are in place, and even restricts employee access to selected information as dictated by SOX. Similarly, *Movaris Certainty* (Movaris, Inc.) is another software tool that companies can use for compliance purposes:

Case-in-Point 3.6   The Chicago Mercantile Exchange creates markets in which customers buy and sell futures contracts for such products as corn, wheat, soybeans, and pork bellies—and uses Movaris Certainty 8.0 to help it meet SOX requirements. The software enables the exchange to centralize all its compliance documentation in one Oracle database, document, report, and review its internal controls, and analyze workflow capabilities.[5]

Some software focuses on risk identification and management. *Certus Governance Suite* (Certus Software, Inc.) is one example. This suite includes SOX compliance and risk software that enables organizations to identify the risks tied to various accounting applications, and to map these risks back to the controls that govern them. The software also enables businesses to identify which divisions and controls involve multiple financial systems or operations, and to consolidate such reviews in a single application.

OpenPages FCM (OpenPages, Inc.) and *BizRights* (Approva Corp.) provide somewhat similar capabilities. *OpenPages FCM* includes a compliance database, workflow management tools, and provides a software dashboard that enables executives to verify that specific managerial controls are now in place as well as to identify control deficiencies that might affect financial reports. *BizRights* software enables users to focus on separation of duties in an ERP environment. Using a rule book, the software determines whether such separations are appropriate, and therefore enables companies to reduce their reliance on external auditors for such work.

---

[5]Hoffman, Thomas, ''Calibrating Toward Compliance'' *Computerworld* Vol. 40, No. 6 (February 6, 2006), pp. 21–24.

END-USER COMPUTING AND DOCUMENTATION

End-user computing refers to the ability of non-IT employees to create computer applications of their own. Today, we take much of this ''computing'' for granted—for example, when employees manipulate data with word processing, spreadsheet, database management systems, or tax packages—because all of these programs were developed expressly so that end users can develop applications for themselves.

The Importance of End-User Documentation

End-user applications often require substantial resources or perform mission-critical functions for busy organizations. Thus, end users should document their applications for many of the same reasons that professionals must document their applications. One rationale for this is that end users require complete, easy-to-follow training manuals, tutorials, and reference guides to help them use computer software and perform application tasks. New software always seems to place us at the ''low end of the learning curve'' (i.e., in unfamiliar territory), thus making documentation important for learning how to accomplish things or undo mistakes.

Documentation is also important when end users develop their own applications (for example, spreadsheet models or database applications). This self-development places the responsibility for documenting these applications on the same employees who created them. Unfortunately, this documentation task is often overlooked or is performed so poorly that it might as well be overlooked. Such oversight can be costly. For example, time is wasted when other employees must alter the system but lack the basic documentation to accomplish this task. Thus, even if the developer is the only one in the office who uses a particular application, managers should insist that he or she document it—for example, in case of sickness or dismissal.

The specific items that should be used to document any particular end-user application will, of course, vary with the application. For example, businesses often find it convenient to use systematic file names to identify word processing documents and to embed these file names within the reports to help others find them later. Figure 3-18 provides some ideas for documenting spreadsheet applications.

1. Name of the developer.
2. Name of the file where the application is stored.
3. Name of the directories and subdirectories where the application is stored.
4. Date the application was first developed.
5. Date the application was last modified, and the name of the person who modified it.
6. Date the application was last run.
7. Name and phone number of person to call in case of problems.
8. Sources of external data used by the system.
9. Important assumptions made in the application.
10. Important parameters that must be modified in order to change assumptions or answer ''what-if'' questions.
11. Range names used in the application and their locations in the spreadsheet.

FIGURE 3-18   Examples of information to include when documenting spreadsheets.

Policies for End-User Computing and Documentation

Besides finding that some applications are poorly documented, organizations sometimes also discover that the end-user applications of one department duplicate those of another. Then, too, a lack of corporate-wide documentation standards can penalize both the developer and the organization in the long run. Finally, many firms find that end-user applications are not well tested and that internal controls are either weak or non-existent. To avoid such problems, businesses should establish and follow the guidelines outlined here to control end-user applications development:

1. *Formally evaluate large projects.* Employees should be allowed to create a large application only after it has withstood the scrutiny of a formal review of its costs and benefits. When projects are large, higher-level management should be involved in the go-ahead decision.

2. *Adopt formal end-user development policies.* Employees usually do not develop poor applications because they wish to do so, but because no organizational policies exist that restrict them from doing so. Policy guidelines should include procedures for testing software, examining internal controls, and periodically auditing systems.

3. *Formalize documentation standards.* At this point in the chapter, the importance of formal documentation should be self-evident. What may be less obvious is the need to create procedures for ensuring that these documentation standards are met.

4. *Limit the number of employees authorized to create end-user applications.* This restricts applications development to those employees in whom management has confidence, or perhaps who have taken formal development classes.

5. *Audit new and existing systems.* The more critical an end-user system is to the functioning of a department or division, the more important it is for organizations to require formal audits of such systems for compliance with the guidelines outlined previously.

Case-in-Point 3.7   In a parallel universe, the apprentice had yet one more question for the master accountant. "Master," he said, "I now understand why documenting end-user applications is so important. Is it not correct, therefore, that all companies require such documentation?"

The master accountant shifted his weight and cleared his throat. "The lesson is over for today," he said.

AIS AT WORK
Flowcharts Help Recover Embezzled Assets

In August 2001, a forensic team of two CPAs and an IT professional were hired by a law firm to help investigate the suspected siphoning of large sums of money from government offices of a Persian Gulf sovereignty. The team only had about seven weeks to collect as much relevant information as possible, analyze this information, and then make it easy to understand (in the event that the case went to court).

The assets were allegedly hidden in investment banks and brokerage firms in a number of European countries. Therefore, the biggest challenges were to determine what information should be collected and then what information was still available—the concern

was that this embezzlement had been going on for more than a decade. The search for transactions and data required extensive information technology and data-mining skills, because the money trail was hidden in databases at financial institutions. The team quickly realized the value of understanding databases, particularly Microsoft Access database software.

But tracing the cash flows became a difficult task because the number of transactions was large and the flow of monies both inside and outside the country was complicated. The forensic team quickly decided that the most efficient and effective method for keeping track of these cash flows (and how they fit into the whole picture) was to use flow charts and decision tables. Understanding how the money was moving inside and outside the organization provided the basis for the team to identify the transactions where assets were misappropriated. As a result, the team could also identify existing controls that could be thwarted, and suggest how the investigation should proceed. In the end, the team found hundreds of suspect transactions that led to an out-of-court settlement.

Source: Marc Siegel, ''Recovery of Embezzled Assets Half a World Away,'' *Journal of Accountancy*, (August 2001), Vol. 192, No. 2, p. 45–49.

SUMMARY

- Nine reasons to document an AIS are: (1) to explain how the system works, (2) to train others, (3) to help developers design new systems, (4) to control system development and maintenance costs, (5) to standardize communications among system designers, (6) to provide information to auditors, (7) to document a business's processes, (8) to help a company comply with the Sarbanes-Oxley Act of 2002, and (9) to establish employee accountability for specific tasks or procedures.

- A document flowchart describes the physical flow of documents through an AIS—for example, by providing an overview of where documents are created, sent, reviewed, and stored, and what activities they trigger.

- A system flowchart describes the electronic flow of data through an AIS, indicates what processing steps and files are used and when, and provides an overview of the entire system.

- Process maps also describe the flow of information through an organization, use only a few symbols, and (to many) are among the easiest to draw and understand.

- Data flow diagrams provide both a physical and a logical view of a system, but concentrate more on the flow and transformation of data than on the physical devices or timing of inputs, processing, or outputs.

- Two additional documentation tools are program flowcharts and decision tables. Accountants do not need to be programmers to evaluate or design an accounting information system, but they should understand in general terms how these tools work.

- A variety of software tools exist for documenting AISs. These include standard personal productivity tools such as word processing and spreadsheet software, specialized CASE tools, and software packages designed to help companies comply with SOX.

- End-user computing is important because many employees do it and also because such applications often contribute significantly to the efficiency of specific departments or divisions. But many employees do not document their applications very well, and this often costs time and money.

KEY TERMS YOU SHOULD KNOW

CASE (computer-assisted software engineering) tools

context diagram

data flow diagrams (DFDs)

decision table

decomposition

document flowchart

documentation

end-user computing

graphical documentation software

job stream

level 0 data flow diagram

level 1 data flow diagram

logical data flow diagram

macro program flowchart

object-oriented software

physical data flow diagram

process map

program flowcharts

rapid application development (RAD)

sandwich rule (system flowcharts)

signed checklists

structured programming

structured walkthrough

system flowchart

TEST YOURSELF

The first three questions refer to this diagram:

Q3-1. The diagram here is most likely a:

a. Document flowchart

b. System flowchart

c. Data flow diagram

d. Program flowchart

Q3-2. In the diagram here, the symbol with the letter A represents:

a. An on-page connector

b. An off-page connector

c. A file

d. An answering machine

Q3-3. In this diagram, the arrow represents:

a. A wireless transmission

b. A telephone call

c. An information flow

d. A management order to a subordinate

Q3-4. Document flowcharts would not be able to represent:

a. The flow of information when ordering office supplies

b. The flow of information when hiring new employees

c. The flow of information when creating orders for new magazine subscriptions

d. The logic in performing payroll processing

Q3-5. Which of the following is *not* true about system flowcharts?

    a. They can depict the flow of information in computerized AISs

    b. They use standardized symbols

    c. They cannot show how documents flow in an AIS

    d. They often document an audit trail

Q3-6. Which of the following is *not* true about process maps?

    a. They depict the flow of information in computerized AISs

    b. They use standardized symbols

    c. Government agencies as well as businesses often use them

    d. Web pages often depict hierarchical versions of them

Q3-7. The *sandwich rule* states that:

    a. You should only create logic diagrams that have some "meat" in them

    b. Every diagram should have a cover page and a summary page

    c. A processing symbol should be between an input and an output symbol

    d. In DFDs, there should always be data flow lines leading to and from files

Q3-8. Which of these is *not* a good guideline to follow when creating DFDs?

    a. Avoid detail in high-level DFDs

    b. Avoid drawing temporary files in DFDs

    c. Classify most of the final recipients of system outputs as external entities

    d. Avoid showing error routines or similar exception tasks

Q3-9. A meeting in which computer programmers outline their logic to others is called a:

    a. Decision meeting

    b. RAD meeting

    c. Mythical event

    d. Structured walkthrough

Q3-10. A decision table shows:

    a. The possible conditions and processing alternatives for a given situation

    b. Who sat where at a board meeting

    c. The rules for drawing DFDs

    d. The local outsourcing vendors in the area for documentation tasks

DISCUSSION QUESTIONS

3-1. Why is documentation important to accounting information systems? Why should accountants be interested in AIS documentation?

3-2. Distinguish between document flowcharts, system flowcharts, data flow diagrams, and program flowcharts. How are they similar? How are they different?

3-3. What are document flowcharts? How does a document flowchart assist each of the following individuals: (1) a systems analyst, (2) a systems designer, (3) a computer programmer, (4) an auditor, and (5) a data security expert?

3-4. Although flowcharting is an art rather than a science, some guidelines can be used to make better flowcharts. What are these guidelines for document, system, and data flow diagram flowcharts?

3-5. What are the four symbols used in data flow diagrams? What does each mean?

3-6. Why are data flow diagrams developed in a hierarchy? What are the names of some levels in the hierarchy?

3-7. Look at the process map in Figure 3-13. Trace the steps in the order fulfillment process. Do you think this figure is more helpful than a narrative would be in understanding the flow of events in the process?

3-8. What is the purpose of a decision table? How might decision tables be useful to accountants?

3-9. What are CASE tools? How are they used? How do CASE tools create documentation for AISs? If you were a systems analyst, would you use a CASE tool?

3-10. What is end-user computing? Why is documentation important to end-user computing? What guidelines should companies develop to control end-user computing?

PROBLEMS

3-11. To view the Drawing ToolBar of Microsoft Excel, select the following options from the main menu: View\Toolbars\Drawing. (You can also click directly on the Drawing icon in the Standard Toolbar.) After the Drawing Toolbar appears, select Autoshapes\Flowchart and observe the symbols that appear in the selection list. There should be approximately 28 of them (using Office 2000). If you allow your mouse to hover over a specific symbol, its title and meaning will appear in a tool-tip box. Finally, if you click on a specific symbol, your mouse icon will change to a cross-hair and you will be able to draw this symbol on your spreadsheet. Create a list with items similar to the one below that contains all the symbols in your version of Excel.

Predefined Process

3-12. Draw a document flowchart to depict each of the following situations.

a. An individual from the marketing department of a wholesale company prepares five copies of a sales invoice, and each copy is sent to a different department.

b. The individual invoices from credit sales must temporarily be stored until they can be matched against customer payments at a later date.

c. A batch control tape is prepared along with a set of transactions to ensure completeness of the data.

d. The source document data found on employee application forms are used as input to create new employee records on a computer master file.

e. Delinquent credit customers are sent as many as four different inquiry letters before their accounts are turned over to a collection agency.

f. Physical goods are shipped back to the supplier if they are found to be damaged upon arrival at the receiving warehouse.

g. The data found on employee time cards are keyed onto a hard disk before they are processed by a computer.

h. The data found on employee time cards are first keyed onto a floppy diskette before they are entered into a computer job stream for processing.

i. A document flowchart is becoming difficult to understand because too many lines cross one another. (Describe a solution.)

j. Three people, all in different departments, look at the same document before it is eventually filed in a fourth department.

k. Certain data from a source document are copied into a ledger before the document itself is filed in another department.

3-13. Develop a document flowchart for the following information flow. The individual stores in the Mark Goodwin convenience chain prepare two copies of a goods requisition form (GRF) when they need to order merchandise from the central warehouse. After these forms are completed, one copy is filed in the store's records and the other copy is sent to the central warehouse. The warehouse staff gets the order and files its copy of the GRF form in its records. When the warehouse needs to restock an item, three copies of a purchase order form (POF) are filled out. One copy is stored in the warehouse files, one copy goes to the vendor, and the third copy goes to the accounts payable department.

3-14. The Garcia-Lanoue Company produces industrial goods. The company receives purchase orders from its customers and ships goods accordingly. Assuming that the following conditions apply, develop a document flowchart for this company:

a. The company receives two copies of every purchase order from its customers.

b. Upon receipt of the purchase orders, the company ships the goods ordered. One copy of the purchase order is returned to the customer with the order, and the other copy goes into the company's purchase order file.

c. The company prepares three copies of a shipping bill. One copy stays in the company's shipping file, and the other two are sent to the customer.

3-15. The data-entry department of the Ron Mitchell Manufacturing Company is responsible for converting all of the company's shipping and receiving information to computer records. Because accuracy in this conversion is essential, the firm employs a strict verification process. Prepare a document flowchart for the following information flow:

a. The shipping department sends a copy of all shipping orders to the data-entry department.

b. A data-entry operator keys the information from a shipping order onto a diskette.

c. A supervisor checks every record with the original shipping order. If no errors are detected, the diskette is sent to the computer operations staff and the original shipping order is filed.

3-16. Amanda M is a regional manufacturer and wholesaler of high-quality chocolate candies. The company's sales and collection process is as follows. Amanda M makes use of an enterprise-wide information system with electronic data interchange (EDI) capability. No paper documents are exchanged in the sales and collection process. The company receives sales orders from customers electronically. Upon receipt of a sales order, shipping department personnel prepare goods for shipment, and input shipping data into the information system. The system sends an electronic shipping notice and invoice to the customer at the time of shipment. Terms are net 30. When payment is due, the customer makes an electronic funds transfer for the amount owed. The customer's information system sends remittance (payment) data to Amanda M. Amanda M's information system updates accounts receivable information at that time.

Draw a context diagram and a level 0 logical data flow diagram for Amanda M's sales and collection process.

3-17. The order-writing department at the Winston Beauchamp Company is managed by Alan Most. The department keeps two types of computer files: (1) a customer file of authorized credit customers and (2) a product file of items currently sold by the company. Both of these files are direct-access files stored on magnetic disks. Customer orders are handwritten on order forms with the Winston Beauchamp name at the top of the form, and item lines for quantity, item number, and total amount desired for each product ordered by the customer.

When customer orders are received, Alan Most directs someone to input the information at one of the department's computer terminals. After the information has been input, the computer program immediately adds the information to a computerized ''order'' file and prepares five copies of the customer order. The first copy is sent back to Alan's department;

the others are sent elsewhere. Design a system flowchart that documents the accounting data processing described here. Also, draw a data flow diagram showing a logical view of the system.

3-18. The LeVitre and Swezey Credit Union maintains separate bank accounts for each of its 20,000 customers. Three major files are the customer master file, the transaction file of deposits and withdrawal information, and a monthly statement file that shows a customer's transaction history for the previous month. The following lists the bank's most important activities during a representative month:

  a. Customers make deposits and withdrawals.

  b. Employers make automatic deposits on behalf of selected employees.

  c. The bank updates its master file daily using the transaction file.

  d. The bank creates monthly statements for its customers, using both the customer master file and the transactions file.

  e. Bank personnel answer customer questions concerning their deposits, withdrawals, or account balances.

  f. The bank issues checks to pay its rent, utility bills, payroll, and phone bills.

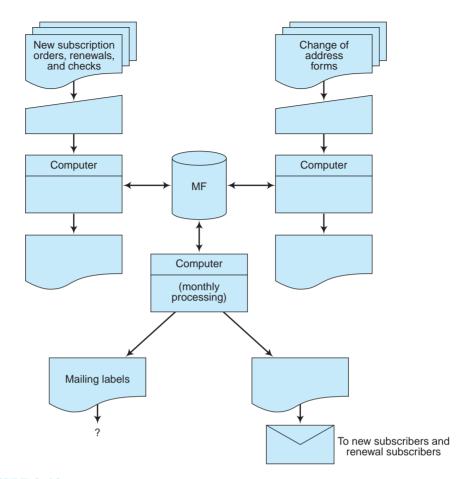 Draw a data flow diagram that graphically describes these activities.

3-19. The Jeffrey Getelman Publishing Company maintains an online database of subscriber records, which it uses for preparing magazine labels, billing renewals, and so forth. New subscription orders and subscription renewals are keyed into a computer file from terminals. The entry data are checked for accuracy and written on a master file. A similar process is performed for change-of-address requests. Processing summaries from both runs provide listings of master file changes.

  Once a month, just prior to mailing, the company prepares mailing labels for its production department to affix to magazines. At the same time, notices to new and renewal subscribers are prepared. These notices acknowledge receipt of payment and are mailed to the subscribers. The company systems analyst, Bob McQuivey, prepared the system flowchart in Figure 3-19 shortly before he left the company. As you can see, the flowchart is incomplete. Finish the flowchart by labeling each flowcharting symbol. Don't forget to label the processing runs marked computer.

3-20. The Bridget Joyce Company is an office products distributor that must decide what to do with delinquent credit-sales accounts. Mr. Bob Smith, the credit manager, divides accounts into the following categories: (1) accounts not past due, (2) accounts 30 days or less past due, (3) accounts 31 to 60 days past due, (4) accounts 61 to 90 days past due, and (5) accounts more than 90 days past due. For simplicity, assume that all transactions for each account fall neatly into the same category.

  Mr. Smith decides what to do about these customer accounts based on the history of the account in general and also the activity during the account's delinquency period. Sometimes, for example, the customer will not communicate at all. At other times, however, the customer will either write to state that a check is forthcoming or make a partial payment. Mr. Smith tends to be most understanding of customers who make partial payments because he considers such payments acts of good faith. Mr. Smith is less understanding of those customers who only promise to pay or who simply ignore follow-up bills from the company.
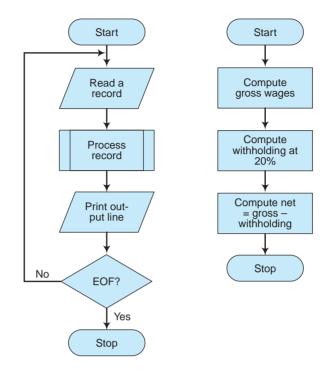
  Mr. Smith has four potential actions to take in cases of credit delinquency. First, he can simply wait (i.e., do nothing). Second, he can send an initial letter to the customer, inquiring about the problem in bill payment and requesting written notification of a payment schedule if payment has not already been made. Third, he can send a follow-up letter indicating that a collection agency will be given the account if immediate payment is not forthcoming. Fourth, he can turn the account over to a collection agency. Of course, Mr. Smith prefers to use one of the first three actions rather than turn the account over to a collection agency, because his company only receives half of any future payments when the collection agency becomes involved.

FIGURE 3-19 System flowchart for processing the subscription orders and changes for the Jeffrey Getelman Company.

    a. Create a decision table for the Bridget Joyce Company and provide a set of reasonable decision rules for Mr. Smith to follow. For now, ignore the influence of a customer's credit history.

    b. Expand the decision table analysis you have prepared in question "a" to include the credit history of the customer accounts. You are free to make any assumptions you wish about how this history might be evaluated by Mr. Smith.

3-21. Follow the directions in Exercise 3-11 to access Excel's drawing tools and then recreate the two program flowcharts shown in Figure 3-20. Draw each flowchart on a separate work sheet. Rename the first sheet "Main" and the second sheet "Sub." To embed text inside a symbol, right-click on that symbol with your mouse and then choose "Add Text" from the dropdown menu that appears. To center text inside a symbol, highlight the text and then click on the centering icon in the main toolbar.

    Create the words "Yes" and "No" that appear in this flowchart using Text Box symbols from the drawing toolbar. To eliminate the black (default) borders around these words, right-click on a Text Box *border* (*not* the text inside the symbol). When the context menu appears, select the "Colors and Lines" tab, and then "No Line" from the "Line\Color" section of that tab. Finally, you can fine-tune the position of any object by clicking on its border and then using the Ctrl key plus an arrow key to reposition it as desired.

FIGURE 3-20 Draw the flowchart on the left on one Excel sheet and the flowchart on the right on a second sheet.

After drawing these two flowcharts, we want to link them together. In this case, we want the user to click on the ''Process Record'' symbol in the main flowchart and then be able to view the second spreadsheet you've created on the alternate sheet. To create this link, click on the border of the ''Process Record'' symbol, and then select Insert\Hyperlink from the main menu. Again, be sure to click on a symbol border—not the text inside the border.

When the Insert Hyperlink dialog box appears, first select ''Place in this Document'' from the choices on the left side of the box, and then click on the name of the sheet in which you've drawn the second flowchart (''Sub'') in the lower box on the right. If you wish, you can also select a particular cell for linking in the top box—a handy feature if you've drawn your flowchart in a lower portion of the Sub sheet. That's it! Now, when you move your mouse over the ''Process Record'' symbol in the ''Main'' sheet, your mouse icon should turn into a hand, indicating that clicking on this symbol links you to the supporting document's sheet.

Using Excel software and the skills described above, recreate the documents from the list below or the ones required by your instructor:

a. The document flowchart in Figure 3-3.

b. The system flowchart in Figure 3-6.

c. The process map in Figure 3-8.

d. The context diagram in Figure 3-11.

e. The physical DFD in Figure 3-12.

 f. The logical data flow diagram shown in Figure 3-13.

g. Link the DFD in part ''e'' to a new DFD similar to Figure 3-14.

h. The program flowchart shown in Figure 3-15.

CASE ANALYSES

3-22. The Berridge Company (Drawing Document Flowcharts)

The Berridge Company is a discount tire dealer that operates 25 retail stores in a metropolitan area. The company maintains a centralized purchasing and warehousing facility and employs a perpetual inventory system. All purchases of tires and related supplies are placed through the company's central purchasing department to take advantage of the quantity discounts offered by its suppliers. The tires and supplies are received at the central warehouse and distributed to the retail stores as needed. The perpetual inventory system at the central facility maintains current inventory records, which include designated reorder points, optimum order quantities, and balance-on-hand information for each type of tire or related supply.

The participants involved in Berridge's inventory system include (1) retail stores, (2) the inventory control department, (3) the warehouse, (4) the purchasing department, (5) accounts payable, and (6) outside vendors. The inventory control department is responsible for maintenance of the perpetual inventory records for each item carried in inventory. The warehouse department maintains the physical inventory of all items carried by the company's retail stores.

All deliveries of tires and related supplies from vendors are received by receiving clerks in the warehouse department, and all distributions to retail stores are filled by shipping clerks in this department. The purchasing department places every order for items needed by the company. The accounts payable department maintains the subsidiary ledger with vendors and other creditors. All payments are processed by this department. The documents used by these various departments are as follows:

Retail Store Requisition (Form RSR). The retail stores submit this document to the central warehouse whenever tires or supplies are needed at the stores. The shipping clerks in the warehouse department fill the orders from inventory and have them delivered to the stores. Three copies of the document are prepared; two of which are sent to the warehouse, and the third copy is filed for reference.

Purchase Requisition (Form PR). An inventory control clerk in the inventory control department prepares this document when the quantity on hand for an item falls below the designated reorder point. Two copies of the document are prepared. One copy is forwarded to the purchasing department, and the other is filed.

Purchase Order (Form PO). The purchasing department prepares this document based on information found in the purchase requisition. Five copies of the purchase order are prepared. The disposition of these copies is as follows: copy 1 to vendor, copy 2 to accounts payable department, copy 3 to inventory control department, copy 4 to warehouse, and copy 5 filed for reference.

Receiving Report (Form RR). The warehouse department prepares this document when ordered items are received from vendors. A receiving clerk completes the document by indicating the vendor's name, the date the shipment is received, and the quantity of each item received. Four copies of the report are prepared. Copy 1 is sent to the accounts payable department, copy 2 to the purchasing department, and copy 3 to the inventory control department; Copy 4 is retained by the warehouse department, compared with the

purchase order form in its files, and filed together with this purchase order form for future reference.

Invoices.   Invoices received from vendors are bills for payment. The vendor prepares several copies of each invoice, but only two copies are of concern to the Berridge Company: the copy that is received by the company's accounts payable department and the copy that is retained by the vendor for reference. The accounts payable department compares the vendor invoice with its file copy of the original purchase order and its file copy of the warehouse receiving report. Based on this information, adjustments to the bill amount on the invoice are made (e.g., for damaged goods, for trade discounts, or for cash discounts), a check is prepared, and the payment is mailed to the vendor.

Requirements

1. Draw a document flowchart for the Berridge Company using the symbols in Figure 3-2.

2. Could the company eliminate one or more copies of its RSR form? Use your flowchart to explain why or why not.

3. Do you think that the company creates too many copies of its purchase orders? Why or why not?

3-23. FreezeTime, Inc. (Drawing Systems Flowcharts)

Carly Riccardi and her mother Nancy own and operate FreezeTime, Inc., a company specializing in freeze-drying flowers from clients' memorable events, such as proms and weddings. The company not only freezes the flowers, but also presents them in a variety of display packages. Each of these packages includes materials such as glass and frames that FreezeTime purchases from local suppliers. In addition to supplies for display, the company purchases office supplies and packaging materials from several vendors.

FreezeTime uses a low-end accounting software package to prepare documents and reports. As employees note a need for supplies and materials, they inform Carly or Nancy, who act as office manager and company accountant. Either Carly or Nancy enters order information into the accounting system and creates a purchase order that they fax to the supplier. Occasionally, Carly or Nancy will also call the supplier if there is something special about the product ordered. When ordered materials and supplies arrive at FreezeTime's small factory, either Carly or Nancy checks the goods received against a copy of the purchase order and enters the new inventory into the computer system.

Nancy pays bills twice each month, on the first and the fifteenth. She checks the computer system for invoices outstanding, and verifies that the goods have been received. She then enters any information needed to produce printed checks from the accounting system. FreezeTime mails checks and printed remittance advices (portions of the vendor bill to be returned) to suppliers.

Requirements

1. Create a systems flowchart for FreezeTime's purchase and payment process.

2. Comment on the value, if any, that having a systems flowchart describing this process would have to Carly or Nancy.

3-24. The Dinteman Company (Document Analysis)

The Dinteman Company is an industrial machinery and equipment manufacturer with several production departments. The company employs automated and heavy equipment in its production departments. Consequently, Dinteman has a large repair and maintenance department (R&M department) for servicing this equipment.

The operating efficiency of the R&M department has deteriorated over the past two years. For example, repair and maintenance costs seem to be climbing more rapidly than other department costs. The assistant controller has reviewed the operations of the R&M department and has concluded that the administrative procedures used since the early days of the department are outmoded due in part to the growth of the company. In the opinion of the assistant controller, the two major causes for the deterioration are an antiquated scheduling system for repair and maintenance work, and the actual cost to distribute the R&M department's costs to the production departments. The actual costs of the R&M department are allocated monthly to the production departments on the basis of the number of service calls made during each month.

The assistant controller has proposed that a formal work order system be implemented for the R&M department. With the new system, the production departments will submit a service request to the R&M department for the repairs and/or maintenance to be completed, including a suggested time for having the work done. The supervisor of the R&M department will prepare a cost estimate on the service request for the work required (labor and materials) and estimate the amount of time for completing the work on the service request. The R&M supervisor will return the request to the production department that initiated the request. Once the production department approves the work by returning a copy of the service request, the R&M supervisor will prepare a repair and maintenance work order and schedule the job. This work order provides the repair worker with the details of the work to be done and is used to record the actual repair and maintenance hours worked and the materials and supplies used.

Production departments will be charged for actual labor hours worked at a predetermined standard rate for the type of work required. The parts and supplies used will be charged to the production departments at cost. The assistant controller believes that only two documents will be required in this new system—a Repair/Maintenance Service Request initiated by the production departments and a Repair/Maintenance Work Order initiated by the R&M department.

Requirements

1. For the Repair/Maintenance Work Order document:
   a. Identify the data items of importance to the repair and maintenance department and the production department that should be incorporated into the work order.
   b. Indicate how many copies of the work order would be required and explain how each copy would be distributed.

2. Prepare a document flowchart to show how the Repair/Maintenance Service Request and the Repair/Maintenance Work Order should be coordinated and used among the departments of Dinteman Company to request and complete the repair and maintenance work, to provide the basis for charging the production departments for the cost of the completed work, and to evaluate the performance of the repair and maintenance department. Provide explanations in the flowchart as appropriate.

(CMA Adapted)

3-25. Lois Hale and Associates (Drawing Data Flow Diagrams)

Lois Hale and Associates is a medium-size manufacturer of musical equipment. The accounts payable department is located at company headquarters in Asbury Park, New Jersey, and it consists of two full-time clerks and one supervisor. They are responsible for processing and paying approximately 800 checks each month. The accounts payable process generally begins with receipt of a purchase order from the purchasing department. The purchase order is held until a receiving report and the vendor's invoice have been forwarded to accounts payable.

At that time, the purchase order, receiving report, and invoice are matched together by an accounts payable clerk, and payment and journal entry information are input to the computer. Payment dates are designated in the input, and these are based on vendor payment terms. Company policy is to take advantage of any cash discounts offered. If there are any discrepancies among the purchase order, receiving report, and invoice, they are given to the supervisor for resolution. After resolving the discrepancies, the supervisor returns the documents to the appropriate clerk for processing. Once documents are matched and payment information is input, the documents are stapled together and filed in a tickler file by payment date until checks are issued.

When checks are issued, a copy of the check is used as a voucher cover and is affixed to the supporting documentation from the tickler file. The entire voucher is then defaced to avoid duplicate payments. In addition to the check and check copy, other outputs of the computerized accounts payable system are a check register, vendor master list, accrual of open invoices, and a weekly cash requirements forecast.

Requirements

Draw a context diagram and data flow diagram similar to those in Figures 3-11 and 3-12 for the company's accounts payable process, using the symbols in Figure 3-10.

REFERENCES AND RECOMMENDED READINGS

Baraban, Regina. ''Look Out for SOX'' *Meetingsnet.com* Vol. 24, No. 2 (February 2005), pp. 23–24.

Bradford, Marianne, Sandra B. Richtermeyer, & Douglas F. Roberts. ''System Diagramming Techniques: An Analysis of Methods Used in Accounting Education and Practice,'' *Journal of Information Systems* Vol. 21, Iss.1 (Spring 2007), pp. 173–213.

Cozad, Matthew A., ''Top 5 SOX Best Practices for Small Companies'' *Strategic Finance* Vol. 87, No. 4 (October 2005), pp. 1–7.

Campbell, Gillian. ''Flowing Through a Process'' *Quality* Vol. 43, No. 6 (June 2004), pp. 50–52.

Hoffman, Thomas. ''Calibrating Toward Compliance'' *Computerworld* Vol. 40, No. 6 (February 6, 2006), pp. 21–24.

Hubbard, Larry. ''Limits to Process Documentation'' *Internal Auditor* Vol. 62, No. 3 (June 2005), pp. 26–27.

Jones, Roberta Ann, Jeffrey J. Tsay, & Kenneth Griggs. ''An Empirical Investigation of the Task Specific Relative Strengths of Selected Accounting and Information Systems Diagramming Techniques'' *Journal of Computer Information Systems* Vol. 46, No. 2 (Winter 2005–2006), pp. 99–114.

Klamm, Bonnie K., & Marcia L. Weidenmier, ''Linking Business Processes and Transaction Cycles'' *Journal of Information Systems* Vol. 18, No. 2 (Fall 2004), pp. 113–125.

Koutoupis, Andreas G., ''Documenting Internal Controls,'' *The Internal Auditor* Vol. 64, Iss.5, pp. 23–25.

Lehman, Mark W. ''Flowcharting Made Simple'' *Journal of Accountancy*, Vol. 190, No. 4 (October 2000), pp. 77–88.

Matsumoto, Isao T., John Stapleton, Jacqueline Glass, & Tony Thorpe, ''Use of Process Maps to Develop a Management Briefing Sheet for a Design Consultancy'' *Engineering Construction & Architectural Management* Vol. 12, No. 5 (2005), pp. 458–469.

Murphy, Michael. ''Demystifying Flowcharts'' *Materials Management in Health Care* Vol. 14, No. 8 (August, 2005), p. 52.

Panell, Chris. ''Teaching Computer Programming as a Language'' *Tech Directions* Vol. 62, No. 8 (March, 2003), pp. 26–27.

Paradiso, J., ''The Essential Process: The Quick and Painless Path to Successful Process Mapping,'' *Industrial Engineer*, Vol. 35, (April 2003), pp. 46–49.

Pojasek, Robert B., ''Putting Hierarchical Process Maps to Work'' *Environmental Quality Management* Vol. 15, No. 3 (Spring, 2006), pp. 73–80.

Shanker, Sheila. ''Write On'' *Journal of Accountancy* Vol. 201, No. 5 (May, 2006), p. 28.

Tuggle, Francis D., & Wendy E. Goldfinger ''A Methodology for Mining Embedded Knowledge from Process Maps'' *Human Systems Management* Vol. 23, No. 1 (2004), pp. 1–13.

Vogt, Nancy. ''Simplifying HIPAA for Staff is One Way to Ensure Good Decisions are Made about Patient Privacy'' *Journal of Health Care Compliance* Vol. 7, No. 5 (September/October, 2005), pp. 75–76.

ANSWERS TO TEST YOURSELF

1. **a**   2. **c**   3. **c**   4. **d**   5. **c**   6. **b**   7. **c**   8. **d**   9. **d**   10. **a**