

Chapter 1

The Systems Development Environment

Chapter Overview

Chapter 1 is a foundation chapter, introducing students to basic, core systems analysis and design concepts. These basic concepts include a definition of information systems analysis and design, an overview of systems analysis and design core concepts, a discussion of a system and its parts, an introduction to the modern approach to systems analysis and design, an overview of the systems analyst's role in systems development, and a review of the types of information systems and systems development. The chapter concludes with an introduction to the systems development life cycle, its four phases, and alternative approaches to development.

Instructional Objectives

Specific student learning objectives are included at the beginning of the chapter. *From an instructor's point of view, the objectives of this chapter are to:*

1. Define information systems analysis and design.
2. Define and discuss the modern approach to systems analysis and design.
3. Illustrate how systems development extends to different types of information systems and not just transaction processing systems.
4. Introduce the traditional information systems development life cycle, which serves as the basis for the organization of the material in this book.
5. Show students that the life cycle is a flexible basis for systems analysis and design and that it can support many different tools and techniques, such as prototyping and JAD.
6. Discuss the importance and role of CASE in systems development.
7. Discuss information systems development options, including IT services firms, packaged software producers, enterprise-wide solutions, open-source software, and in-house developers.
8. Discuss the different approaches to systems development, including prototyping, CASE tools, joint application development, rapid application development, participatory design, and Agile Methodologies.

Classroom Ideas

1. Emphasize the differences between methodologies, techniques, and tools. Such differences are not obvious to students; often they think of methodologies as just a set of techniques and that techniques and tools are synonymous.
2. Give concrete examples during class discussion of the following types of IS: transaction processing systems, management information systems, and decision support systems. Ask students to talk about the information systems with which they are familiar.
3. When discussing different types of information systems, discuss the differences between systems that support back room operations, such as basic accounting functions, and systems that directly affect the bottom line. If students are aware of such systems, they usually are not aware of how these systems provide competitive advantage. Discussing different types of systems provides a chance to talk about the differences between these two categories of systems.
4. When introducing the life cycle model in the text, you may want to introduce other life cycle models from other textbook authors or in other forms. This shows students that there is no one standard life cycle model and that the model they will rely on when they begin work as a systems analyst will likely differ from the life cycle model in the text. The point is to show them they can use the life cycle as an archetype to understand other models, and they should understand there is no one “correct” life cycle model. The life cycle represents activities that must be done, and the phases are a way to introduce, in an organized way, the methods, techniques, tools, and skills necessary for successful systems analysis and design.
5. Give a brief overview of the activities and outputs from each of the four life cycle phases, based on your own experience or from your reading of the rest of the book. Table 1–4 identifies the products of the four SDLC phases.
6. Although prototyping and joint application design are covered in more depth later in the book, you can provide a more in-depth introduction to these techniques than discussed in Chapter 1. Figure 1–13 is a good point of departure for a discussion of prototyping and what it adds to structured development techniques.
7. Participatory Design is briefly introduced in this chapter but can be discussed in more depth. A brief overview of the Participatory Design concept is available at http://www.cpsr.org/issues/pd/index_html. Several books are available on this topic, as well as numerous Web sites. You can also ask your students to find recent articles about this topic.
8. Spend a few minutes discussing the importance of analytical, technical, managerial, and interpersonal skills for a systems analyst. Ask students to identify which of the skills categories they think is the most important and why.
9. Figure 1–2 is an excellent tool for illustrating the basic system components.

10. Use Table 1–1 and an updated version of the same information from the most recent Software Magazine survey to begin a discussion of the many, varied sources of software in the marketplace.
11. Use Table 1–2 to summarize the alternative sources for software and how to choose among them for specific software needs. This table can serve as the basis for a discussion of the “make versus buy” decision and can be expanded to include the “not invented here” syndrome.
12. Compare and contrast Agile Methodologies with the traditional SDLC.

Lecture Notes

What Is Information Systems Analysis and Design?

All organizations have information systems, and use them for operational, tactical, and strategic advantage. For information systems to remain effective, these systems must efficiently capture, store, process, and distribute information according to business objectives. To properly maintain these systems, **systems analysts** will perform **information systems analysis and design**. Information systems analysis and design is based on an understanding of the organization’s objectives, structure, and processes, as well as the analyst’s knowledge of how to exploit information technology for competitive advantage.

Systems Analysis and Design: Core Concepts

Improving organizational information systems is the major goal of systems analysis and design. **Figure 1–1** illustrates the four systems development life cycle phases. Although a systems analyst’s primary role is **application software** development, an analysis of the other computer-based information systems components is warranted. As **Figure 1–2** shows, these components include hardware, controls, specific job roles, users of the system, system software, and documentation and training manuals. As shown in **Figure 1–3**, methodologies, techniques, and tools are central to the software engineering process.

Systems

This section introduces students to the definition of a system, its parts, and other important system concepts. The textbook defines a **system** as a group of interrelated procedures used for a business function, with an identifiable boundary, working together for some purpose. A system has nine characteristics. **Figure 1–4** shows seven of the nine characteristics. The nine characteristics are: **components, interrelated components, boundary, purpose, environment, interfaces**, input, output, and **constraints**. **Figure 1–5** describes a university as a system. A systems analyst should also be familiar with **decomposition, modularity, coupling**, and **cohesion** system concepts. **Figure 1–6** illustrates decomposition.

A Modern Approach to Systems Analysis and Design

Systems integration and enterprise-wide systems are two alternatives for company systems. The client/server model, illustrated in **Figure 1–7**, stresses the importance of systems integration, which is the focus of today's systems development efforts. Organizations may also purchase an enterprise-wide system from companies like SAP or Oracle.

The systems analyst is the person primarily responsible for performing systems analysis and design within an organization. A job advertisement for a systems analyst position is provided in **Figure 1–8**. As this job advertisement indicates, a systems analyst's skills set should include analytical, technical, managerial, and interpersonal skills.

Types of Information Systems and Systems Development

Figure 1–9 illustrates that an organization has three classes of information systems: transaction processing systems (TPS), management information systems (MIS), and decision support systems (DSS). Each of these systems has certain characteristics and suggested systems development methods. **Table 1–1** summarizes the IS characteristics and suggested systems development methods.

Developing Information Systems and the Systems Development Life Cycle

Organizations use a **systems development methodology** to develop and support their information systems. A **systems development life cycle** (SDLC) is a common methodology for systems development, and each organization will use a slightly different version. The life cycle presented in the textbook has four phases: (1) systems planning and selection; (2) systems analysis; (3) systems design; and (4) systems implementation and operation. **Figure 1–10** illustrates these phases. **Figure 1–11** illustrates the circular nature of the systems development life cycle.

Systems planning and selection has two primary activities: (1) identifying a potential project and (2) investigating the system and determining the proposed system's scope. **Systems analysis** consists of three subphases: requirements determination, requirements structuring, and alternative generation and selection. **Systems design**, the third life cycle phase, has two primary activities: logical design and physical design. **Figure 1–12** illustrates the differences between logical and physical design. The final phase, **systems implementation and operation**, involves implementing and operating the system. **Table 1–2** summarizes the products of each SDLC phase.

Approaches to Development

This chapter discusses six approaches to systems development. These approaches are: prototyping, computer-aided software engineering (CASE) tools, joint application design (JAD), rapid application development (RAD), participatory design (PD), and the use of Agile Methodologies. **Prototyping** is the process of building a scaled-down version of the desired information system. **Figure 1–13** illustrates the prototyping method. **Computer-aided Software Engineering (CASE)** tools provide automated

support for some portion of the systems development process. Computer-aided Software Engineering tools use a **repository**. **Joint Application Design (JAD)** brings users, managers, and analysts together for several days in a series of intensive meetings to specify or review system requirements. **Rapid Application Development (RAD)** radically decreases the time necessary to design and implement information systems. **Figure 1–14** compares the RAD systems development life cycle with the standard systems development life cycle. **Participatory Design (PD)**, developed in northern Europe, has as its central focus users and the improvement of their work lives. **Agile Methodologies** focus on adaptive methodologies, people instead of roles, and a self-adaptive process.

Key Terms Checkpoint Solutions

Answers for the Key Terms Checkpoint section are provided below. The number following each key term indicates its location in the key term list.

- | | |
|---|--|
| 1. systems planning and selection (28) | 16. Interface (12) |
| 2. information systems analysis and design (11) | 17. constraints (7) |
| 3. participatory design (PD) (16) | 18. systems implementation and operation (27) |
| 4. application software (2) | 19. systems development methodology (26) |
| 5. systems analyst (23) | 20. systems development life cycle (SDLC) (25) |
| 6. Joint Application Design (JAD) (14) | 21. decomposition (9) |
| 7. prototyping (17) | 22. modularity (15) |
| 8. system (21) | 23. coupling (8) |
| 9. component (5) | 24. cohesion (4) |
| 10. interrelated components (13) | 25. Computer-Aided Software Engineering (CASE) (6) |
| 11. boundary (3) | 26. repository (20) |
| 12. purpose (18) | 27. Rapid Application Development (RAD) (19) |
| 13. systems design (24) | 28. Agile Methodologies (1) |
| 14. systems analysis (22) | |
| 15. environment (10) | |

Review Questions Solutions

1. What is information systems analysis and design?

Information systems analysis and design is the process of developing and maintaining an information system.

2. What is systems thinking? How is it useful for thinking about computer-based information systems?

Systems thinking involves identifying something as a system, visualizing the system and translating it into abstract terms, and thinking about the characteristics of the specific situation. Systems thinking is useful for thinking about computer-based information systems because information systems can be seen as subsystems in larger organizational systems, taking input from, and returning output to, their organizational environments.

3. What is decomposition? Coupling? Cohesion?

Decomposition is the process of breaking down a system into its component parts. Coupling is the extent to which subsystems are dependent on each other. Cohesion is the extent to which a system or a subsystem performs a single function.

4. In what way are organizations systems?

Organizations are systems because they are made up of interrelated components working together for a purpose. They take input from and return output to their environments. Organizations can be redesigned through a systems analysis and design process by which system components are replaced while preserving interconnections between components.

5. List the different classes of information systems described in this chapter. How do they differ from one another?

(1) Transaction processing systems, (2) Management information systems, and (3) Decision support systems (for individuals, groups, and executives). Transaction processing systems (TPS) automate data handling about business activities and transactions. TPS systems can generate reports based on data input from transactions and other activities. A management information system can take data from the TPS and convert them into a meaningful aggregated form. They typically use information from several TPSs. Decision support systems provide an interactive environment where decision makers can quickly manipulate data and models of business operations to attempt to make predictions.

6. List and explain the different phases in the systems development life cycle.

A systems development life cycle is a set of phases that mark the development of an information system. The systems development life cycle in the textbook consists of four phases: (1) systems planning and selection; (2) systems analysis; (3) systems design; and (4) systems implementation and operation. During planning and selection, an organization's total information systems needs are analyzed and prioritized, a potential information systems project is identified, and an argument for continuing or not continuing with the project is made. During the analysis phase, the current system is studied, and new requirements are determined and structured according to their interrelationships. After requirements determination, alternative design strategies are generated. During design, the chosen alternative design strategy is converted into logical and physical design specifications. During implementation and operation, the

information system is coded, tested, and installed in the organization. Once the system is put into production, the system is systematically repaired and improved.

7. What is prototyping?

Prototyping is an iterative process of systems development by which requirements are converted to a working system, which is continually revised through close work between an analyst and users.

8. What are CASE tools? What is a CASE repository and how is it used?

CASE tools provide automated support for some portion of the systems development life cycle. CASE tools include diagramming tools, computer display and report generators, analysis tools, a central repository, documentation generators, and code generators. CASE tools are built around a repository that contains all of the metadata (such as data names, format, uses, and location) for data elements as well as the system specifications. CASE tools automate the repository for easier updating. CASE tools automate routine tasks to help programmers and analysts to do their jobs better.

9. What is JAD? What is Participatory Design?

Joint Application Design (JAD) is a group process involving users and systems development staff in which all parties discuss the needs for an information system and reach a shared understanding. Participatory Design (PD) is a systems development approach that originated in northern Europe in which users and the improvement in their work lives is the central focus.

10. What is RAD? How does it compare to the typical SDLC?

Rapid application development (RAD) is a systems development methodology created to decrease the time needed to design and implement an information system. One of the hallmarks of RAD is extensive user involvement through traditional JAD sessions as well as evaluation of prototypes. RAD also includes integrated CASE tools and code generators.

11. What are Agile Methodologies?

Agile Methodologies promote a self-adaptive software development process. While other methodologies focus on roles that individuals play in a project team, Agile Methodologies focus more on the individual. As software is developed, the process used to develop it is refined and improved through a review process done by the development team. Agile Methodologies differ from traditional system development methodologies in that there is less focus on process and more focus on the people involved in the project.

Problems and Exercises Solutions

- 1. Why is it important to use systems analysis and design methodologies when building a system? Why not just build the system in whatever way seems to be “quick and easy”? What value is provided by using an “engineering” approach?**

Methodologies, techniques, and tools help ensure the quality and appropriateness of the system being built. Following a systems methodology, applying techniques, and using appropriate tools provides structure to the systems development process, since they have been tested and perfected by others. The quick and easy approach to building systems may be easier, cheaper, and quicker in the short run, but it almost always results in a poorly developed system, meaning that the system will be less than optimal and require extra work to maintain. In the long run, a poorly developed system requires more time and money to make right. Following an engineering-type approach ensures that systems analysis and design is rigorous, structured, and systematic.

2. Describe your university or college as a system. What is the input? The output? The boundary? The components? Their interrelationships? The constraints? The purpose? The interfaces? The environment? Draw a diagram of this system.

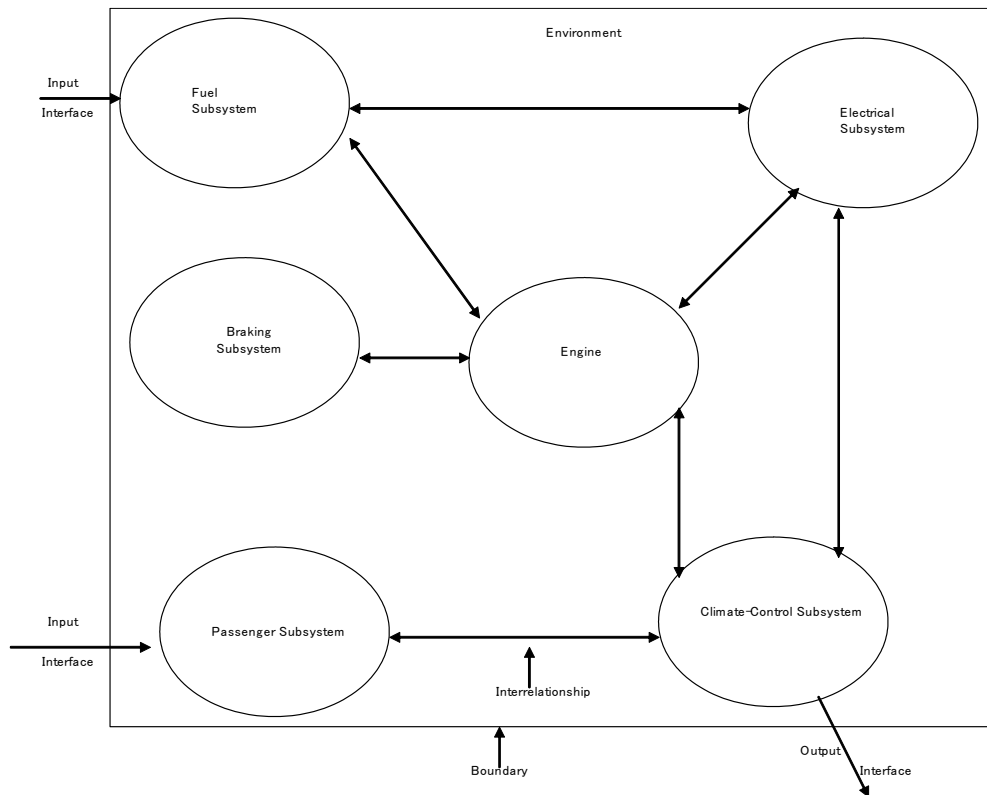
Students will identify a variety of inputs and outputs for the selected university or college. Possible inputs include high school transcripts, applications, tuition payments, and state and federal regulations. Outputs include diplomas, transcripts, billing statements, and inventions. A university's boundary is more difficult to define. Does the physical boundary of the campus serve as the logical boundary for the organization? What if the school delivers outreach education in the community, state, or region? What if the school delivers technology-based distance education across the globe? How would you classify a university-sponsored high-tech start-up business that is not located on campus?

It should be easy for students to list the components of a university. They typically have "business" functions, such as procurement, facilities management, and accounting. In addition, they have academic colleges and departments, and they have academic functions such as registration and advising. Universities are usually organized along a functional hierarchy much like traditional business organizations, with vertical reporting relationships and interdisciplinary committees and task forces for horizontal coordination. Nearly all universities are faced with constraints on funding.

Many universities are also constrained by their state-granted mission. For example, they may be defined by state law as being an exclusively teaching or research institution. Alternatively, state law may mandate from where and what types of students may be admitted. The mission of most universities includes providing education, conducting research, and/or serving their communities. Universities interact with other universities, community colleges, high schools, business organizations, professional organizations, alumni, and many other external entities. The interfaces with these external entities are sometimes formal and sometimes informal. Some examples of formal interfaces include research collaborations between professors and researchers in business or "shadowing" programs where business faculty or students go into the field and learn from a business executive.

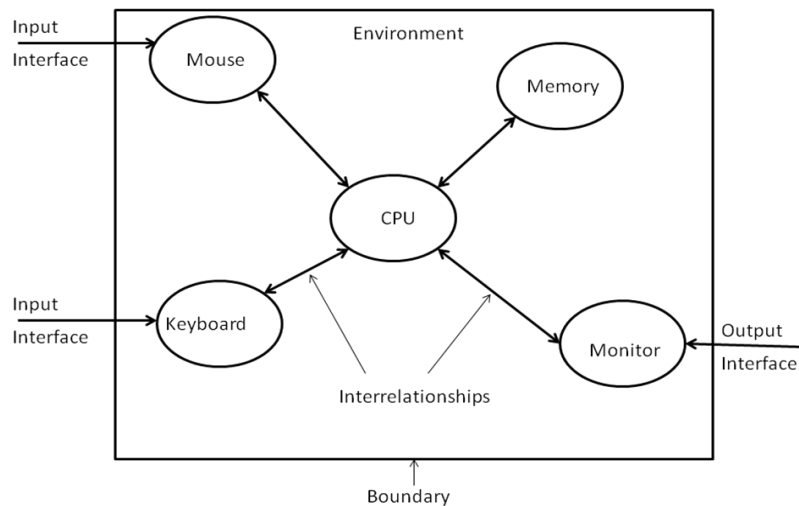
3. **A car is a system with several subsystems, including the braking subsystem, the electrical subsystem, the engine, the fuel subsystem, the climate-control subsystem, and the passenger subsystem. Draw a diagram of a car as a system and label all of its system characteristics.**

The purpose of this question is to encourage your students to begin viewing various entities as systems, and decompose these systems into their components. A simple diagram, showing examples of the car system's characteristics, is provided below. The car's purpose is to provide safe transportation to wherever the driver needs to go. A constraint is the number of passengers that can ride in the vehicle.



4. Your personal computer is a system. Draw and label a personal computer as a system as you did for a car in Problem and Exercise 3.

Student answers for this question will vary. However, a suggested answer, showing examples of a personal computer system's characteristics, is provided below. The purpose of the computer is to perform operations for the user. A constraint for the system is its processing speed.



- 5. Choose a business transaction you undertake regularly, such as using an ATM machine, buying groceries at the supermarket, or buying a ticket for a university's basketball game. For this transaction, define the data, draw the data-flow diagram, and describe processing logic.**

For an ATM transaction, the data include customer name, customer account number, customer personal identification number, customer account balance, transaction type, and transaction amount. At this point, the student should not be expected to know the structure or nomenclature of a data-flow diagram or of processing logic. For the ATM example, they should be able to explain that a customer's name is read from the account identification number on his/her ATM card. Customers input their personal identification number by hand and, if this number is matched with their account identification number, they are granted access to begin an ATM transaction. They will either request to inquire into the status of their account, withdraw money, or deposit money. If, for example, they request to withdraw money, their request will be matched with their available funds and the allowable daily limit for that ATM machine. If acceptable, the cash will be dispensed, their account will be debited, and a receipt will be provided. Do not worry whether or not the student knows the technique or nomenclature at this point. It is more important that the student can analyze the transaction, break it down into its component parts and pieces of data, and understand the process.

- 6. How is the Joint Application Design (JAD) approach different from the Participatory Design (PD) approach developed in northern Europe? (You may have to do some digging at the library to answer this question adequately.) What are the benefits in using approaches like this in building information systems? What are the barriers?**

Both Joint Application Design and Participatory Design are development processes designed to help build better systems by engaging the direct participation of users. The primary difference between them lies in the locus of control for systems development. With the JAD approach, control of systems development typically still rests with the systems staff. Indeed, the outputs from JAD sessions are commonly summarized and handled by the systems staff after users have a chance to review the transcripts. With the PD approach, control of systems development is either shared by systems personnel and users or rests solely with the users and their managers. With the movement toward end-user development in the United States, we are moving more toward shared control of systems and systems development. The benefits to the JAD and PD approaches are that they are likely to result in better systems and higher user commitment to the systems than would be the case if these techniques were not used. Some of the barriers to these approaches are that they require extra systems analyst skills and knowledge; in the short run they add more time and expense to the systems development process, and they require more time and effort from already busy users and user managers.

- 7. How would you organize a project team of students to work with a small business client? How would you organize a project team if you were working for a professional consulting organization? How might these two methods of organization differ? Why?**

The student project team should be of a size that is adequate for the task at hand. The team members should also possess the necessary set of skills and experience for the task at hand. It is important that there be diversity of skills and abilities across team members, but it is also helpful if the team members have some common interests and values on which to build collegiality and trust. It is not necessary that there be a clearly defined “leader” for the team (leadership can rotate by time or phase), but there should be clearly defined roles and responsibilities for each of the team members. There should also be a reward structure (for example, a grade) that promotes shared responsibility and accountability. Finally, because their project is for a small business client, the team members must act professionally and deliver a quality product on time. Surprisingly, these steps do not necessarily change much for organizing a project team within a professional consulting organization. One difference might be that, because you would be concerned more with the long-term professional growth of the team members, you might make team member selections and project assignments that take into account the long-term career development of each of the team members.

8. How might prototyping be used as part of the SDLC?

Prototyping is useful for requirements determination, helping to clarify and communicate user requirements. Also, a prototype can serve as the basis for the final system. Imagine that an analyst is helping to develop a system that will enable a sales representative to access information about inventory levels in real time rather than having to phone someone in production who then physically checks inventory levels and calls the sales representative back. The analyst might begin by asking the sales representative what kinds of information about inventory he/she needs, including when and where he/she needs to access this information. The analyst might then use a graphical, object-oriented development tool, such as Microsoft’s VB.Net, to quickly build some sample interface displays that meet the sales representative’s needs. The analyst would then have the sales representative review these displays and give the analyst feedback. The analyst could then modify the displays and, again, solicit feedback from the sales representative. Given the ease-of-use of VB.Net, the analyst could even build the sample interface “on-the-fly” with the sales representative present and helping to build the displays. The sample interface could then be used to build the actual system, either in VB.Net or in some other development environment.

9. Describe the difference in the role of a systems analyst in the SDLC versus prototyping?

The role of the systems analyst in the SDLC is essentially the same as that in prototyping. The primary difference is that in prototyping, the analyst is simultaneously performing tasks from the analysis, logical design, and physical design phases of the SDLC. In cases where all or part of the prototype will be used for the actual system, the analyst is also performing tasks from the implementation phase of the SDLC. In cases where the analyst builds the prototype with the direct, real-time assistance of the users, the analyst and users are collaboratively completing several steps of the SDLC in one step.

10. Compare Figures 1–10 and 1–11. What similarities and differences do you see?

Figure 1–11 illustrates an evolutionary model of the SDLC, and Figure 1–10 illustrates the circular nature of the systems development life cycle. While both figures convey the iterative nature of systems analysis and design, Figure 1–11 implies that we cycle through the systems development life cycle at varying levels of detail.

Discussion Question Solutions

- 1. If someone at a party asked you what a systems analyst was and why anyone would want to be one, what would you say? Support your answer with evidence from this chapter.**

A good starting point for answering this question is to discuss the importance of an organization's information systems and the systems analyst's role in the systems development process. A systems analyst is a problem solver. The challenge of tackling a problem or opportunity, designing a solution, and implementing a viable, valuable, and improved information system has much appeal. Additionally, the results of the analyst's work directly impact how the organization operates and whether or not the organization can achieve and maintain a competitive advantage. In addition, the analyst is paid quite well.

- 2. Explain how a computer-based information system designed to process payroll is a specific example of a system. Be sure to account for all nine components of any system in your explanation.**

A payroll system can be thought of as a system because it is composed of interrelated subsystems that work together to accomplish a purpose. A payroll system has many components, including components that generate paychecks, make direct deposits, generate various internal reports, process time cards, and process various forms and tax returns. Many of these subsystems or components are related. For instance, the reporting component interacts with the tax, paycheck, and direct deposit components. A direct deposit made to an employee's bank account is an example of an interface. The payroll system's boundary encompasses all of the payroll system's activities. Employees, banks, and the local, state, and federal governments are part of the payroll system's environment. Hours worked, number of dependents, and number of sick days are examples of input to the payroll system. The various reports, paychecks, and tax forms are types of output. Federal and state tax regulations are examples of system constraints.

3. How does the Internet, and more specifically the World Wide Web, fit into the picture of systems analysis and systems development drawn in this chapter?

The Internet has opened up a new frontier for organizations, enabling them to compete on a global basis. This new frontier of electronic commerce is having a tremendous impact on the way that organizations are conducting business. Companies are using the World Wide Web to conduct business with their customers and suppliers, as well as facilitate internal operations. Companies need to reengineer their information systems to take advantage of the World Wide Web and all that it offers. Systems analysis and design is at the heart of this required change.

4. What do you think systems analysis and design will look like in the next decade? As you saw earlier in the chapter, changes in systems development have been pretty dramatic in the past. A computer programmer suddenly transported from the 1950s to the 2000s would have trouble recognizing the computing environment that had evolved just 50 years later. What dramatic changes might occur in the next 10 years?

As the textbook suggests, much of today's systems development work focuses on systems integration, which is likely to continue in the coming years. Companies are recognizing the tremendous benefits of electronic commerce and the World Wide Web, so we can expect a continued movement in that area. Over the past 50 years, we have seen much improvement in the methodologies, techniques, and tools that are used for systems development work. This trend should continue. You should encourage students to find articles about the future of systems analysis and design. Many articles are available on the Web. These articles will serve as a launching pad for class discussion.

Case Problem Solutions

1. Pine Valley Furniture Case Exercises

a. How did Pine Valley Furniture go about developing its information systems? Why do you think the company chose this option? What other options were available?

As mentioned in the scenario, Pine Valley Furniture developed its applications in-house. Many reasons are plausible for why the company chose this option. The company may have had unique processing needs that required the system to be built in-house, as opposed to purchasing a prepackaged system. The company may also have viewed its information systems as helping it achieve a competitive advantage. Pine Valley Furniture had several options, including purchasing a system off-the-shelf, implementing an enterprise-wide system, or hiring an information technology services firm to develop its information systems.

- b. One option available to Pine Valley Furniture was an enterprise-wide system. What features does an enterprise-wide system, such as SAP, provide? What is the primary advantage of an enterprise-wide system?**

To answer this question, encourage your students to visit a Web site devoted to enterprise-wide systems. Information about SAP can be found at its Web site: <http://www.sap.com>. You may wish to have your students compare enterprise-wide systems, such as SAP and PeopleSoft. Information about PeopleSoft can be obtained at <http://www.oracle.com/applications/peoplesoft-enterprise.html>.

Enterprise-wide systems enable companies of all sizes to better manage their financial, human resources, sales, production, and distribution processes. The primary advantage of an enterprise-wide system is its ability to integrate information across the organization.

- c. Pine Valley Furniture will be hiring two systems analysts next month. Your task is to develop a job advertisement for these positions. Locate several Web sites and/or newspapers that have job advertisements for systems analysts. What skills are required?**

Encourage your students to classify the required skills into four categories: analytical, technical, management, and interpersonal. A good place to start looking on the Web is at <http://www.monster.com>. Students should be able to find numerous systems analyst job advertisements at this Web site.

- d. What types of information systems are currently utilized at Pine Valley Furniture? Provide an example of each.**

Pine Valley Furniture is currently using transaction processing systems, management information systems, and decision support systems. Although not specified in the case scenario, students should be able to quickly identify order processing, management reporting, and forecasting as examples of the different types of information systems.

2. Hoosier Burger Case Exercises

- a. Apply the SDLC approach to Hoosier Burger.**

The systems development life cycle can be used to analyze, develop, and support Hoosier Burger's information systems. During systems planning and selection, Bob, Thelma, and the analyst, recognize the need for improvement in the existing Hoosier Burger systems. This need will translate itself into several projects, such as new order-taking, inventory control, and management reporting systems. Also, needs are prioritized, a scope is identified, and feasibility is assessed.

During systems analysis, the analyst examines Hoosier Burger to determine system requirements, structure these requirements, and generate alternative design strategies. During systems design, both logical and physical designs are prepared. During logical design, the analyst concentrates on the business aspects of Hoosier Burger. During physical design, Hoosier Burger's logical design is translated into physical design specifications. During systems implementation and operation, the design specifications for the new Hoosier

Burger system become a working system, and modifications to the new information system are made when warranted.

b. Using the Hoosier Burger scenario, identify an example of each system characteristic.

The ordering system is an example of a component; the order and inventory systems are interrelated components. The boundary encompasses the ordering system, inventory system, and management reporting system. Hoosier Burger's purpose is to make a profit for its owners and to provide quality products and services to its customers. Customers, suppliers, funding agencies, and regulatory agencies exist in Hoosier Burger's environment. An interface exists between the customer and the counter. An order serves as input; a sales receipt serves as output. Constraints would include health regulations.

c. Decompose Hoosier Burger into its major subsystems.

Hoosier Burger has four major subsystems. These systems are order-taking, food preparation, inventory, and management reporting. Students may classify these subsystems differently; however, the goal is for students to decompose the Hoosier Burger system into its primary subsystems.

d. Briefly summarize the approaches to systems development discussed in this chapter. Which approach do you feel should be used by Hoosier Burger?

The textbook discusses several approaches, including systems development life cycle (SDLC), prototyping, computer-aided software engineering (CASE), joint application design (JAD), rapid application development (RAD), participatory design (PD), and Agile Methodologies. The SDLC is a series of steps used to mark the phases of development for an information system. Prototyping involves iteratively designing and building a scaled-down working model of a desired system. CASE tools provide automated support for the systems development process. Joint Application Design is a structured process that brings together end users, managers, and analysts in an effort to identify system requirements and review system designs. Rapid Application Development quickly builds systems through user involvement, JAD sessions, prototyping, integrated CASE tools, and code generators. Participatory design is a systems development approach that originated in Northern Europe in which users and the improvement in their work-lives is the central focus. Agile Methodologies focus on adaptive methodologies, people, and a self-adaptive process. Systems development at Hoosier Burger will probably use the traditional SDLC. However, prototyping can be used to build working models of the system.

3. Natural Best Health Food Stores Case Exercises

a. Identify the different types of information systems used at Natural Best Health Food Stores. Provide an example of each. Is an expert system currently used? If not, how could Natural Best benefit from the use of such a system?

From the description provided above, students should recognize the existence of

a transaction processing system, a management information system, and a decision support system. The transaction processing system is responsible for taking customer orders, scheduling deliveries, and updating inventory. The management information system is responsible for generating several management reports, including sales summary, delivery, and low-in-stock reports for each store location. Mr. Davis uses an Excel worksheet for decision support.

The case scenario does not specifically mention an expert system. However, students should identify several potential uses of an expert system, including using the expert system to make product recommendations for customers.

b. Figure 1–4 identifies seven characteristics of a system. Using the Natural Best Health Food Stores scenario, provide an example of each system characteristic.

Inventory management and sales are two system components. Inventory management and sales are interrelated, because a product cannot be sold unless it is in stock or placed on back order. On a physical level, the boundary is defined by each store building. On a systems level, the boundary can be defined by the NBDS system activities, such as transaction processing, management reporting, and facilitating decision making. The purpose of Natural Best Health Food Stores is to provide healthy, quality products to its customers at an affordable price. The environment includes customers, suppliers, banks, and regulatory agencies. Interfaces include order taking, product deliveries, management reports, and low-in-stock notifications. Inputs include the order information, packing slips, and customer profiles. Outputs include sales receipts, delivery slips, and management reports. Constraints include the number of deliveries that can be made on a given day, lack of data exchange between stores, and the inability to deliver prepackaged meals at the present time.

c. What type of computing environment does Natural Health Food Best Stores have?

Natural Best Health Food Stores currently has a client/server environment. The clients are responsible for capturing the sales data. Data are then transmitted to a central server, which is responsible for processing this data and updating the central database.