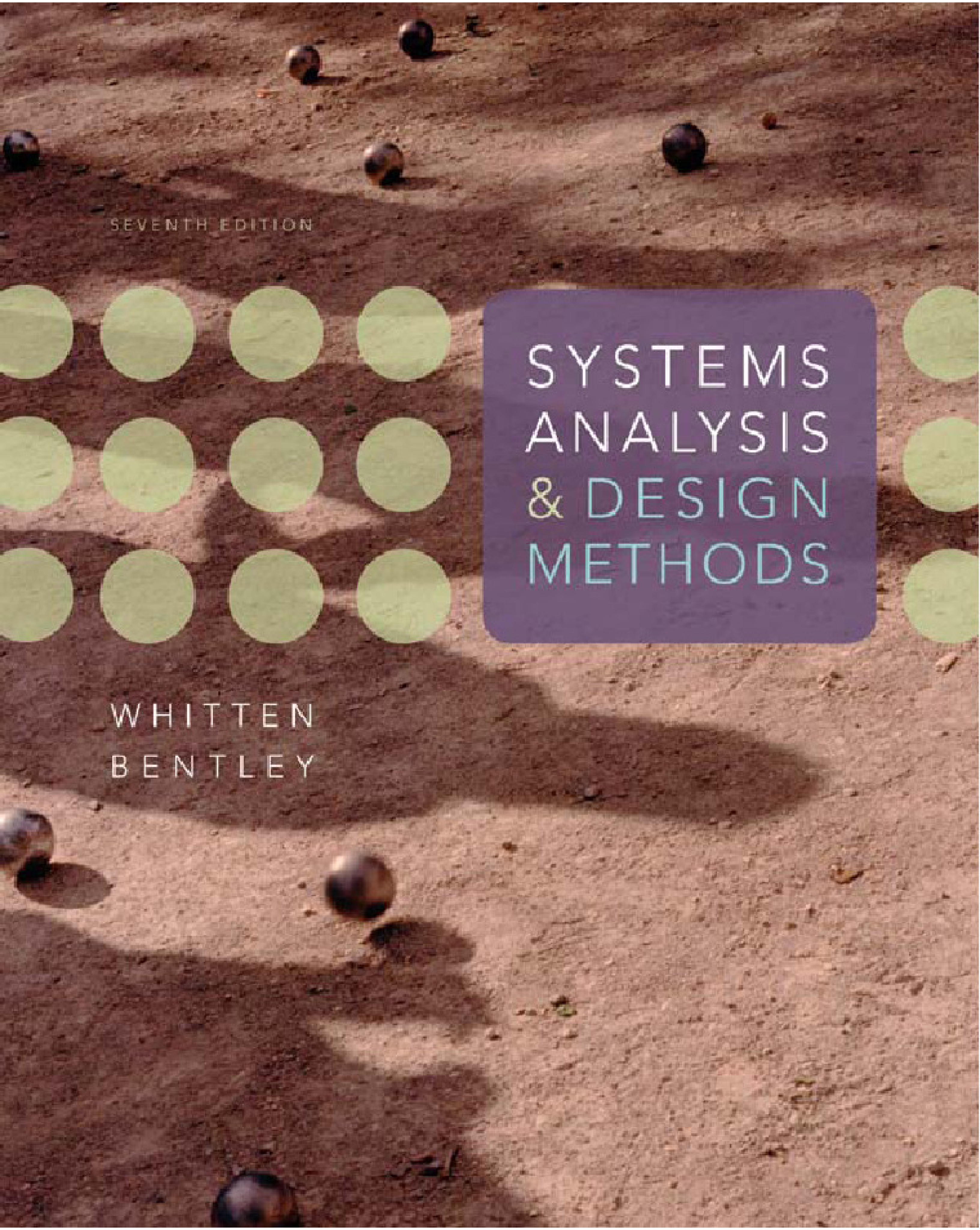

SEVENTH EDITION
SYSTEMS ANALYSIS & DESIGN METHODS
WHITTEN
BENTLEY

Part Two

Systems Analysis Methods

The five chapters in Part Two introduce you to systems analysis activities and methods. Chapter 5, "Systems Analysis," provides the context for all the subsequent chapters by introducing the activities of *systems analysis.* Systems analysis is the most critical phase of a project. During systems analysis we learn about the existing business system, come to understand its problems, define objectives for improvement, and define the detailed business requirements that must be fulfilled by *any* subsequent technical solution. Clearly, any subsequent system design and implementation of a new system depends on the quality of the preceding systems analysis. Systems analysis is often shortchanged in a project because (1) many analysts are not skilled in the concepts and logical modeling techniques to be used, and (2) many analysts do not understand the significant impact of those shortcuts. Chapter 5 introduces you to systems analysis and its overall importance in a project. Subsequent chapters teach you specific systems analysis skills with an emphasis on logical system modeling.

Chapter 6, "Fact-Finding Techniques for Requirements Discovery," teaches various fact-finding techniques and strategies used to solicit user requirements for a new system.

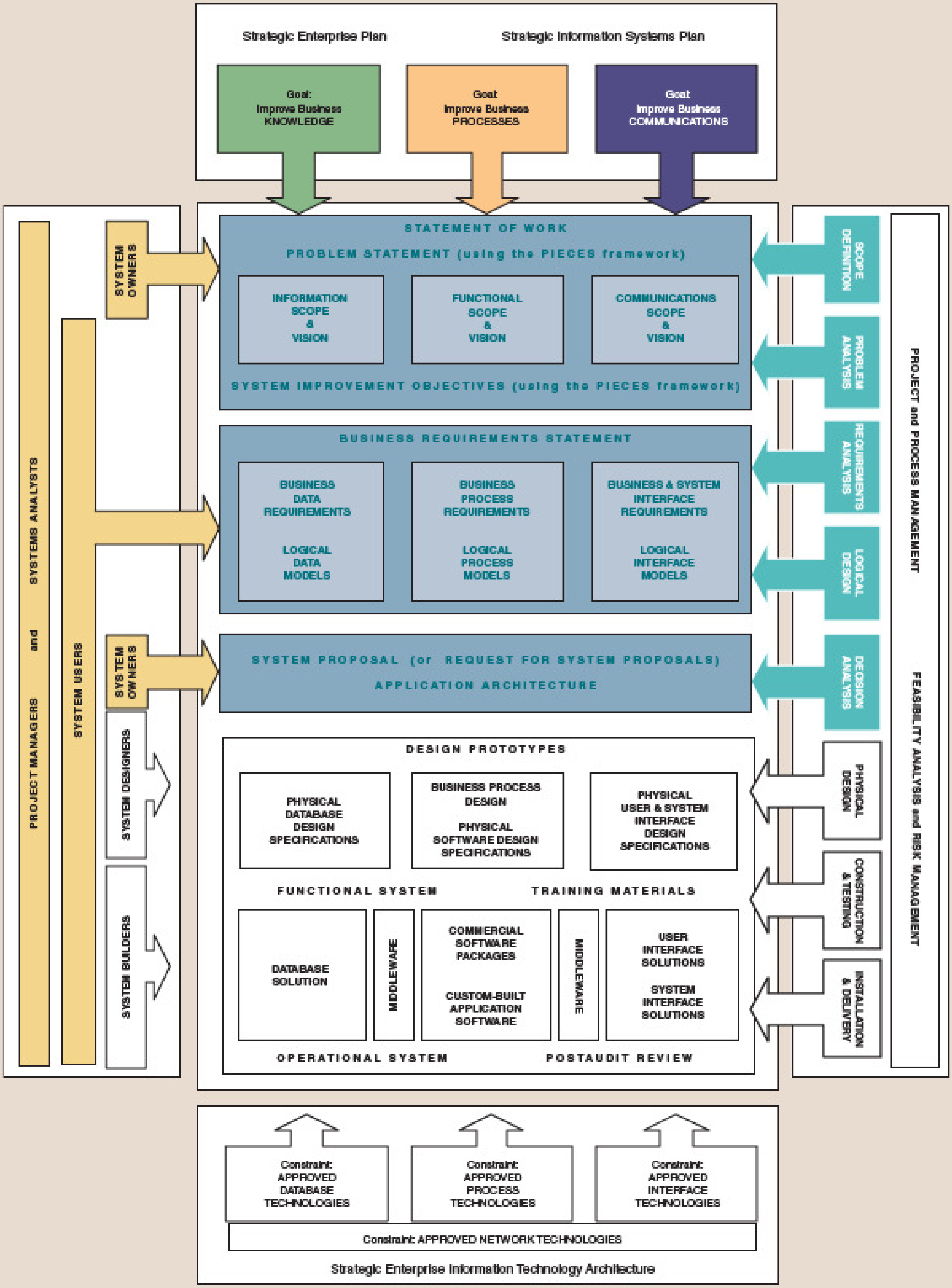
In Chapter 7, "Modeling System Requirements with Use Cases," you will learn about the tools and techniques necessary to perform use-case modeling to document system requirements.

In Chapter 8, "Data Modeling and Analysis," we teach you *data modeling,* a technique for organizing and documenting the stored data requirements for a system. You will learn to draw entity relationship diagrams as a tool for structuring business data that will eventually be designed as a database. These models will capture the business associations and rules that must govern the data.

Chapter 9, "Process Modeling," introduces *process modeling.* It explains how data flow diagrams can be used to depict the essential business processes in a system, the flow of data through a system, and policies and procedures to be implemented by processes. If you've done any programming, you recognize the importance of understanding the business processes for which you are trying to write the programs.

Chapter 10, "Object-Oriented Analysis and Modeling with UML," teaches you about the object-oriented approach to performing systems analysis using UML tools.

Chapter 11, "Feasibility Analysis and the System Proposal," teaches you how to brainstorm possible system solutions, analyze those solutions for feasibility, select the best overall solution, and then present your recommendation in the form of a written and oral proposal to management.



Strategic Enterprise Plan
Strategic Information Systems Plan
Goal: Improve Business KNOWLEDGE
Goal: Improve Business PROCESSES
Goal: Improve Business COMMUNICATIONS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
BUSINESS DATA REQUIREMENTS
LOGICAL DATA MODELS
BUSINESS PROCESS REQUIREMENTS
LOGICAL PROCESS MODELS
BUSINESS & SYSTEM INTERFACE REQUIREMENTS
LOGICAL INTERFACE MODELS
SYSTEM PROPOSAL (or REQUEST FOR SYSTEM PROPOSALS)
APPLICATION ARCHITECTURE
DESIGN PROTOTYPES
PHYSICAL DATABASE DESIGN SPECIFICATIONS
BUSINESS PROCESS DESIGN
PHYSICAL SOFTWARE DESIGN SPECIFICATIONS
PHYSICAL USER & SYSTEM INTERFACE DESIGN SPECIFICATIONS
FUNCTIONAL SYSTEM
TRAINING MATERIALS
DATABASE SOLUTION
MIDDLEWARE
COMMERCIAL SOFTWARE PACKAGES
CUSTOM-BUILT APPLICATION SOFTWARE
MIDDLEWARE
USER INTERFACE SOLUTIONS
SYSTEM INTERFACE SOLUTIONS
OPERATIONAL SYSTEM
POSTAUDIT REVIEW
PROJECT MANAGERS and SYSTEMS ANALYSTS
SYSTEM USERS
SYSTEM OWNERS
SYSTEM OWNERS
SYSTEM DESIGNERS
SYSTEM BUILDERS
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
DECISION ANALYSIS
PHYSICAL DESIGN
CONSTRUCTION & TESTING
INSTALLATION & DELIVERY
PROJECT and PROCESS MANAGEMENT
FEASIBILITY ANALYSIS and RISK MANAGEMENT
Constraint: APPROVED DATABASE TECHNOLOGIES
Constraint: APPROVED PROCESS TECHNOLOGIES
Constraint: APPROVED INTERFACE TECHNOLOGIES
Constraint: APPROVED NETWORK TECHNOLOGIES
Strategic Enterprise Information Technology Architecture

5 Systems Analysis

Chapter Preview and Objectives

In this chapter you will learn more about the systems analysis phases in a systems development project—namely, the scope definition, problem analysis, requirements analysis, and decision analysis phases. The first three phases are collectively referred to as *systems analysis*. The latter phase provides transition between systems analysis and systems design. You will know that you understand the process of systems analysis when you can:

- Define systems analysis and relate the term to the scope definition, problem analysis, requirements analysis, logical design, and decision analysis phases of this book's systems development methodology.
- Describe a number of systems analysis approaches for solving business system problems.
- Describe the scope definition, problem analysis, requirements analysis, logical design, and decision analysis phases in terms of your information system building blocks.
- Describe the scope definition, problem analysis, requirements analysis, logical design, and decision analysis phases in terms of purpose, participants, inputs, outputs, techniques, and steps.
- Identify the chapters in this textbook that can help you learn specific systems analysis tools and techniques.

NOTE: Although some of the tools and techniques of systems analysis are previewed in this chapter, it is *not* the intent of this chapter to teach those tools and techniques. This chapter teaches only the *process* of systems analysis. The tools and techniques will be taught in the subsequent six chapters.

Introduction

Bob Martinez remembers learning in college that systems analysis defines what an information system needs to do while system design defines how it needs to do it. At the time, it sounded like a simple two-step process. Now, as he begins working on the SoundStage Member Services system project, he sees that there are multiple phases and several steps within systems analysis and system design.

The SoundStage project is at the beginning of systems analysis, in what Sandra, his boss, calls the scope definition phase. After that they'll do problem analysis, requirements analysis, and decision analysis. It sounds like a lot of work just to understand *what* the system needs to do. But this is a complicated system. As Sandra says, would you build a house without a good set of plans?

What Is Systems Analysis?

systems analysis a problem-solving technique that decomposes a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose.

systems design a complementary problem-solving technique (to systems analysis) that reassembles a system's component pieces back into a complete system—hopefully, an improved system. This may involve adding, deleting, and changing pieces relative to the original system.

information systems analysis those development phases in an information systems development project that primarily focus on the business problem and requirements, independent of any technology that can or will be used to implement a solution to that problem.

repository a location (or set of locations) where systems analysts, systems designers, and system builders keep all of the documentation associated with one or more systems or projects.

In Chapter 3 you learned about the systems development process. In that chapter we purposefully limited our discussion to only briefly examining each phase. In this chapter, we take a much closer look at those phases that are collectively referred to as **systems analysis.** Formally defined in the margin, systems analysis is the study of a system and its components. It is a prerequisite to **systems design,** the specification of a new and improved system. This chapter will focus on systems analysis. Chapter 12 will do the same for systems design.

Moving from this classic definition of systems analysis to something a bit more contemporary, we see that *systems analysis* is a term that collectively describes the early phases of systems development. Figure 5-1 uses color to identify the systems analysis phases in the context of the full classic route for our *FAST* methodology (from Chapter 3). There has never been a universally accepted definition of systems analysis. In fact, there has never been universal agreement on when information systems analysis ends and when information systems design begins. For the purpose of this book, **information systems analysis** emphasizes *business* issues, *not* technical or implementation concerns.

Systems analysis is driven by the business concerns of SYSTEM OWNERS and SYSTEM USERS. Hence, it addresses the KNOWLEDGE, PROCESS, and COMMUNICATIONS building blocks from SYSTEM OWNERS' and SYSTEM USERS' perspectives. The SYSTEMS ANALYSTS serve as facilitators of systems analysis. This context is illustrated in the chapter home page that preceded the objectives for this chapter.

The documentation and deliverables produced by systems analysis tasks are typically stored in a repository. A repository may be created for a single project or shared by all projects and systems. A repository is normally implemented as some combination of the following:

- A network directory of word processing, spreadsheet, and other computer-generated files that contain project correspondence, reports, and data.
- One or more CASE tool dictionaries or encyclopedias (as discussed in Chapter 3).
- Printed documentation (such as that stored in binders and system libraries).
- An *intranet* Web site interface to the above components (useful for communication).

Hereafter, we will refer to these components collectively as the **repository.**

This chapter examines each of our five systems analysis phases in greater detail. But first, let's examine some overall strategies for systems analysis.

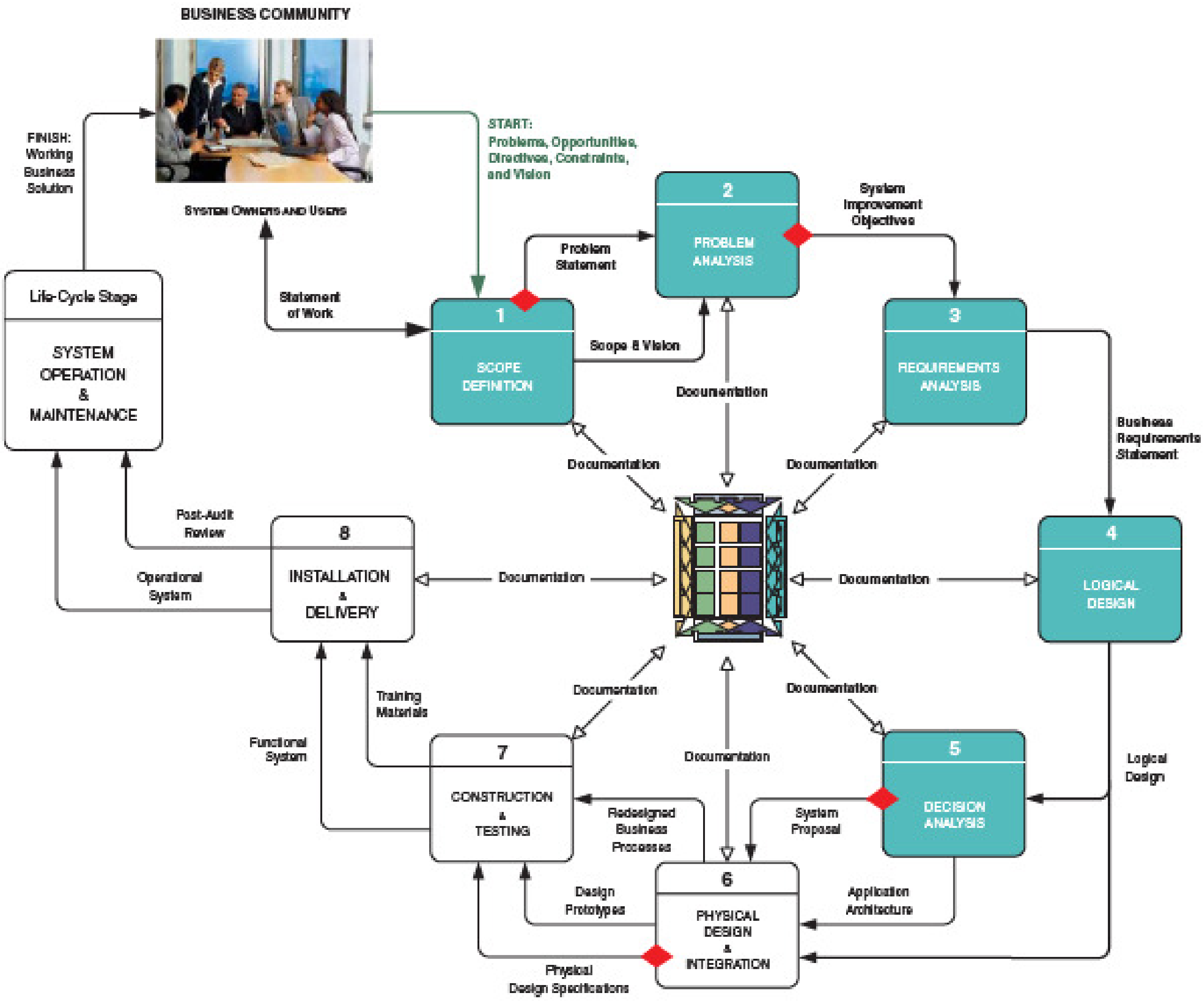


FIGURE 5-1 The Context of Systems Analysis

Systems Analysis Approaches

Fundamentally, systems analysis is about *problem solving.* There are many approaches to problem solving; therefore, it shouldn't surprise you that there are many approaches to systems analysis. These approaches are often viewed as *competing* alternatives. In reality, certain combinations can and should actually complement one another. This was characterized in Chapter 3 as *agile methods.* Let's briefly examine the varied approaches.

NOTE: The intent here is to develop a high-level understanding only. Subsequent chapters in this unit will actually teach you the underlying techniques.

> Model-Driven Analysis Approaches

Structured analysis, information engineering, and object-oriented analysis are examples of **model-driven analysis.** Model-driven analysis uses pictures to communicate

model-driven analysis a problem-solving approach that emphasizes the drawing of pictorial system models to document and validate existing and/or proposed systems. Ultimately, the system model becomes the blueprint for designing and constructing an improved system.

model a representation of either reality or vision. Since "a picture is worth a thousand words," most models use pictures to represent the reality or vision.

business problems, requirements, and solutions. Examples of **models** with which you may already be familiar include flowcharts, structure or hierarchy charts, and organization charts.

Today, model-driven approaches are almost always enhanced by the use of automated tools. Some analysts draw system models with general-purpose graphics software such as Microsoft *Visio*. Other analysts and organizations require the use of repository-based CASE or modeling tools such as *System Architect, Visible Analyst*, or *Rational ROSE*. CASE tools offer the advantage of consistency and completeness analysis as well as rule-based error checking.

Model-driven analysis approaches are featured in the model-driven methodologies and routes that were introduced in Chapter 3. Let's briefly examine today's three most popular model-driven analysis approaches.

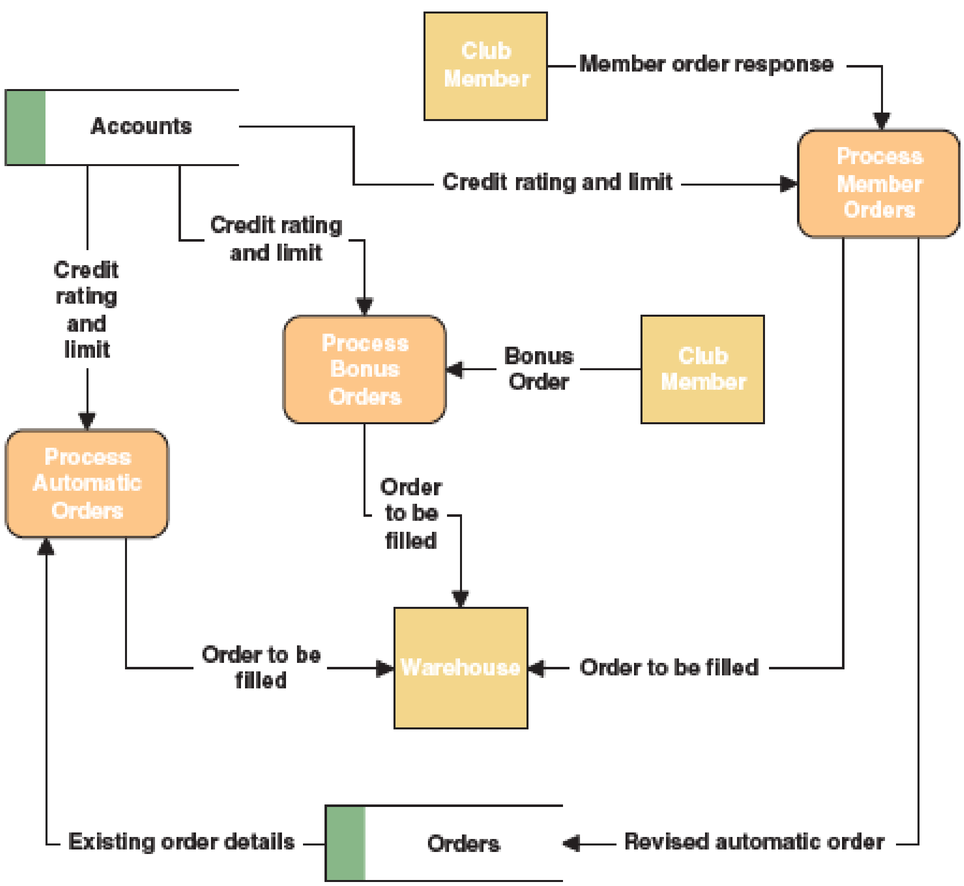
structured analysis a model-driven, PROCESS-centered technique used to either analyze an existing system or define business requirements for a new system, or both. The models are pictures that illustrate the system's component pieces: processes and their associated inputs, outputs, and files.

Traditional Approaches Various traditional approaches to system analysis and design were developed beginning in the 1970s. One of the first formal approaches, which is still widely used today, is *structured analysis*. **Structured analysis** focuses on the flow of data through business and software processes. It is said to be *process-centered*. By process-centered, we mean that the emphasis is on the PROCESS building bocks in your information system framework.

One of the key tools used to model processes is the *data flow diagram* (Figure 5-2), which depicts the existing and/or proposed processes in a system along with their inputs, outputs, and data. The models show the flow of data between and through processes and show the places where data is stored. Ultimately these process models serve as blueprints for business processes to be implemented and software to be purchased or constructed.

FIGURE 5-2

A Simple Process Model (Also Called a Data Flow Diagram)



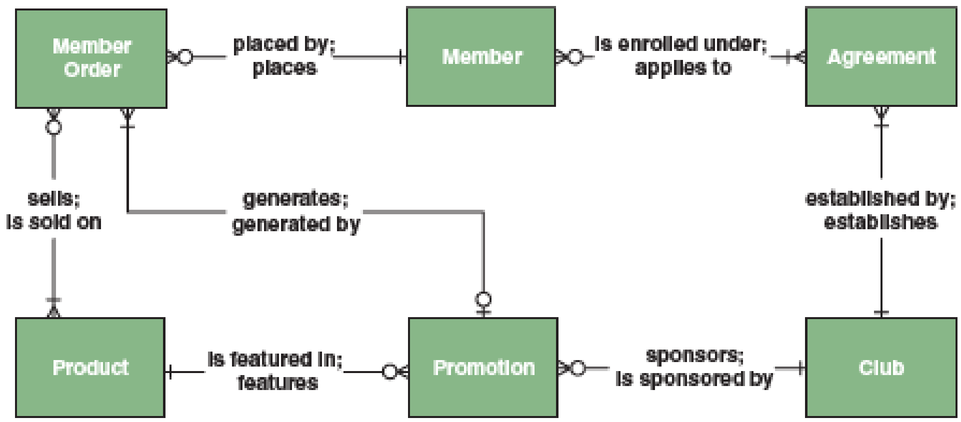


FIGURE 5-3
A Simple Data Model (Also Called an Entity Relationship Diagram)

The practice of structured analysis for software design has greatly diminished in favor of object-oriented methods. However, process modeling is enjoying something of a revival thanks to the renewed emphasis on business process redesign, which is discussed later in this chapter.

Another traditional approach, called **information engineering (IE)**, focuses on the structure of stored data in a system rather than on processes. Thus, it was said to be *data-centered*, emphasizing the analysis of KNOWLEDGE (or data) requirements. The key tool to model data requirements is the entity relationship diagram (Figure 5-3). Entity relationship diagrams are still widely used in designing relational databases.

information engineering (IE) a model-driven and DATA-centered, but PROCESS-sensitive, technique for planning, analyzing, and designing information systems. IE models are pictures that illustrate and synchronize the system's data and processes.

Originally, information engineering was seen as a competing approach to structured analysis. But over time many people made them as complementary: using data flow diagrams to model a system's processes and entity relationship diagrams to model a system's data.

Object-Oriented Approach Traditional approaches deliberately separated the concerns of KNOWLEDGE (data) from those of PROCESSES. Although most systems analysis methods attempted to synchronize data and process models, the attempt did not always work well in practice. **Object** technologies have since emerged to eliminate this artificial separation of data and processes. The **object-oriented approach** views information systems not as data and processes but as a collection of objects that encapsulate data and processes. Objects can contain data attributes. However, the only way to create, read, update, or delete an object's data is through one of its embedded processes (called methods). Object-oriented programming languages, such as *Java*, *C++*, and the *.NET* languages, are becoming increasingly popular.

object the encapsulation of the data (called *properties*) that describes a discrete person, object, place, event, or thing, with all of the processes (called *methods*) that are allowed to use or update the data and properties. The only way to access or update the object's data is to use the object's predefined processes.

The object-oriented approach has a complete suite of modeling tools known as the Unified Modeling Language (UML). One of the UML diagrams, an object class diagram, is shown in Figure 5-4. Some of the UML tools have gained acceptance for systems projects even when the information system will not be implemented with object-oriented technologies.

object-oriented approach a model-driven technique that integrates data and process concerns into constructs called *objects*. Object models are pictures that illustrate the system's objects from various perspectives, such as the structure, behavior, and interactions of the objects.

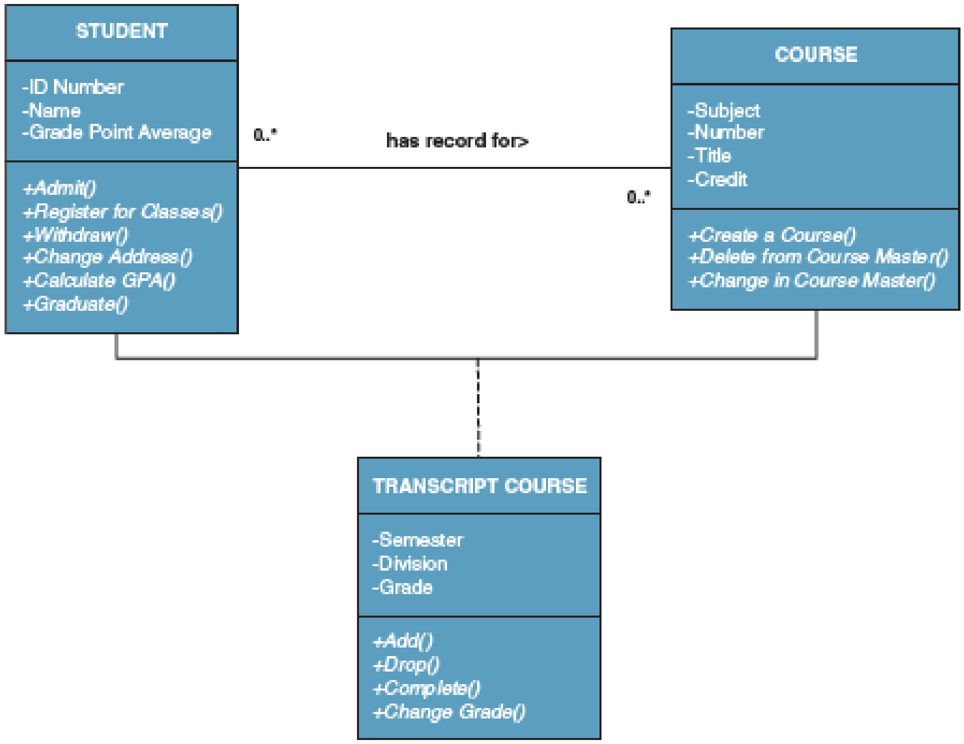
> Accelerated Systems Analysis Approaches

Discovery prototyping and rapid architected development are examples of accelerated systems analysis approaches that emphasize the construction of prototypes to more rapidly identify business and user requirements for a new system. Most such approaches derive from some variation on the construction of **prototypes**, working but incomplete samples of a desired system. Prototypes cater to the "I'll know what I want when I see it" way of thinking that is characteristic of many users and managers. By "incomplete," we mean that a prototype will not include the error checking, input data validation, security, and processing completeness of a finished application. Nor will it be as polished or offer the user help as in a final system. But because it can be

prototype a small-scale, incomplete, but working sample of a desired system.

FIGURE 5-4

An Object Model (Using the Unified Modeling Language Standard)



developed quickly, it can quickly identify the most crucial of business-level requirements. Sometimes, prototypes can evolve into the actual, completed information systems and applications.

In a sense, accelerated analysis approaches place much emphasis on the COMMUNICATIONS building blocks in your information system framework by constructing sample forms and reports. At the same time, the software tools used to build prototypes also address the DATA and PROCESS building blocks.

These accelerated approaches are common in the rapid application development (RAD) methodologies and routes that were introduced in Chapter 3. RAD approaches require automated tools. While some repository-based CASE tools include very simple RAD facilities, most analysts use true RAD programming environments such as Sybase *Powerbuilder*, Microsoft *Access*, Microsoft *Visual Basic .NET*, or IBM *Websphere Studio for Application Development* (*Java*-based).

Let's briefly examine two popular accelerated analysis approaches.

discovery prototyping a technique used to identify the users' business requirements by having them react to a quick-and-dirty implementation of those requirements.

Discovery Prototyping **Discovery prototyping** uses rapid development technology to help users discover their business requirements. For example, it is very common for systems analysts to use a simple development tool like Microsoft *Access* to rapidly create a simple database, user input forms, and sample reports to solicit user responses as to whether the database, forms, and reports truly represent business requirements. The intent is usually to develop the final new system in a more sophisticated application development tool and language, but the simpler tool allows the analyst to more quickly prototype the user's requirements.

In discovery prototyping, we try to discourage users from becoming preoccupied with the final "look and feel" of the system prototypes—that can be changed during system design! Therein lies the primary criticism of prototyping—software templates exist in prototyping tools to quickly generate some very elegant and visually appealing prototypes. Unfortunately, this can encourage a premature focus on, and commitment to, design represented in the prototype. Users can also be misled to believe (1) that the completed system can be built just as rapidly or (2) that tools like *Access* can be used to build the final system. While tools like *Access* can indeed accelerate systems development, their use in discovery prototyping is fast only because we omit most of the detailed database and application programming required for a complete and secure application. Also, tools like *Access* typically cannot support the database sizes, number of users, and network traffic that are required of most enterprise applications.

Regardless, discovery prototyping is a preferred and recommended approach. Unfortunately, some systems analysts and developers are using discovery prototyping to completely replace model-driven design, only to learn what true engineers have known for years: you cannot prototype without some amount of more formal design . . . enter rapid architected analysis.

Rapid Architected Analysis **Rapid architected analysis** is an accelerated analysis approach that also builds system models. Rapid architecture analysis is made possible by **reverse-engineering** technology that is included in many automated tools such as CASE and programming languages (as introduced in Chapter 3). Reverse-engineering tools generate system models from existing software applications or system prototypes. The resulting system models can then be edited and improved by systems analysts and users to provide a blueprint for a new and improved system. It should be apparent that rapid architected analysis is a blending of model-driven and accelerated analysis approaches.

rapid architected analysis an approach that attempts to derive system models (as described earlier in this section) from existing systems or discovery prototypes.

reverse engineering the use of technology that reads the program code for an existing database, application program, and/or user interface and automatically generates the equivalent system model.

There are two different techniques for applying rapid architected analysis:

- Most systems have already been automated to some degree and exist as legacy information systems. Many CASE tools can read the underlying database structures and/or application programs and reverse engineer them into various system models. Those models serve as a point of departure for defining model-driven user requirements analysis.
- If prototypes have been built into tools like Microsoft *Access* or *Visual Basic*, those prototypes can sometimes be reverse engineered into their equivalent system models. The system models usually better lend themselves to analyzing the users' requirements for consistency, completeness, stability, scalability, and flexibility to future change. Also, the system models can frequently be forward engineered by the same CASE tools and ADEs (application development environments) into databases and application templates or skeletons that will use more robust enterprise-level database and programming technology.

Both techniques address the previous issue that engineers rarely prototype in the total absence of a more formal design, and, at the same time, they preserve the advantages of accelerating the systems analysis phases.

> Requirements Discovery Methods

Both model-driven and accelerated systems analysis approaches attempt to express user requirements for a new system, either as models or as prototypes. But both approaches are, in turn, dependent on the more subtle need to actually identify and manage those requirements. Furthermore, the requirements for systems are dependent on the analysts' ability to discover the problems and opportunities that exist in the current system—thus, analysts must become skilled in identifying problems, opportunities, and requirements! Consequently, all approaches to systems analysis require some form of **requirements discovery.** Let's briefly survey a couple of common requirements discovery approaches.

requirements discovery the process, used by systems analysts, of identifying or extracting system problems and solution requirements from the user community.

Fact-Finding Techniques **Fact-finding** is an essential skill for all systems analysts. The fact-finding techniques covered in this book (in fact, in the next chapter) include:

fact-finding the process of collecting information about system problems, opportunities, solution requirements, and priorities. Also called *information gathering.*

- Sampling of existing documentation, reports, forms, files, databases, and memos.
- Research of relevant literature, benchmarking of others' solutions, and site visits.

- Observation of the current system in action and the work environment.
- Questionnaires and surveys of the management and user community.
- Interviews of appropriate managers, users, and technical staff.

joint requirements planning (JRP) the use of facilitated workshops to bring together all of the system owners, users, and analysts and some systems designers and builders to jointly perform systems analysis. JRP is generally considered a part of a larger method called *joint application development (JAD),* a more comprehensive application of the JRP techniques to the entire systems development process.

Joint Requirements Planning The fact-finding techniques listed above are invaluable; however, they can be time-consuming in their classic forms. Alternatively, requirements discovery and management can be significantly accelerated using **joint requirements planning (JRP)** techniques. A JRP-trained or -certified analyst usually plays the role of *facilitator* for a workshop that will typically run from three to five full working days. This workshop can replace weeks or months of classic fact-finding and follow-up meetings.

JRP provides a working environment in which to accelerate all systems analysis tasks and deliverables. It promotes enhanced SYSTEM OWNER and SYSTEM USER participation in systems analysis. But it also requires a facilitator with superior mediation and negotiation skills to ensure that all parties receive appropriate opportunities to contribute to the system's development.

JRP is typically used in conjunction with the model-driven analysis approaches we described earlier, and it is typically incorporated into rapid application development (RAD) methodologies and routes (which were introduced in Chapter 3).

> Business Process Redesign Methods

business process redesign (BPR) the application of systems analysis methods to the goal of dramatically changing and improving the fundamental business processes of an organization, independent of information technology.

One of the most interesting contemporary applications of systems analysis methods is **business process redesign (BPR).** The interest in BPR was driven by the discovery that most current information systems and applications have merely automated existing and inefficient business processes. Automated bureaucracy is still bureaucracy; automation does not necessarily contribute value to the business, and it may actually subtract value from the business. Introduced in Chapter 1, BPR is one of many types of projects triggered by the trends we call *total quality management (TQM)* and *continuous process improvement (CPI)*.

Some BPR projects focus on all business processes, regardless of their automation. Each business process is thoroughly studied and analyzed for bottlenecks, value returned, and opportunities for elimination or streamlining. Process models, such as data flow diagrams (discussed earlier), help organizations visualize their processes. Once the business processes have been redesigned, most BPR projects conclude by examining how information technology might best be applied to the improved business processes. This may create new information system and application development projects to implement or support the new business processes.

BPR is also applied within the context of information system development projects. It is not uncommon for IS projects to include a study of existing business processes to identify problems, bureaucracy, and inefficiencies that can be addressed in requirements for new and improved information systems and computer applications.

BPR has also become common in IS projects that will be based on the purchase and integration of commercial off-the-shelf (COTS) software. The purchase of COTS software usually requires that a business adapt its business processes to fit the software. An analysis of existing business processes during systems analysis is usually a part of such projects.

> *FAST* Systems Analysis Strategies

agile method the integration of various approaches of systems analysis and design for application as deemed appropriate to the problem being solved and the system being developed.

Like most commercial methodologies, our hypothetical *FAST* methodology does not impose a single approach on systems analysts. Instead, it integrates all the popular approaches introduced in the preceding paragraphs into a collection of **agile methods.** The SoundStage case study will demonstrate these methods in the context of a typical first assignment for a systems analyst. The systems analysis techniques will be applied within the framework of:

- Your information system building blocks (from Chapter 2).
- The *FAST* phases (from Chapter 3).
- *FAST* tasks that implement a phase (described in this chapter).

Given this context for studying systems analysis, we can now explore the systems analysis phases and tasks.

The Scope Definition Phase

Recall from Chapter 3 that the *scope definition phase* is the first phase of the classic systems development process. In other methodologies this might be called the *preliminary investigation phase, initial study phase, survey phase,* or *planning phase.* The scope definition phase answers the question, "Is this project worth looking at?" To answer this question, we must define the scope of the project and the perceived problems, opportunities, and directives that triggered the project. Assuming the project *is* deemed worth looking at, the scope definition phase must also establish the project plan in terms of scale, development strategy, schedule, resource requirements, and budget.[1]

The context for the scope definition phase is shaded in Figure 5-5. Notice that the scope definition phase is concerned primarily with the SYSTEM OWNERS' view of the existing system and the problems or opportunities that triggered the interest. System owners tend to be concerned with the big picture, not details. Furthermore, they determine whether resources can and will be committed to the project.

Figure 5-6 is the first of five task diagrams we will introduce in this chapter to take a closer look at each systems analysis phase. A *task diagram* shows the work (= tasks) that should be performed to complete a phase. Our task diagrams do not mandate any specific methodology, but we will describe in the accompanying paragraphs the approaches, tools, and techniques you might want to consider for each task. Figure 5-6 shows the tasks required for the scope definition phase. It is important to remember that these task diagrams are only templates. The project team and project manager may expand on or alter these templates to reflect the unique needs of any given project.

As shown in Figure 5-6, the final deliverable for the preliminary investigation phase is completion of a PROJECT CHARTER. (Such major deliverables are indicated in each task diagram in all-capital letters.) A project charter defines the project scope, plan, methodology, standards, and so on. Completion of the project charter represents the first milestone in a project.

The scope definition phase is intended to be quick. The entire phase should not exceed two or three days for most projects. The phase typically includes the following tasks:

1.1 Identify baseline problems and opportunities.
1.2 Negotiate baseline scope.
1.3 Assess baseline project worthiness.
1.4 Develop baseline schedule and budget.
1.5 Communicate the project plan.

Let's now examine each of these tasks in greater detail.

---

[1]If your course or reading has already included Chapter 4, you should recognize these planning elements as part of project management. Chapter 4 surveyed and demonstrated the process used by project managers to develop a project plan.

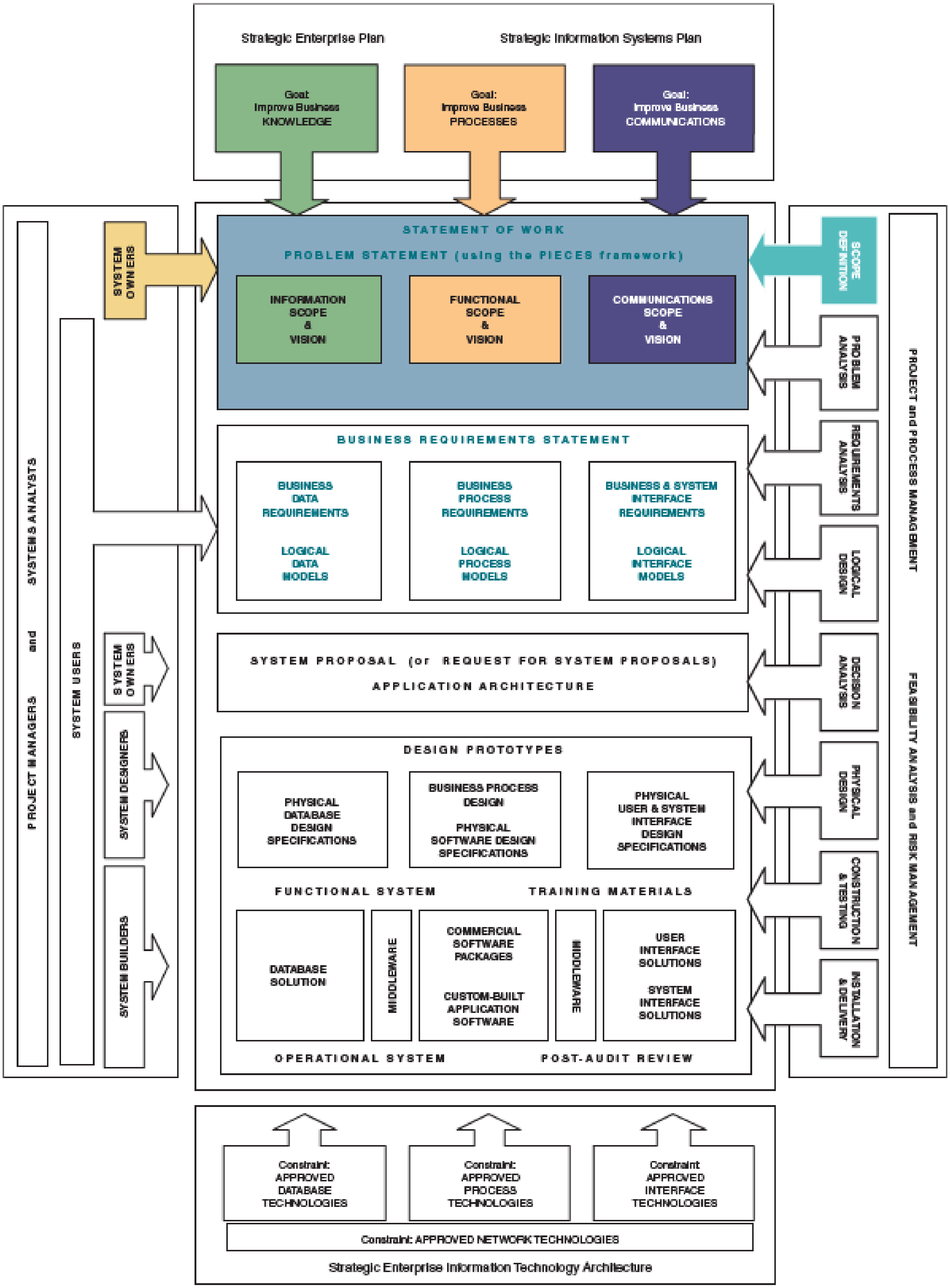


FIGURE 5-5 The Context of the Scope Definition Phase of Systems Analysis

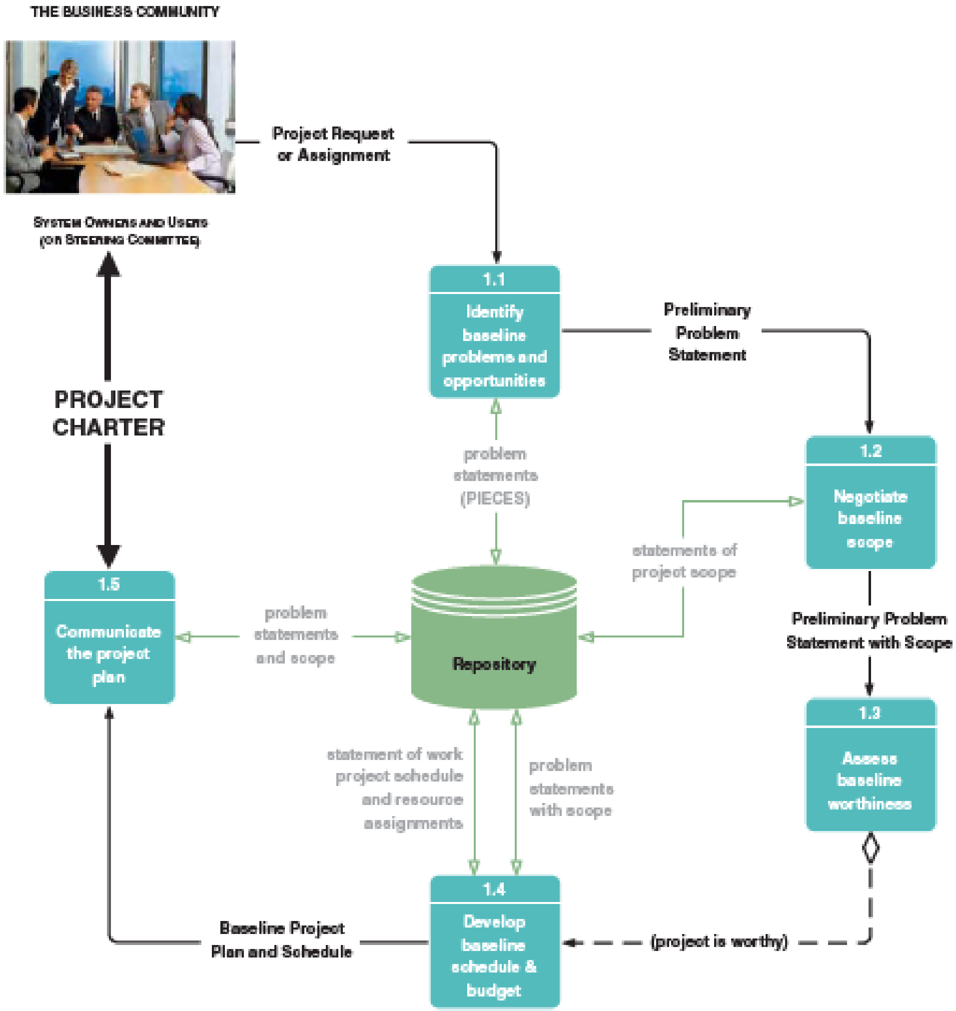


FIGURE 5-6

Tasks for the Scope Definition Phase of Systems Analysis

> Task 1.1—Identify Baseline Problems and Opportunities

One of the most important tasks of the **scope** definition phase is establishing an initial baseline of the problems, opportunities, and/or directives that triggered the project. Each problem, opportunity, and directive is assessed with respect to urgency, visibility, tangible benefits, and priority. Any additional, detailed analysis is not relevant at this stage of the project. It may, however, be useful to list any perceived constraints (limits) on the project, such as deadlines, maximum budget, or general technology.

A senior systems analyst or project manager usually leads this task. Most of the other participants are broadly classified as SYSTEM OWNERS. This includes the executive sponsor(s), the highest-level manager(s) who will pay for and support the project. It also includes managers of all organizational units that may be impacted by the system and possibly includes information systems managers. SYSTEM USERS, SYSTEM DESIGNERS, and SYSTEM BUILDERS are not typically involved in this task.

As shown in Figure 5-6, a PROJECT REQUEST OR ASSIGNMENT triggers the task. This trigger may take one of several alternative forms. It may be as simple as a memorandum of authority from an information systems steering body. Or it may be a memorandum from a business team or unit requesting systems development. Some organizations require that all project requests be submitted on some standard request-for-service form, such as Figure 5-7.

scope the boundaries of a project—the areas of a business that a project may (or may not) address.

FIGURE 5-7
A Request for Systems Services

SoundStage Entertainment Club
Information System Services
Phone: 494-0666 Fax: 494-0999
Internet: http://www.soundstage.com
Intranet: http://www.soundstage.com/iss

REQUEST FOR INFORMATION SYSTEM SERVICES

DATE OF REQUEST	SERVICE REQUESTED FOR DEPARTMENT(S)
January 9, 2003	Member Services, Warehouse, Shipping

SUBMITTED BY (key user contact)		EXECUTIVE SPONSOR (funding authority)	
Name	Sarah Hartman	Name	Galen Kirkhoff
Title	Business Analyst, Member Services	Title	Vice President, Member Services
Office	B035	Office	G242
Phone	494-0987	Phone	494-1242

TYPE OF SERVICE REQUESTED:

- ☐ Information Strategy Planning
- ☒ Business Process Analysis and Redesign
- ☒ New Application Development
- ☐ Other (please specify)__________
- ☐ Existing Application Enhancement
- ☐ Existing Application Maintenance (problem fix)
- ☐ Not Sure

BRIEF STATEMENT OF PROBLEM, OPPORTUNITY, OR DIRECTIVE (attach additional documentation as necessary)
The information strategy planning group has targeted member services, marketing, and order fulfillment (inclusive of shipping) for business process redesign and integrated application development. Currently serviced by separate information systems, these areas are not well integrated to maximize efficient order services to our members. The current systems are not adaptable to our rapidly changing products and services. In some cases, separate systems exist for similar products and services. Some of these systems were inherited through mergers that expanded our products and services. There also exist several marketing opportunities to increase our presence to our members. One example includes Internet commerce services. Finally, the automatic identification system being developed for the warehouse must fully interoperate with member services.

BRIEF STATEMENT OF EXPECTED SOLUTION
We envision completely new and streamlined business processes that minimize the response time to member orders for products and services. An order shall not be considered fulfilled until it has been received by the member. The new system should provide for expanded club and member flexibility and adaptability of basic business products and services.

We envision a system that extends to the desktop computers of both employees and members, with appropriate shared services provided across the network, consistent with the ISS distributed architecture. This is consistent with strategic plans to retire the AS/400 central computer and replace it with servers.

ACTION (ISS Office Use Only)

- ☐ Feasibility assessment approved — Assigned to Sandra Shepherd
- ☒ Feasibility assessment waived — Approved Budget $ 450,000
- Start Date ASAP Deadline ASAP
- ☐ Request delayed — Backlogged until date: __________
- ☐ Request rejected — Reason: __________

Authorized Signatures:
Rebecca J. Todd
Chair, ISS Executive Steering Body

Galen Kirkhoff
Project Executive Sponsor

FORM ISS-100-RFSS (Last revised December, 1999)

The key deliverable of this task, the PRELIMINARY PROBLEM STATEMENT, consists of the problems, opportunities, and directives that were identified. The PROBLEM STATEMENTS are stored in the repository for later use in the project. Figure 5-8 is a sample document that summarizes problems, opportunities, and directives in terms of:

- *Urgency*—In what time frame must/should the problem be solved or the opportunity or directive be realized? A rating scale could be developed to consistently answer this question.
- *Visibility*—To what degree would a solution or new system be visible to customers and/or executive management? Again, a rating scale could be developed for the answers.
- *Benefits*—Approximately how much would a solution or new system increase annual revenues or reduce annual costs? This is often a guess, but if all participants are involved in that guess, it should prove sufficiently conservative.

Problem Statements

Project: Member services information system	Project manager: Sandra Shepherd
Created by: Sandra Shepherd	Last updated by: Robert Martinez
Date created: January 9, 2003	Date last updated: January 15, 2003

Brief Statements of Problem, Opportunity, or Directive	Urgency	Visibility	Annual Benefits	Priority or Rank	Proposed Solution
1. Order response time as measured from time of order receipt to time of customer delivery has increased to an average of 15 days.	ASAP	High	$175,000	2	New development
2. The recent acquisitions of Private Screenings Video Club and GameScreen will further stress the throughput requirements for the current system.	6 months	Med	75,000	2	New development
3. Currently, three different order entry systems service the audio, video, and game divisions. Each system is designed to interface with a different warehousing system; therefore, the intent to merge inventory into a single warehouse has been delayed.	6 months	Med	515,000	2	New development
4. There is a general lack of access to management and decision-making information. This will become exasperated by the acquisition of two additional order processing systems (from Private Screenings and GameScreen).	12 months	Low	15,000	3	After new system is developed, provide users with easy-to-learn and -use reporting tools.
5. There currently exist data inconsistencies in the member and order files.	3 months	High	35,000	1	Quick fix; then new development
6. The Private Screenings and GameScreen file systems are incompatible with the SoundStage equivalents. Business data problems include data inconsistencies and lack of input edit controls.	6 months	Med	Unknown	2	New development. Additional quantification of benefit might increase urgency.
7. There is an opportunity to open order systems to the Internet, but security and control are an issue.	12 months	Low	Unknown	4	Future version of newly developed system
8. The current order entry system is incompatible with the forthcoming automatic identification (bar-coding) system being developed for the warehouse.	3 months	High	65,000	1	Quick fix; then new development

FIGURE 5-8 Sample Problem Statements

- *Priority*—Based on the above answers, what are the consensus priorities for each problem, opportunity, or directive. If budget or schedule becomes a problem, these priorities will help to adjust project scope.
- *Possible solutions* (OPT)—At this early stage of the project, possible solutions are best expressed in simple terms such as (a) leave well enough alone, (b) use a quick fix, (c) make a simple to moderate enhancement of the existing system, (d) redesign the existing system, or (e) design a new system. The participants listed for this task are well suited to an appropriately high-level discussion of these options.

The PIECES framework that was introduced in Chapter 3 can be used as a framework for categorizing problems, opportunities, directives, and constraints. For example, Problem 1 in Figure 5-8 could be classified according to PIECES as P.B.—Performance, Response Times. (See Figure 3-4 in Chapter 3). Problem 4 in Figure 5-8 could be classified as I.A.2—Information, Outputs, Lack of necessary information.

The primary techniques used to complete this task include fact-finding and meetings with SYSTEM OWNERS. These techniques are taught in Chapter 6.

> Task 1.2—Negotiate Baseline Scope

Scope defines the boundary of the project—those aspects of the business that will and will not be included in the project. Scope can change during the project; however, the initial project plan must establish the preliminary or baseline scope. Then if the scope changes significantly, all parties involved will have a better appreciation for why the budget and schedule have also changed. This task can occur in parallel with the prior task.

Once again, a senior systems analyst or project manager usually leads this task. Most of the other participants are broadly classified as SYSTEM OWNERS. This includes the executive sponsor, managers of all organizational units that may be impacted by the system, and possibly information systems managers. SYSTEM USERS, SYSTEM DESIGNERS, and SYSTEM BUILDERS are not typically involved in this task.

As shown in Figure 5-6, this task uses the PRELIMINARY PROBLEM STATEMENT produced by the previous task. It should make sense that those problems, opportunities, and directives form the basis for defining scope. The STATEMENTS OF PROJECT SCOPE are added to the repository for later use. These statements are also formally documented as the task deliverable, PRELIMINARY PROBLEM STATEMENT WITH SCOPE.

Scope can be defined easily within the context of your information system building blocks. For example, a project's scope can be described in terms of:

- What types of DATA describe the system being studied? For example, a sales information system may require data about such things as CUSTOMERS, ORDERS, PRODUCTS, and SALES REPRESENTATIVES.
- What business PROCESSES are included in the system being studied? For example, a sales information system may include business processes for CATALOG MANAGEMENT, CUSTOMER MANAGEMENT, ORDER ENTRY, ORDER FULFILLMENT, ORDER MANAGEMENT, and CUSTOMER RELATIONSHIP MANAGEMENT.
- How must the system INTERFACE with users, locations, and other systems? For example, potential interfaces for a sales information system might include CUSTOMERS, SALES REPRESENTATIVES, SALES CLERKS AND MANAGERS, REGIONAL SALES OFFICES, and the ACCOUNTS RECEIVABLE and INVENTORY CONTROL INFORMATION SYSTEMS.

Notice that each statement of scope can be described as a simple list. We don't necessarily "define" the items in the list. Nor are we very concerned with precise requirements analysis. And we definitely are not concerned with any time-consuming steps such as modeling or prototyping.

Once again, the primary techniques used to complete this task are fact-finding and meetings. Many analysts prefer to combine this task with both the previous and the next tasks and accomplish them within a single meeting.

> Task 1.3—Assess Baseline Project Worthiness

This is where we answer the question, "Is this project worth looking at?" At this early stage of the project, the question may actually boil down to a "best guess": Will solving the problems, exploiting the opportunities, or fulfilling the directives return enough value to offset the costs that we will incur to develop this system? It is impossible to do a thorough feasibility analysis based on the limited facts we've collected to date.

Again, a senior systems analyst or project manager usually leads this task. But the SYSTEM OWNERS, inclusive of the executive sponsor, the business unit managers, and the information systems managers, should make the decision.

As shown in Figure 5-6, the completed PRELIMINARY PROBLEM STATEMENT WITH SCOPE triggers the task. This provides the level of information required for this preliminary assessment of worth. There is no physical deliverable other than the GO OR NO-GO DECISION. There are actually several alternative decisions. The project can be approved or canceled, and project scope can be renegotiated (increased or decreased!). Obviously, the remaining tasks in the preliminary investigation phase are necessary only if the project has been deemed worthy and approved to continue.

> Task 1.4—Develop Baseline Schedule and Budget

If the project has been deemed worthy to continue, we can now plan the project in depth. The initial project plan should consist of at least the following:

- A preliminary master plan that includes schedule and resource assignments for the entire project. This plan will be updated at the end of each phase of the project. It is sometimes called a *baseline plan.*
- A detailed plan and schedule for completing the next phase of the project (the problem analysis phase).

The task is the responsibility of the *project manager.* Most project managers find it useful to include as much of the project team, including SYSTEM OWNERS, USERS, DESIGNERS, and BUILDERS, as possible. Chapter 4 coined the term *joint project planning* to describe the team approach to building a project plan.

As shown in Figure 5-6, this task is triggered by the GO OR NO-GO DECISION to continue the project. This decision represents a consensus agreement on the project's scope, problems, opportunities, directives, and worthiness. (This "worthiness" must still be presented and approved.) The PROBLEM STATEMENTS WITH SCOPE are the key input (from the repository). The deliverable of this task is the BASELINE PROJECT PLAN AND SCHEDULE. The STATEMENT OF WORK (see Chapter 4) and PROJECT SCHEDULE AND RESOURCE ASSIGNMENTS are also added to the repository for continuous monitoring and, as appropriate, updating. The schedule and resources are typically maintained in the repository as a project management software file.

The techniques used to create a project plan were covered in depth in Chapter 4. Today, these techniques are supported by project management software such as Microsoft *Project.* Chapter 4 also discussed the detailed steps for completing the plan.

> Task 1.5—Communicate the Project Plan

In most organizations, there are more potential projects than resources to staff and fund those projects. Unless our project has been predetermined to be of the highest priority (by some sort of prior tactical or strategic planning process), then it must be presented and defended to a **steering body** for approval. Most organizations use a steering body to approve and monitor projects and progress. The majority of any steering body should consist of non–information systems professionals or managers. Many organizations designate vice presidents to serve on a steering body. Other

steering body a committee of executive business and system managers that studies and prioritizes competing project proposals to determine which projects will return the most value to the organization and thus should be approved for continued systems development. Also called a *steering committee.*

organizations assign the direct reports of vice presidents to the steering body. And some organizations utilize two steering bodies, one for vice presidents and one for their direct reports. Information systems managers serve on the steering body only to answer questions and to communicate priorities back to developers and project managers.

Regardless of whether or not a project requires steering committee approval, it is equally important to formally launch the project and communicate the project, goals, and schedule to the entire business community. Opening the lines of communication is an important capstone to the preliminary investigation. For this reason, we advocate the "best practices" of conducting a *project kickoff event* and creating an *intranet project Web site.* The project kickoff meeting is open to the entire business community, not just the business units affected and the project team. The intranet project Web site establishes a community portal to all nonsensitive news and documentation concerning the project.

Ideally, the executive sponsor should jointly facilitate the task with the chosen project manager. The visibility of the executive sponsor establishes instant credibility and priority to all who participate in the kickoff meeting. Other kickoff meeting participants should include the entire project team, including assigned SYSTEM OWNERS, USERS, ANALYSTS, DESIGNERS, and BUILDERS. Ideally, the kickoff meeting should be open to any and all interested staff from the business community. This builds community awareness and consensus while reducing both the volume and the consequences of rumor and misinformation. For the intranet component, a Webmaster or Web author should be assigned to the project team.

As shown in Figure 5-6, this task is triggered by the completion of the BASELINE PROJECT PLAN AND SCHEDULE. The PROBLEM STATEMENTS AND SCOPE are available from the repository. The deliverable is the PROJECT CHARTER. The project charter is usually a document. It includes various elements that define the project in terms of participants, problems, opportunities, and directives; scope; methodology; statement of work to be completed; deliverables; quality standards; schedule; and budget. The project charter should be added to the project Web site for all to see. Elements of the project charter may also be reformatted as slides and handouts (using software such as Microsoft *PowerPoint*) for inclusion in the project kickoff event.

Effective interpersonal and communications skills are the keys to this task. These include principles of persuasion, selling change, business writing, and public speaking.

This concludes our discussion of the scope definition phase. The participants in the scope definition phase might decide the project is not worth proposing. It is also possible the steering body may decide that other projects are *more* important. Or the executive sponsor might not endorse the project. In each of these instances, the project is terminated. Little time and effort have been expended. On the other hand, with the blessing of all the system owners and the steering committee, the project can now proceed to the problem analysis phase.

The Problem Analysis Phase

There is an old saying, "Don't try to fix it unless you understand it." That statement aptly describes the *problem analysis phase* of systems analysis. There is always a current or existing system, regardless of the degree to which it is automated with information technology. The problem analysis phase provides the analyst with a more thorough understanding of the problems, opportunities, and/or directives that triggered the project. The problem analysis phase answers the questions, "Are the problems really worth solving?" and "Is a new system really worth building?" In other methodologies, the problem analysis phase may be known as the *study phase, study of the current system, detailed investigation phase,* or *feasibility analysis phase.*

Can you ever skip the problem analysis phase? Rarely! You almost always need some level of understanding of the current system. But there may be reasons to accelerate the problem analysis phase. First, if the project was triggered by a strategic or tactical plan, the worthiness of the project is probably not in doubt—the problem analysis phase would be reduced to understanding the current system, not analyzing it. Second, a project may be initiated by a directive (such as compliance with a governmental directive and deadline). Again, in this case project worthiness is not in doubt. Finally, some methodologies and organizations deliberately consolidate the problem analysis and requirements analysis phases to accelerate systems analysis.

The goal of the problem analysis phase is to study and understand the problem domain well enough to thoroughly analyze its problems, opportunities, and constraints. Some methodologies encourage a very detailed understanding of the current system and document that system in painstaking detail using system models such as data flow diagrams. Today, except when business processes must be redesigned, the effort required and the value added by such detailed modeling is questioned and usually bypassed. Thus, the current version of our hypothetical *FAST* methodology encourages only enough system modeling to refine our understanding of project scope and problem statement, and to define a common vocabulary for the system.

The context for the problem analysis phase is shaded in Figure 5-9. Notice that the problem analysis phase is concerned primarily with both the SYSTEM OWNERS' and the SYSTEM USERS' views of the existing system. Notice that we build on the lists created in the preliminary investigation phase to analyze the KNOWLEDGE, PROCESS, and COMMUNICATIONS building blocks of the existing system. Also notice that we imply minimal system modeling. We may still use the PIECES framework to analyze each building block for problems, causes, and effects.

Figure 5-10 is the task diagram for the problem analysis phase. The final phase deliverable and milestone is producing SYSTEM IMPROVEMENT OBJECTIVES that address problems, opportunities, and directives. Depending on the size of the system, its complexity, and the degree to which project worthiness is already known, the illustrated tasks may consume one to six weeks. Most of these tasks can be accelerated by JRP-like sessions. The problem analysis phase typically includes the following tasks:

2.1 Understand the problem domain.
2.2 Analyze problems and opportunities.
2.3 Analyze business processes.
2.4 Establish system improvement objectives.
2.5 Update or refine the project plan.
2.6 Communicate findings and recommendations.

Let's now examine each of these tasks in greater detail.

> Task 2.1—Understand the Problem Domain

During the problem analysis phase, the *team* initially attempts to learn about the current system. Each SYSTEM OWNER, USER, and ANALYST brings a different level of understanding to the system—different detail, different vocabulary, different perceptions, and different opinions. A well-conducted study can prove revealing to all parties, including the system's own management and users. It is important to study and *understand the problem domain*, that domain in which the business problems, opportunities, directives, and constraints exist.

This task will be led by the project manager but facilitated by the lead systems analyst. It is not uncommon for one individual to play both roles (as Sandra does in the SoundStage case). Other SYSTEMS ANALYSTS may also be involved since they conduct interviews, scribe for meetings, and document findings. A comprehensive study should include representative SYSTEM OWNERS and USERS from all business units that will be supported or impacted by the system and project. It is extraordinarily important that enough users be included to encompass the full scope of the

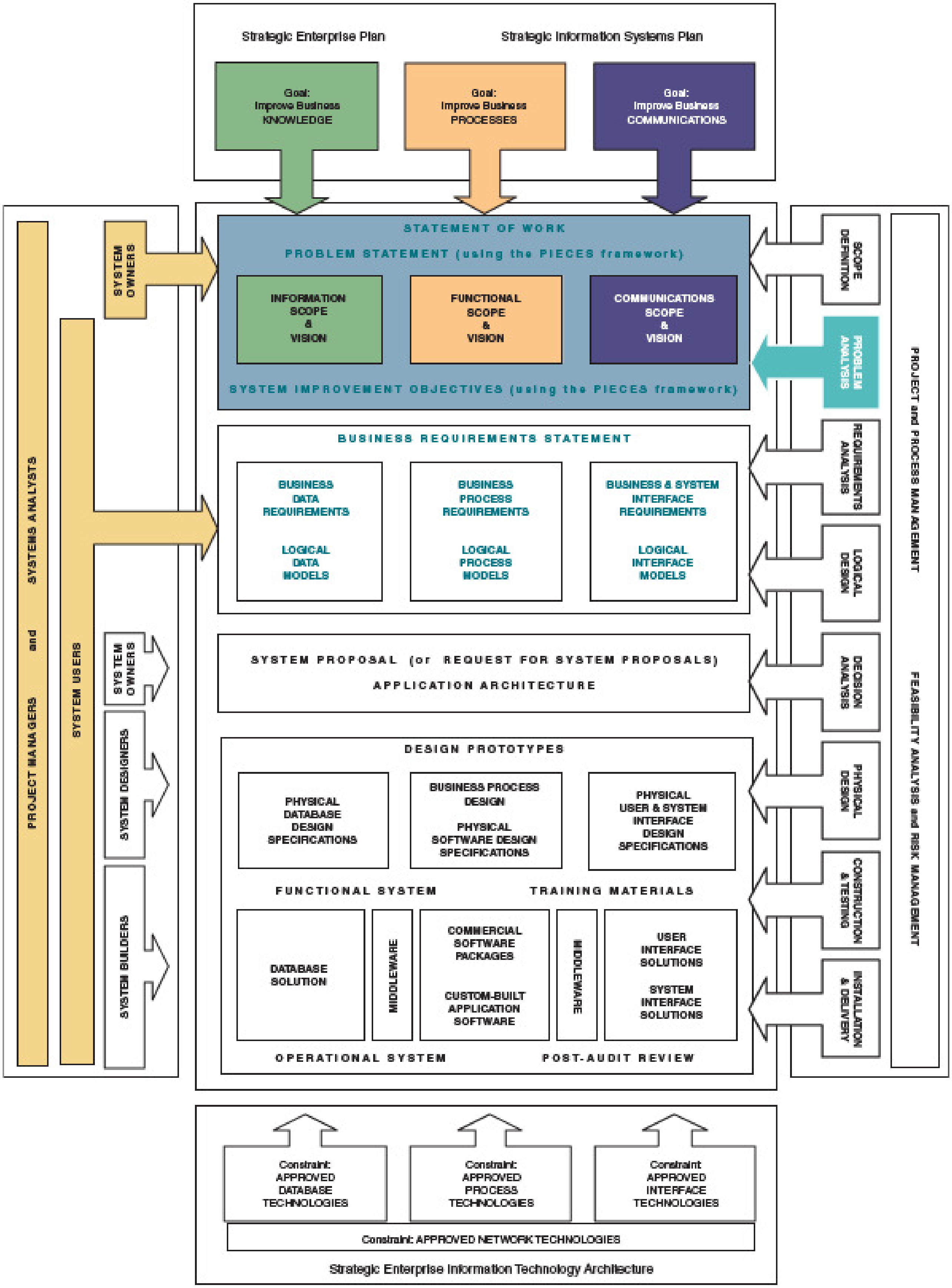


FIGURE 5-9 The Context of the Problem Analysis Phase of Systems Analysis

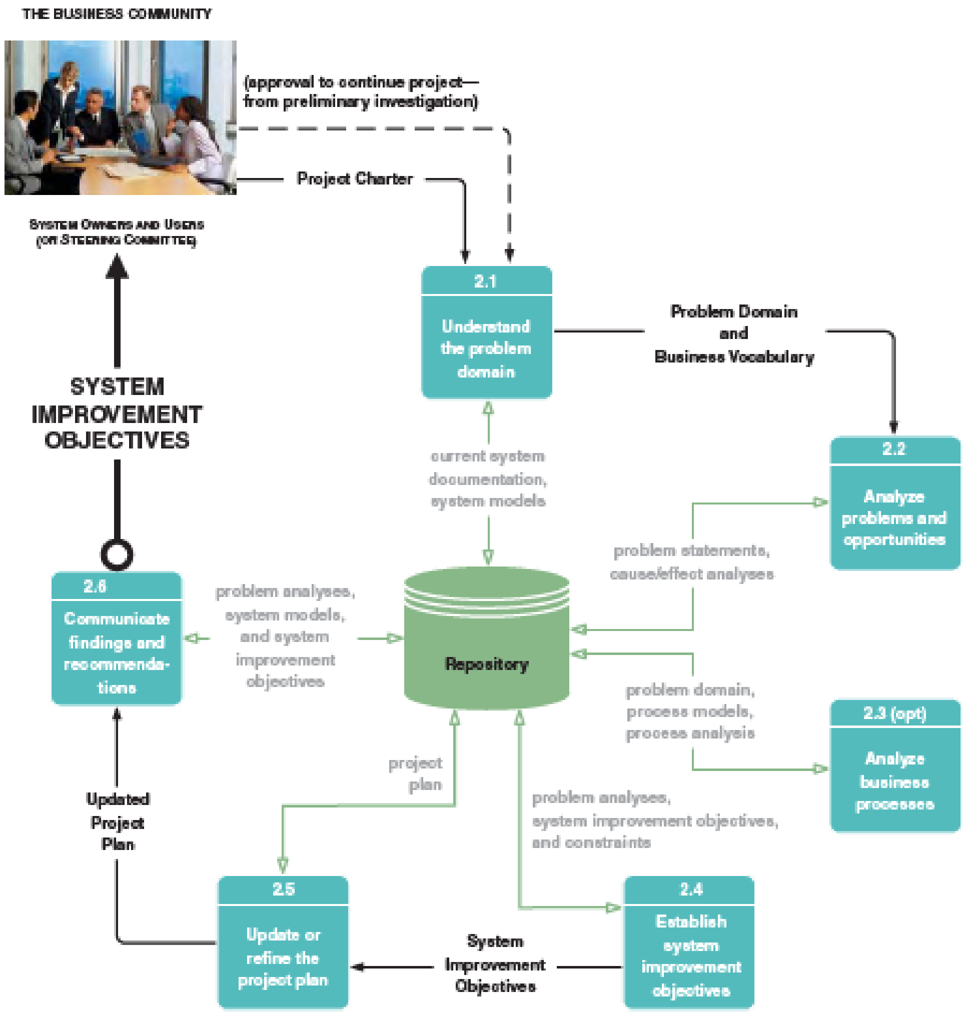


FIGURE 5-10
Tasks for the Problem Analysis Phase of Systems Analysis

system being studied. In some organizations, one or more experienced users are "loaned" to the project full-time as *business analysts;* however, it is rare that any one user can fully represent the interests of all users. Business analysts can, however, serve as facilitators to get the right people involved and sustain effective communication back to the business units and management. SYSTEM DESIGNERS and BUILDERS are rarely involved in this task unless they are interviewed to determine any technical limitations of the current system.

In Figure 5-10, this task is triggered by APPROVAL TO CONTINUE THE PROJECT—from the scope definition phase. (The dashed line indicates this approval is an event or trigger, not a data or information flow.) The approval comes from the SYSTEM OWNERS or steering committee. The key informational input is the PROJECT CHARTER and any CURRENT SYSTEM DOCUMENTATION that may exist in the repository and program libraries for the current system. Current system documentation doesn't always exist. And when it does exist, it must be carefully checked for currency—most such documentation is notoriously out of date because analysts and programmers are not always diligent about updating that documentation as changes occur throughout the lifetime of a system.

The deliverables of this task are an understanding of the PROBLEM DOMAIN AND BUSINESS VOCABULARY. Your understanding of the existing problem domain should be documented so that it can be verified that you truly understand it. There are several ways to document the problem domain. Certainly, drawing SYSTEM MODELS of the current system can help, but they can lead to a phenomenon called "analysis paralysis," in which the desire to produce perfect models becomes counterproductive to the schedule. Another approach might be to use your information system building blocks as a framework for listing and defining the system domain:

- Knowledge—List all the "things" about which the system currently stores data (in files, databases, forms, etc.). Define each thing in business terms. For example, "An order is a business transaction in which a customer requests to purchase products."

  Additionally, we could list all the reports produced by the current system and describe their purpose or use. For example, "The open orders report describes all orders that have not been filled within one week of their approval to be filled. The report is used to initiate customer relationship management through personal contact."
- Processes—Define each business event for which a business response (process) is currently implemented. For example, "A customer places a new order," or "A customer requests changes to a previously placed order," or "A customer cancels an order."
- Communications—Define all the locations that the current system serves and all of the users at each of those locations. For example, "The system is currently used at regional sales offices in San Diego, Dallas, St. Louis, Indianapolis, Atlanta, and Manhattan. Each regional sales office has a sales manager, assistant sales manager, administrative assistant, and 5 to 10 sales clerks, all of whom use the current system. Each region is also home to 5 to 30 sales representatives who are on the road most days but who upload orders and other transactions each evening."

  Another facet of interfaces is system interfaces—that is, interfaces that exist between the current information system and other information systems and computer applications. These can be quickly listed and described by the information systems staff.

Ultimately, the organization's systems development methodology and project plan will determine what types and level of documentation are expected.

The business vocabulary deliverable is all too often shortchanged. Understanding the business vocabulary of the system is an excellent way of understanding the system itself. It bridges the communication gap that often exists or develops between business and technology experts.

If you elect to draw system models during this task, we suggest that "if you want to learn *anything,* you must not try to learn *everything*—at least not all in this task." To avoid analysis paralysis, we suggest that the following system models may be appropriate:

- Knowledge—A one-page data model is very useful for establishing business vocabulary and rules. Data modeling is taught in Chapter 8.
- Processes—Today, it is widely accepted that a one- or two-page functional decomposition diagram should prove sufficient to get a feel for the current system processing. Decomposition modeling is taught in Chapter 9.
- Communications—A one-page context diagram or use-case diagrams are very useful for illustrating the system's inputs and outputs with other organizations, business units, and systems. Context diagrams are discussed below. Use case diagrams are taught in Chapter 7.

Several other techniques and skills are useful for developing an understanding of an existing system. Obviously, fact-finding techniques (taught in the next chapter) are critical to learning about any existing system. Also, joint requirements planning, or JRP, techniques (also taught in the next chapter) can accelerate this task. Finally, the ability to clearly communicate back to users what you've learned about a system is equally crucial.

Context Diagram The purpose of a context diagram is to analyze how the system interacts with the world around it and to specify in general terms the system inputs and outputs. Context diagrams can be drawn in various ways. Chapter 9 presents the traditional format, which was done as the first step in drawing data flow diagrams.

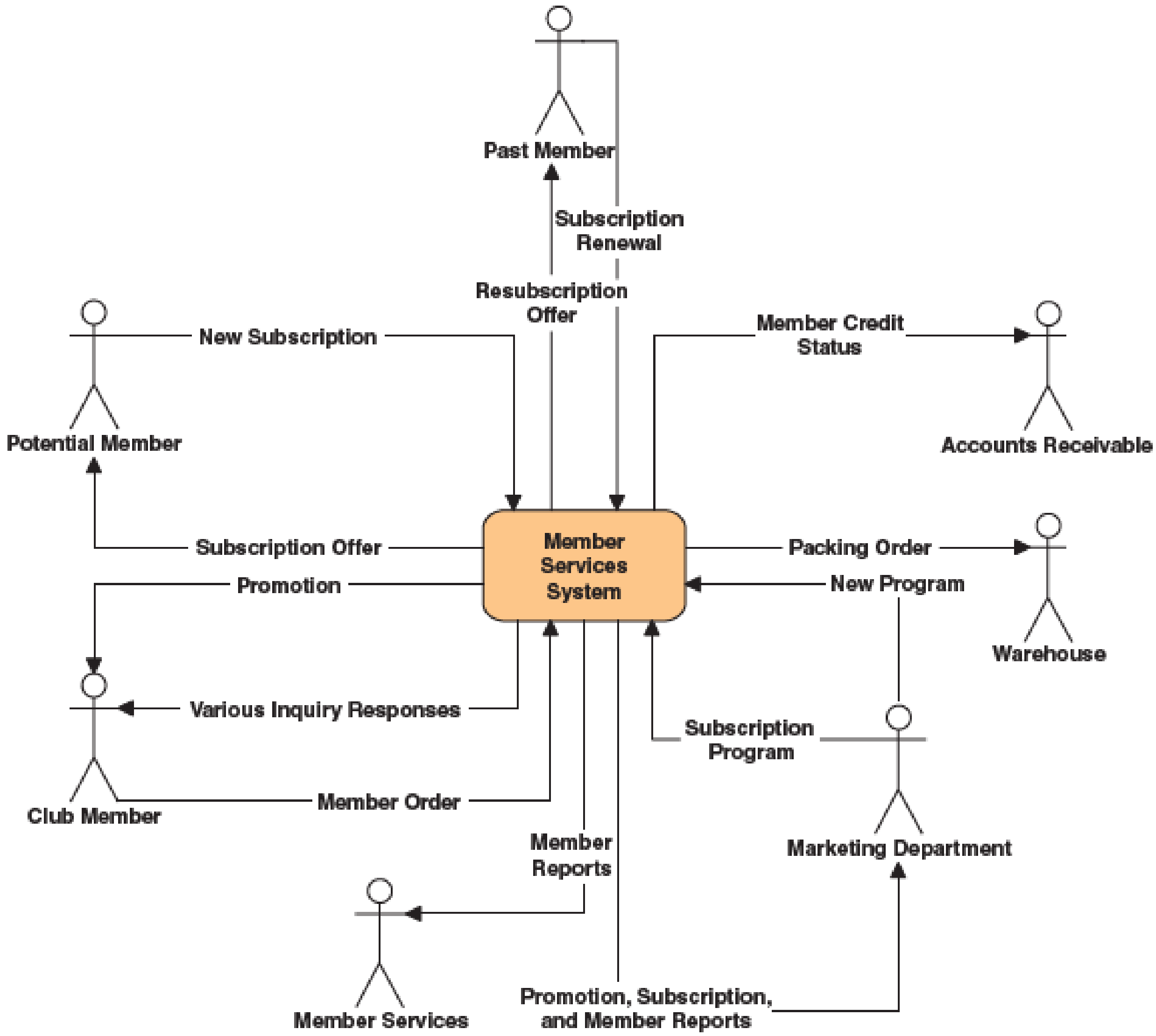


FIGURE 5-11 Context Diagram

Chapter 7 shows a different format for a context diagram. The context diagram shown in Figure 5-11 employs a hybrid approach. It employs use case symbols as use cases are becoming a generally accepted tool of the requirements analysis phase.

The system itself is shown as a "black box" in the middle of the diagram. We are not yet ready to look inside the box. For now we just want to see how everyone will use the box. The stick figures around the outside of the diagram are the persons, organizations, and other information systems that will interact with the system. In use cases, these are called actors, and we can call them that here. In traditional data flow diagrams, they are called external agents. In Chapters 7 and 9 you will learn that once you look inside the system box, other things such as time or devices like sensors can also be actors or external agents. But for a context diagram they are rarely shown.

The lines indicate the inputs (arrows pointing to the system) provided by actors to the system and the outputs (arrows pointing to the actors) created by the system. Each input and output is identified with a noun phrase that describes it.

To build a context diagram ask the users what business transactions the system must respond to; these are the inputs. Also ask the users what reports, notifications, and other outputs must be produced by the system. A system can have many reports that can quickly clutter the diagram; consolidate them as needed to keep the diagram readable. During other phases in the process they will be analyzed separately.

We certainly couldn't build an information system from a context diagram. But it is a solid first step. From this simple diagram we know what inputs the system must respond to and what outputs it must produce. In other words, it helps us understand the problem domain. We will see in Chapter 7 how to detect use cases from a context diagram. That will be the first step in cracking open the "black box." We are following the principles for systems development presented in Chapter 2: "use a problem-solving approach" and "divide and conquer."

> Task 2.2—Analyze Problems and Opportunities

In addition to learning about the current system, the project team must work with system owners and system users to *analyze problems and opportunities.* You might be asking, "Weren't problems and opportunities identified earlier, in the preliminary investigation phase?" Yes, they were. But those initial problems may be only symptoms of other problems, perhaps problems not as well known or understood by the users. Besides, we haven't yet really analyzed any of those problems in the classic sense.

True problem analysis is a difficult skill to master, especially for inexperienced systems analysts. Experience suggests that most new systems analysts (and many system owners and users) try to solve problems without truly analyzing them. They might state a problem like this: "We need to . . ." or "We want to . . ." In doing so, they are stating the problem in terms of a solution. More effective problem solvers have learned to truly analyze the problem before stating any possible solution. They analyze each perceived problem for **causes and effects.** In practice, an effect may actually be a symptom of a different, more deeply rooted or basic problem. That problem must also be analyzed for causes and effects, and so on until such a time as the causes and effects do not yield symptoms of other problems. Cause-and-effect analysis leads to true understanding of problems and can lead to not-so-obvious but more creative and valuable solutions.

cause-and-effect analysis a technique in which problems are studied to determine their causes and effects.

SYSTEMS ANALYSTS facilitate this task; however, all SYSTEMS OWNERS and USERS should actively participate in the process of cause-and-effect analysis. They are the problem domain experts. SYSTEM DESIGNERS and BUILDERS are not usually involved in this process unless they are called on to analyze technical problems that may exist in the current system.

As shown in Figure 5-10, the team's understanding of the SYSTEM DOMAIN AND BUSINESS VOCABULARY triggers this task. This understanding of the problem domain is crucial because the team members should not attempt to analyze problems unless they understand the domain in which those problems occur. The other informational input to this task is the initial PROBLEM STATEMENTS (from the scope definition phase). The deliverables of this task are the updated PROBLEM STATEMENTS and the CAUSE-EFFECT ANALYSIS for each problem and opportunity. Figure 5-12 illustrates one way to document a cause-and-effect analysis.

Once again, fact-finding and JRP techniques are crucial to this task. These techniques, as well as cause-and-effect analysis, are taught in the next chapter.

> Task 2.3—Analyze Business Processes

This task is appropriate only to *business process redesign (BPR)* projects or system development projects that build on or require significant business process redesign. In such a project, the team is asked to examine its business processes in much greater detail to measure the value added or subtracted by each process as it relates to the total organization. Business process analysis can be politically charged. System owners and users alike can become very defensive about their existing business

PROBLEMS, OPPORTUNITIES, OBJECTIVES, AND CONSTRAINTS MATRIX

Project:	Member Services Information System	Project Manager:	Sandra Shepherd
Created by:	Robert Martinez	Last Updated by:	Robert Martinez
Date Created:	January 21, 2003	Date Last Updated:	January 31, 2003

CAUSE-AND-EFFECT ANALYSIS		SYSTEM IMPROVEMENT OBJECTIVES	
Problem or Opportunity	**Causes and Effects**	**System Objective**	**System Constraint**
1. Order response time is unacceptable.	1. Throughput has increased while number of order clerks was downsized. Time to process a single order has remained relatively constant. 2. System is too keyboard-dependent. Many of the same values are keyed for most orders. Net result is (with the current system) each order takes longer to process than is ideal. 3. Data editing is performed by the AS/400. As that computer has approached its capacity, order edit responses have slowed. Because order clerks are trying to work faster to keep up with the volume, the number of errors has increased. 4. Warehouse picking tickets for orders were never designed to maximize the efficiency of order fillers. As warehouse operations grew, order filling delays were inevitable.	1. Decrease the time required to process a single order by 30%. 2. Eliminate keyboard data entry for as much as 50% of all orders. 3. For remaining orders, reduce as many key-strokes as possible by replacing keystrokes with point-and-click objects on the computer display screen. 4. Move data editing from a shared computer to the desktop. 5. Replace existing picking tickets with a paperless communication system between member services and the warehouse.	1. There will be no increase in the order processing workforce. 2. Any system developed must be compatible with the existing Windows 95 desktop standard. 3. New system must be compatible with the already approved automatic identification system (for bar coding).

FIGURE 5-12 A Sample Cause-and-Effect Analysis

processes. The analysts involved must keep the focus on the processes, not the people who perform them, and constantly remind everyone that the goal is to identify opportunities for fundamental business change that will benefit the business and everyone in the business.

One or more systems analysts or business analysts facilitate the task. Ideally, the ANALYSTS should be experienced, trained, or certified in BPR methods. The only other participants should be appropriate SYSTEM OWNERS and USERS. Business process analysis should avoid any temptation to focus on information technology solutions until well after the business processes have been redesigned for maximum efficiency. Some analysts find it useful to assume the existence of "perfect people" and "perfect technology" that can make anything "possible." They ask, "If the world were perfect, would we need this process?"

As depicted in Figure 5-10, a business process analysis task is dependent only on some PROBLEM DOMAIN knowledge (from Task 2.1). The deliverables of this task are business "as is" PROCESS MODELS and PROCESS ANALYSES. The process models can look very much like data flow diagrams (Figure 5-2) except they are significantly annotated to show (1) the volume of data flowing through the processes, (2) the response times of each process, and (3) any delays or bottlenecks that occur in the system. The process analysis data provides additional information such as (a) the cost of each process, (b) the value added by each process, and (c) the consequences of eliminating or streamlining the process. Based on the as-is models and their analysis, the team develops "to be" models that redesign the business processes to eliminate redundancy and bureaucracy and increase efficiency and service.

Several techniques are applicable to this task. Once again, fact-finding techniques and facilitated team meetings (Chapter 6) are invaluable. Also, process modeling techniques (Chapter 9) are critical to BPR success.

> Task 2.4—Establish System Improvement Objectives

Given our understanding of the current system's scope, problems, and opportunities, we can now *establish system improvement objectives*. The purpose of this task is to establish the criteria against which any improvements to the system will be measured and to identify any constraints that may limit flexibility in achieving those improvements. The criteria for success should be measured in terms of **objectives.** Objectives represent the first attempt to establish expectations for any new system. In addition to identifying objectives, we must also identify any known constraints. **Constraints** place limitations or delimitations on achieving objectives. Deadlines, budgets, and required technologies are examples of constraints.

objective a measure of success. It is something that you expect to achieve, if given sufficient resources.

constraint something that will limit your flexibility in defining a solution to your objectives. Essentially, constraints cannot be changed.

The SYSTEMS ANALYSTS facilitate this task. Other participants include the same SYSTEM OWNERS and USERS who have participated in other tasks in this problem analysis phase. Again, we are not yet concerned with technology; therefore, SYSTEM DESIGNERS and BUILDERS are not involved in this task.

This task is triggered by the PROBLEM ANALYSES completed in Tasks 2.2 and 2.3. For each verified and significant problem, the analysts and users should define specific SYSTEM IMPROVEMENT OBJECTIVES. They should also identify any CONSTRAINTS that may limit or prevent them from achieving the system improvement objectives.

System improvement objectives should be precise, measurable statements of business performance that define the expectations for the new system. Some examples are:

> Reduce the number of uncollectible customer accounts by 50 percent within the next year.
>
> Increase by 25 percent the number of loan applications that can be processed during an eight-hour shift.
>
> Decrease by 50 percent the time required to reschedule a production lot when a workstation malfunctions.

The following is an example of a poor objective:

> Create a delinquent accounts report.

This is a poor objective because it states only a requirement, not an actual objective. Now, let's reword that objective:

> Reduce credit losses by 20 percent through earlier identification of delinquent accounts.

This gives us more flexibility. Yes, the delinquent accounts report would work. But a customer delinquency inquiry might provide an even better way to achieve the same objective.

System improvement objectives may be tempered by identifiable constraints. Constraints fall into four categories, as listed below (with examples):

- *Schedule:* The new system must be operational by April 15.
- *Cost:* The new system cannot cost more than $350,000.
- *Technology:* The new system must be online, or all new systems must use the DB2 database management system.
- *Policy:* The new system must use double-declining-balance inventory techniques.

The last two columns of Figure 5-12 document typical system improvement objectives and constraints.

> Task 2.5—Update or Refine the Project Plan

Recall that project scope is a moving target. Based on our baseline schedule and budget from the scope definition phase, scope may have grown or diminished in size and complexity. (Growth is much more common!) Now that we're approaching the completion of the problem analysis phase, we should reevaluate project scope and *update or refine the project plan* accordingly.

The project manager, in conjunction with SYSTEM OWNERS and the entire project team, facilitates this task. The SYSTEMS ANALYSTS and SYSTEM OWNERS are the key individuals in this task. The analysts and owners should consider the possibility that not all objectives may be met by the new system. Why? The new system may be larger than expected, and they may have to reduce the scope to meet a deadline. In this case the system owner will rank the objectives in order of importance. Then, if the scope must be reduced, the higher-priority objectives will tell the analyst what's most important.

As shown in Figure 5-10, this task is triggered by completion of the SYSTEM IMPROVEMENT OBJECTIVES. The initial PROJECT PLAN is another key input, and the UPDATED PROJECT PLAN is the key output. The updated plan should now include a detailed plan for the requirements analysis phase that should follow. The techniques and steps for updating the project plan were taught in Chapter 4, "Project Management."

> Task 2.6—Communicate Findings and Recommendations

As with the scope definition phase, the problem analysis phase concludes with a communication task. We must *communicate findings and recommendations* to the business community. The project manager and executive sponsor should jointly facilitate this task. Other meeting participants should include the entire project team, including assigned SYSTEM OWNERS, USERS, ANALYSTS, DESIGNERS, and BUILDERS. And, as usual, the meeting should be open to any and all interested staff from the business community. Also, if an intranet Web site was established for the project, it should have been maintained throughout the problem analysis phase to ensure continuous communication of project progress.

This task is triggered by the completion of the UPDATED PROJECT PLAN. Informational inputs include the PROBLEM ANALYSES, any SYSTEM MODELS, the SYSTEM IMPROVEMENT OBJECTIVES, and any other documentation that was produced during the problem analysis phase. Appropriate elements are combined into the SYSTEM IMPROVEMENT OBJECTIVES, the major deliverable of the problem analysis phase. The format may be a report, a verbal presentation, or an inspection by an auditor or peer group (called a *walkthrough*). An outline for a written report is shown in Figure 5-13.

Interpersonal and communications skills are essential to this task. Systems analysts should be able to write a formal business report and make a business presentation without getting into technical issues or alternatives.

FIGURE 5-13 An Outline for a System Improvement Objectives and Recommendations Report

Analysis of the Current _______________ System

I. Executive summary (approximately 2 pages)
   A. Summary of recommendation
   B. Summary of problems, opportunities, and directives
   C. Brief statement of system improvement objectives
   D. Brief explanation of report contents
II. Background information (approximately 2 pages)
   A. List of interviews and facilitated group meetings conducted
   B. List of other sources of information that were exploited
   C. Description of analytical techniques used
III. Overview of the current system (approximately 5 pages)
   A. Strategic implications (if the project is part of or impacts an existing information systems strategic plan)
   B. Models of the current system
      1. Interface model (showing project scope)
      2. Data model (showing project scope)
      3. Geographic models (showing project scope)
      4. Process model (showing functional decomposition only)
IV. Analysis of the current system (approximately 5–10 pages)
   A. Performance problems, opportunities, and cause-effect analysis
   B. Information problems, opportunities, and cause-effect analysis
   C. Economic problems, opportunities, and cause-effect analysis
   D. Control problems, opportunities, and cause-effect analysis
   E. Efficiency problems, opportunities, and cause-effect analysis
   F. Service problems, opportunities, and cause-effect analysis
V. Detailed recommendations (approximately 5–10 pages)
   A. System improvement objectives and priorities
   B. Constraints
   C. Project plan
      1. Scope reassessment and refinement
      2. Revised master plan
      3. Detailed plan for the definition phase
VI. Appendixes
   A. Any detailed system models
   B. Other documents as appropriate

This concludes the problem analysis phase. One of the following decisions must be made after the conclusion of this phase:

- Authorize the project to continue, as is, to the requirements analysis phase.
- Adjust the scope, cost, and/or schedule for the project and then continue to the requirements analysis phase.
- Cancel the project due to (1) lack of resources to further develop the system, (2) realization that the problems and opportunities are simply not as important as anticipated, or (3) realization that the benefits of the new system are not likely to exceed the costs.

With some level of approval from the SYSTEM OWNERS, the project can now proceed to the requirements analysis phase.

The Requirements Analysis Phase

Many inexperienced analysts make a critical mistake after completing the problem analysis phase. The temptation at that point is to begin looking at alternative solutions, particularly technical solutions. One of the most frequently cited errors in new information systems is illustrated in the statement, "Sure the system works, and it is technically impressive, but it just doesn't do what we needed it to do." The *requirements analysis phase* defines the business requirements for a new system.

Did you catch the key word in the quoted sentence? It is "what," not "how"! Analysts are frequently so preoccupied with the *technical* solution that they inadequately define the *business* requirements for that solution. The requirements analysis phase answers the question, "What do the users need and want from a new system?" The requirements analysis phase is critical to the success of any new information system. In different methodologies the requirements analysis phase might be called the *definition phase* or *logical design phase.*

Can you ever skip the requirements analysis phase? Absolutely not! New systems will always be evaluated, first and foremost, on whether or not they fulfill business objectives and requirements, regardless of how impressive or complex the technological solution might be!

It should be acknowledged that some methodologies integrate the problem analysis and requirements analysis phases into a single phase.

Once again, your information systems building blocks (Figure 5-14) can serve as a useful framework for documenting the information systems requirements. Notice that we are still concerned with the SYSTEM USERS' perspectives. Requirements can be defined in terms of the PIECES framework or in terms of the types of data, processes, and interfaces that must be included in the system.

Figure 5-15 illustrates the typical tasks of the requirements analysis phase. The final phase deliverable and milestone is producing a BUSINESS REQUIREMENTS STATEMENT that will fulfill the system improvement objectives identified in the previous phase. One of the first things you may notice in this task diagram is that most of the tasks are not as sequential as those in previous task diagrams. Instead, many of these tasks occur in parallel as the team works toward the goal of completing the requirements statement. The requirements analysis phase typically includes the following tasks:

3.1 Identify and express system requirements.
3.2 Prioritize system requirements.
3.3 Update or refine the project plan.
3.4 Communicate the requirements statement.

Let's now examine each of these tasks in greater detail.

> Task 3.1—Identify and Express System Requirements

The initial task of the requirements analysis phase is to *identify and express requirements.* While this may seem to be an easy or trivial task, it is often the source of many errors, omissions, and conflicts. The foundation for this task was established in the problem analysis phase when we identified system improvement objectives. Minimally, this task translates those objectives into an outline of **functional** and **nonfunctional requirements** that will be needed to meet the objectives. Functional requirements are frequently identified in terms of inputs, outputs, processes, and stored data that are needed to satisfy the system improvement objectives. Examples of nonfunctional requirements include performance (throughput and response time); ease of learning and use; budgets, costs, and cost savings; timetables and deadlines; documentation and training needs; quality management; and security and internal auditing controls.

functional requirement a description of activities and services a system must provide.

nonfunctional requirement a description of other features, characteristics, and constraints that define a satisfactory system.

Rarely will this definition task identify *all* the functional or nonfunctional business requirements. But the outline will frame your thinking as you proceed to later

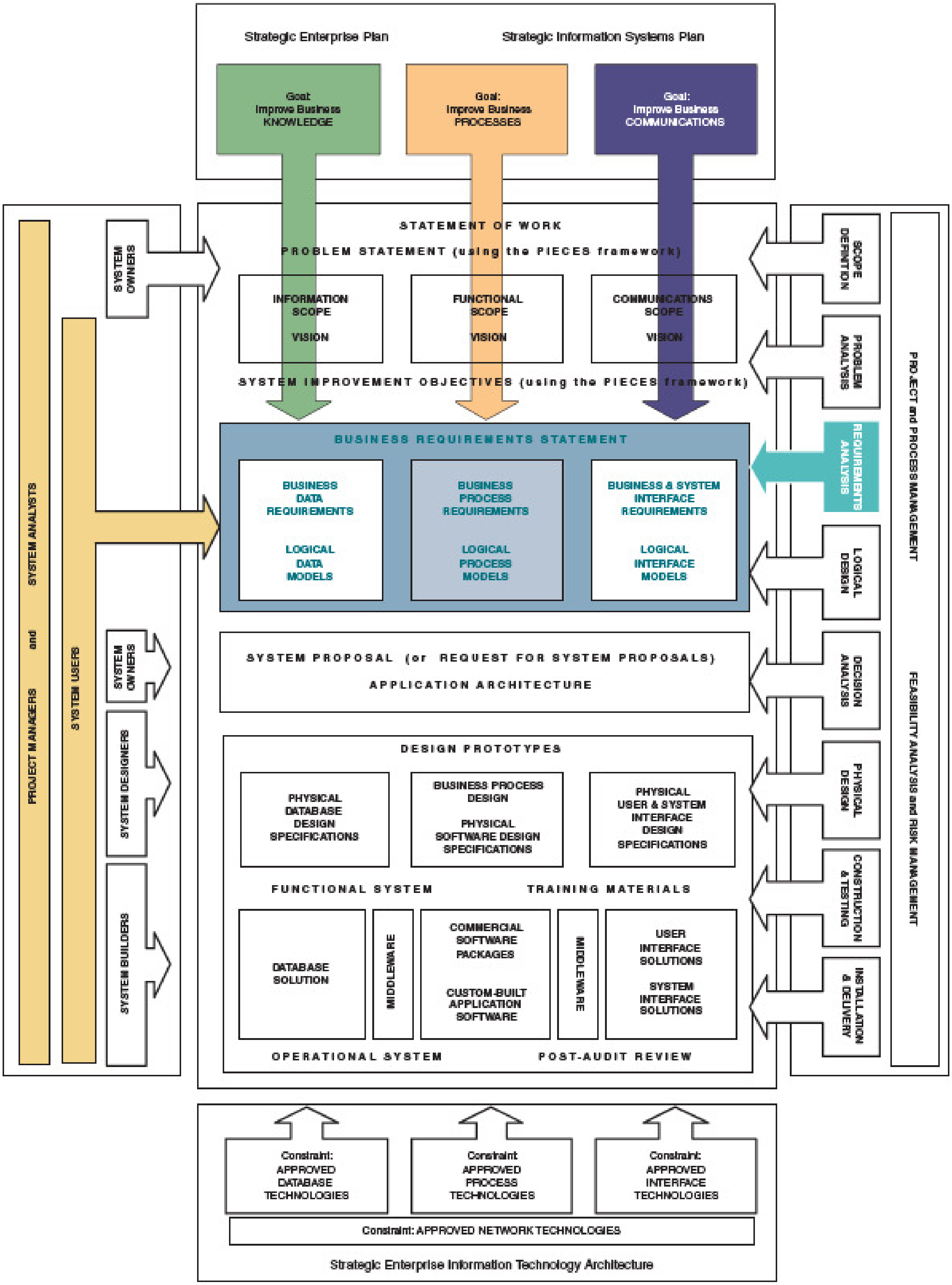


FIGURE 5-14 The Context of the Requirements Analysis Phase of Systems Analysis

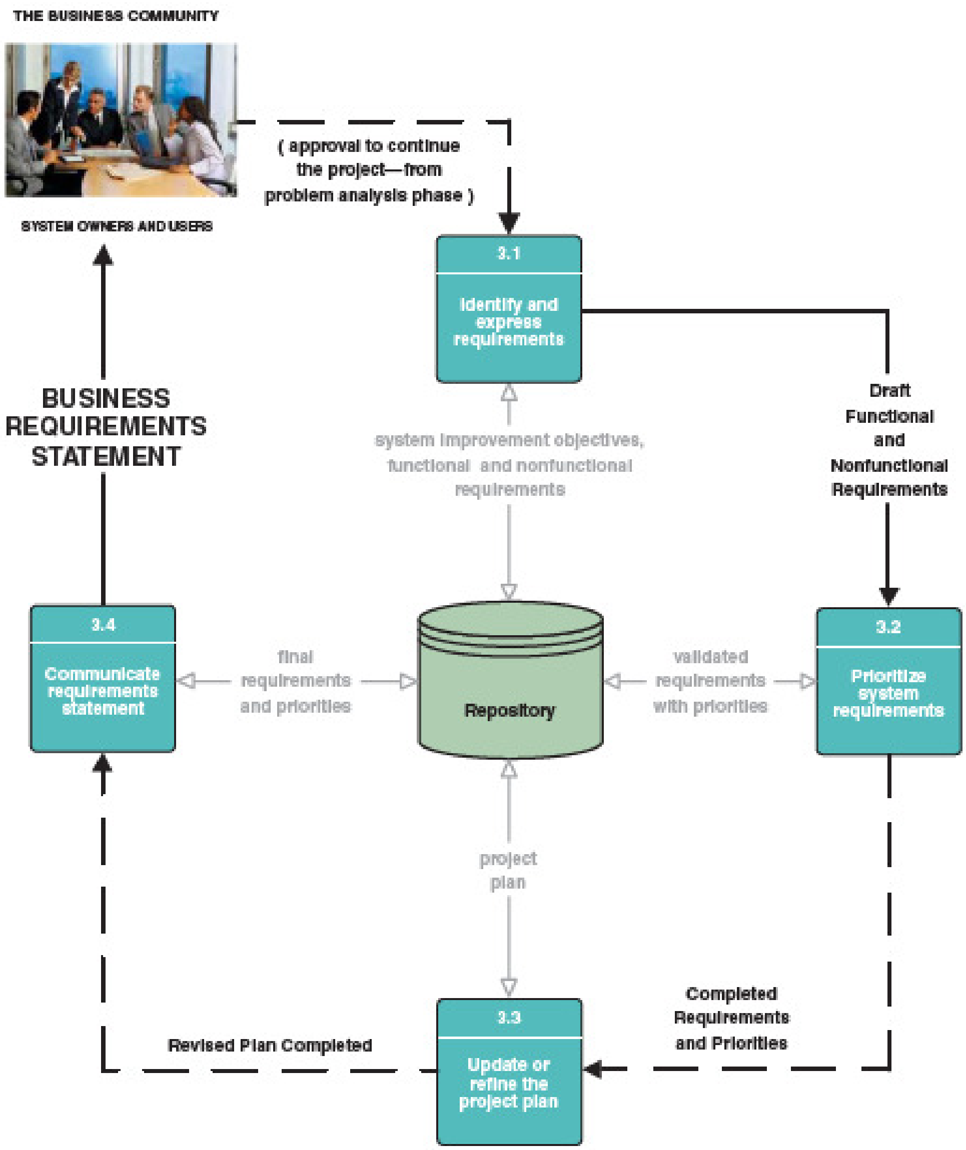


FIGURE 5-15

Tasks for the Requirements Analysis Phase of Systems Analysis

tasks that will add new requirements and details to the outline. Thus, neither completeness nor perfection is a goal of this task.

SYSTEMS ANALYSTS facilitate the task. They also document the results. Obviously, SYSTEM USERS are the primary source of business requirements. Some SYSTEM OWNERS may elect to participate in this task since they played a role in framing the system improvement objectives that will guide the task. SYSTEM DESIGNERS and BUILDERS should not be involved because they tend to prematurely redirect the focus to the technology and technical solutions.

As shown in Figure 5-15, this task (and phase) is triggered by the APPROVAL TO CONTINUE THE PROJECT FROM THE PROBLEM ANALYSIS PHASE. The key input is the SYSTEM IMPROVEMENT OBJECTIVES from the problem analysis phase (via the repository). Of course, any and all relevant information from the problem analysis phase is available from the repository for reference as needed.

The only deliverable of this task is the DRAFT FUNCTIONAL AND NONFUNCTIONAL REQUIREMENTS. Various formats can work. In its simplest format, the outline could be divided into four logical sections: the original list of system improvement objectives and, for each objective, a sublist of (a) inputs, (b) processes, (c) outputs, and (d) stored data needed to fulfill the objective. Increasingly, however, system analysts are

use case a business scenario or event for which the system must provide a defined response. Use cases evolved out of object-oriented analysis; however, their use has become common in many other methodologies for systems analysis and design.

expressing functional requirements using a modeling tool called **use cases.** Use cases model business scenarios and events that must be handled by a new system. They are introduced in Chapter 7 and used throughout this book.

The PIECES framework that was used earlier to identify problems, opportunities, and constraints can also be used as a framework for defining draft requirements.

Several techniques are applicable to this task. Joint requirements planning (JRP) is the preferred technique for rapidly outlining business requirements. Alternatively, the analysts could use other fact-finding methods such as surveys and interviews. Both JRP and fact-finding are taught in the next chapter.

> Task 3.2—Prioritize System Requirements

We stated earlier that the success of a systems development project can be measured in terms of the degree to which business requirements are met. But not all requirements are created equal. If a project gets behind schedule or over budget, it may be useful to recognize which requirements are more important than others. Thus, given the validated requirements, system owners and users should *prioritize system requirements.*

timeboxing a technique that delivers information systems functionality and requirements through versioning. The development team selects the smallest subset of the system that, if fully implemented, will return immediate value to the system owners and users. That subset is developed, ideally with a time frame of six to nine months or less. Subsequently, value-added versions of the system are developed in similar time frames.

Prioritization of requirements can be facilitated using a popular technique called **timeboxing.** Timeboxing attempts to divide requirements into "chunks" that can be implemented within a period of time that does not tax the patience of the user and management community. Timeboxing forces priorities to be clearly defined.

SYSTEMS ANALYSTS facilitate the prioritization task. SYSTEM OWNERS and USERS establish the actual priorities. SYSTEM DESIGNERS and BUILDERS are not involved in the task. The task is triggered by the VALIDATED REQUIREMENTS. It should be obvious that you cannot adequately prioritize an incomplete set of requirements. The deliverable of this task is the REQUIREMENTS WITH PRIORITIES. Priorities can be classified according to their relative importance:

- A *mandatory requirement* is one that must be fulfilled by the minimal system, version 1.0. The system is useless without it. Careful! There is a temptation to label too many requirements as mandatory. A mandatory requirement cannot be ranked because it is essential to any solution. In fact, if an alleged mandatory requirement can be ranked, it is actually a desirable requirement.
- A *desirable requirement* is one that is not absolutely essential to version 1.0. It may still be essential to the vision of some future version. Desirable requirements can and should be ranked. Using version numbers as the ranking scheme is an effective way to communicate and categorize desirable requirements.

> Task 3.3—Update or Refine the Project Plan

Here again, recall that project scope is a moving target. Now that we've identified the business system requirements, we should step back and redefine our understanding of the project scope and update our project plan accordingly. The team must consider the possibility that the new system may be larger than originally expected. If so, the team must adjust the schedule, budget, or scope accordingly. We should also secure approval to continue the project into the next phase. (Work may have already started on the design phases; however, the decisions still require review.)

The project manager, in conjunction with SYSTEM OWNERS and the entire project team, facilitates this task. As usual, the project manager and SYSTEM OWNERS are the key individuals in this task. They should consider the possibility that the requirements now exceed the original vision that was established for the project and new system. They may have to reduce the scope to meet a deadline or increase the budget to get the job done.

As shown in Figure 5-15, this task is triggered by completion of the COMPLETED REQUIREMENTS AND PRIORITIES. The up-to-date PROJECT PLAN is the other key input, and it is updated in the repository as appropriate. The tools, techniques, and steps for maintenance of the project plan were covered in Chapter 4, "Project Management."

> Task 3.4—Communicate the Requirements Statement

Communication is an ongoing task of the requirements analysis phase. We must communicate requirements and priorities to the business community throughout the phase. Users and managers will frequently lobby for requirements and priority consideration. Communication is the process through which differences of opinion must be mediated. The project manager and executive sponsor should jointly facilitate this task. Today, a project intranet or portal is frequently used to communicate requirements. Some systems allow users and managers to *subscribe* to requirements documents to ensure they are notified as changes occur. Interpersonal, communications, and negotiation skills are essential to this task.

> Ongoing Requirements Management

The requirements analysis phase is now complete. Or is it? It was once popular to freeze the business requirements before beginning the system design and construction phases. But today's economy has become increasingly fast-paced. Businesses are measured on their ability to quickly adapt to constantly changing requirements and opportunities. Information systems can be no less responsive than the business itself. Thus, requirements analysis really never ends. While we quietly transition to the remaining phases of our project, there remains an ongoing need to continuously manage requirements through the course of the project and the lifetime of the system.

Requirements management defines a process for system owners, users, analysts, designers, and builders to submit proposed changes to requirements for a system. The process specifies how changes are to be requested and documented, how they will be logged and tracked, when and how they will be assessed for priority, and how they will eventually be satisfied (if they are ever satisfied).

The Logical Design Phase

Not all projects embrace model-driven development, but most include some amount of system modeling. A logical design further documents business requirements using system models that illustrate data structures, business processes, data flows, and user interfaces (increasingly using object models, as introduced earlier in the chapter). In a sense, they validate the requirements established in the previous phase.

Once again, your information systems building blocks (Figure 5-16) can serve as a useful framework for documenting the information systems requirements. Notice that we are still concerned with the SYSTEM USERS' perspectives. In this phase, we draw various system models to document the requirements for a new and improved system. The models depict various aspects of our building blocks. Alternatively, prototypes could be built to "discover requirements." Discovery prototypes were introduced earlier in the chapter. Recall that some prototypes can be reverse engineered into system models.

Figure 5-17 illustrates the typical tasks of the logical design phase. The final phase deliverable and milestone is producing a BUSINESS REQUIREMENTS STATEMENT that will fulfill the system improvement objectives identified in the previous phase. One of the first things you may notice in this task diagram is that most of the tasks are not as sequential as in previous task diagrams. Instead, many of these tasks occur in parallel as the team works toward the goal of completing the requirements statement.

The logical design phase typically includes the following tasks:

4.1a Structure functional requirements.
4.1b Prototype functional requirements.
4.2 Validate functional requirements.
4.3 Define acceptance test cases.

Let's now examine each of these tasks in greater detail.

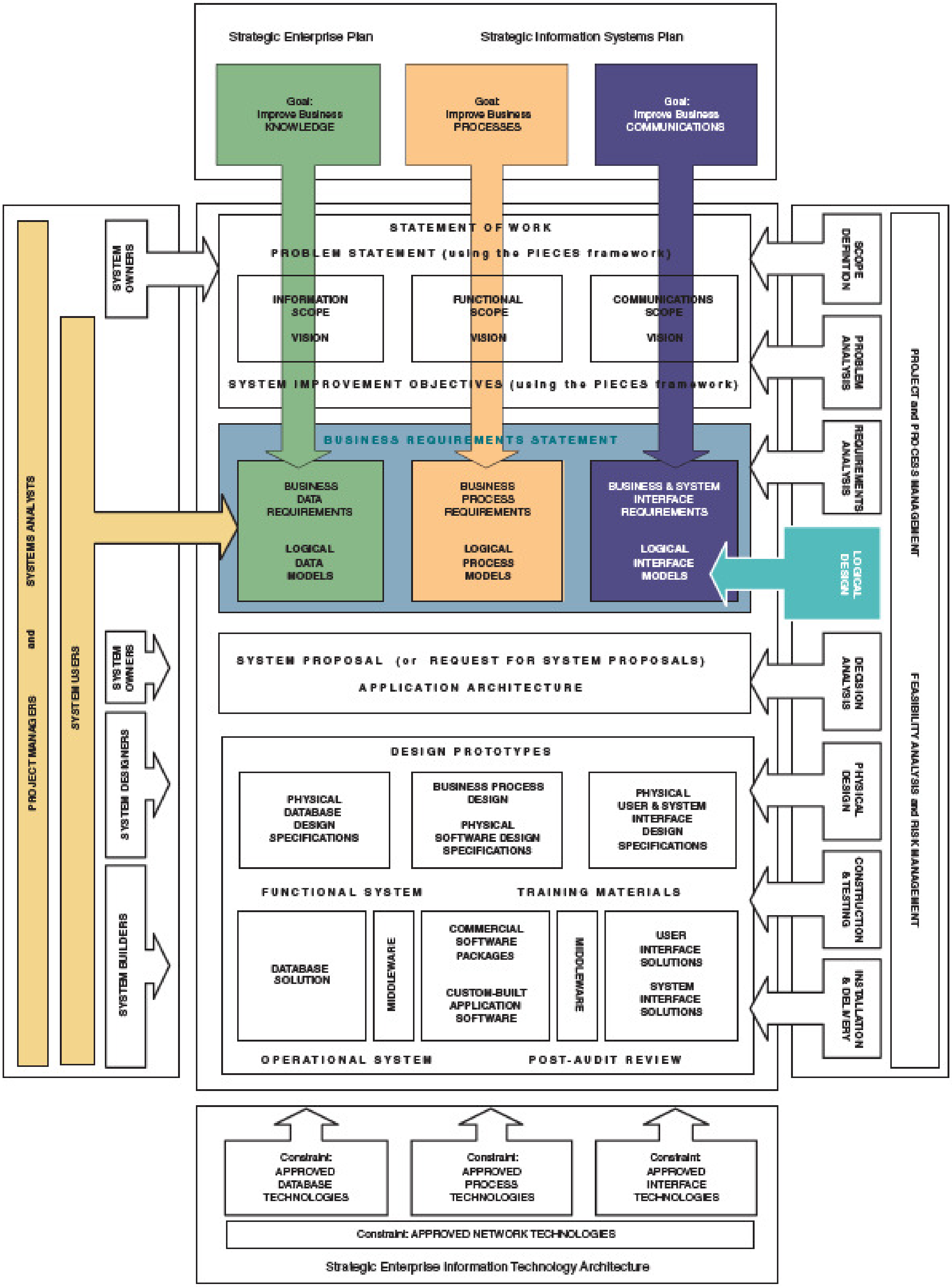


FIGURE 5-16 The Context of the Logical Design Phase of Systems Analysis

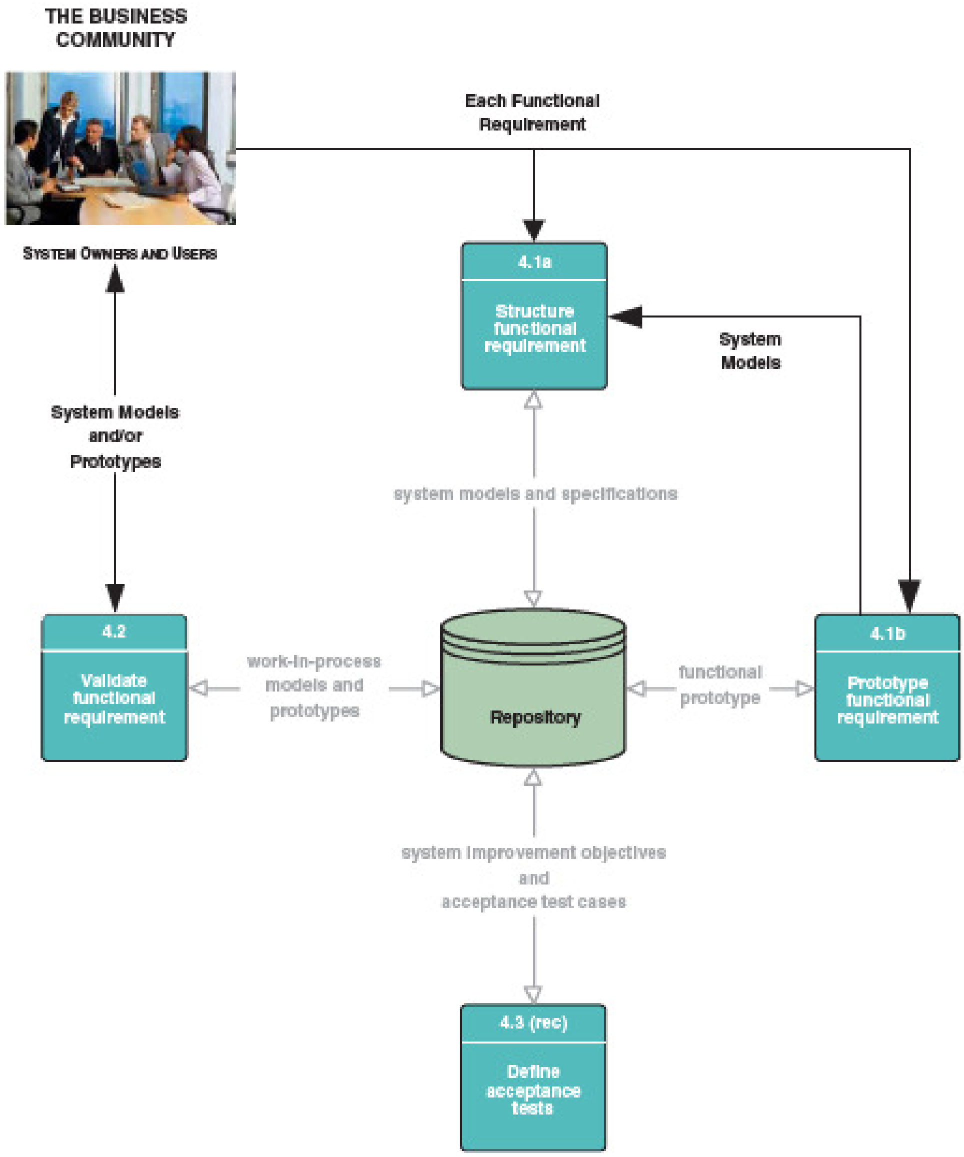


FIGURE 5-17
Tasks for the Logical Design Phase of Systems Analysis

> Task 4.1a—Structure Functional Requirements

One approach to logical design is to *structure the functional requirements.* This means that, using agile methods, you should draw or update one or more system models to illustrate the functional requirement. These may include any combination of data, process, and object models that accurately depict the business and user requirements (but not any technical solution). System models are not complete until all appropriate functional requirements have been modeled. Models are frequently supplemented with detailed logical specifications that describe data attributes, business rules and policies, and the like.

SYSTEMS ANALYSTS facilitate the task. They also document the results. Obviously, SYSTEM USERS are the primary source of factual details needed to draw the models. As shown in Figure 5-17, this task (and phase) is triggered by *each* FUNCTIONAL REQUIREMENT. The outputs are the actual SYSTEM MODELS AND DETAILED SPECIFICATIONS. The level of detail required depends on the methodology being followed. Agile methods usually require "just enough" documentation. How much is enough? That is arguable, but agile methodologists hold that every deliverable should be essential to the forthcoming design and programming phases. This textbook will teach you a variety of different system modeling tools and techniques to apply to logical design.

> Task 4.1b—Prototype Functional Requirements (alternative)

Prototyping is an alternative (and sometimes a prerequisite) to system modeling. Sometimes users have difficulty expressing the facts necessary to draw adequate system models. In such a case, an alternative or complementary approach to system modeling is to build discovery prototypes. Prototyping is typically used in the requirements analysis phase to build sample inputs and outputs. These inputs and outputs help to construct the underlying database and the programs for inputting and outputting the data to and from the database. Although discovery prototyping is optional, it is frequently applied to systems development projects, especially in cases where the users are having difficulty stating or visualizing their business requirements. The philosophy is that the users will recognize their requirements when they see them.

SYSTEMS BUILDERS facilitate this analysis task. SYSTEM ANALYSTS document and analyze the results. As usual, SYSTEM USERS are the primary source of factual input to the task. Figure 5-17 demonstrates that this task is dependent on one or more FUNCTIONAL REQUIREMENTS that have been identified by the users. The system builders and analysts respond by constructing the PROTOTYPES. As described earlier in this chapter, it may be possible to *reverse engineer* some SYSTEM MODELS directly from the prototype databases and program libraries.

> Task 4.2—Validate Functional Requirements

Both SYSTEM MODELS and PROTOTYPES are representations of the users' requirements. They must be validated for completeness and correctness. SYSTEMS ANALYSTS facilitate the prioritization task by interactively engaging system users to identify errors and omissions or make clarifications.

> Task 4.3—Define Acceptance Test Cases

While not a required task, most experts agree that it is not too early to begin planning for system testing. System models and prototypes very effectively define the processing requirements, data rules, and business rules for the new system. Accordingly, these specifications can be used to define TEST CASES that can ultimately be used to test programs for correctness. Either SYSTEM ANALYSTS or SYSTEM BUILDERS can perform this task and validate the test cases with the SYSTEM USERS.

Recall that SYSTEM IMPROVEMENT OBJECTIVES were defined earlier in the project. Test cases can be defined to test these objectives as well.

The Decision Analysis Phase

Given the business requirements for an improved information system, we can finally address how the new system—including computer-based alternatives—*might* be implemented with technology. The purpose of the *decision analysis phase* is to identify candidate solutions, analyze those candidate solutions, and recommend a target system that will be designed, constructed, and implemented. Chances are that someone has already championed a vision for a technical solution. But alternative solutions, perhaps better ones, nearly always exist. During the decision analysis phase, it is imperative that you identify options, analyze those options, and then sell the best solution based on the analysis.

Once again, your information systems building blocks (Figure 5-18) can serve as a useful framework for the decision analysis phase. One of the first things you should notice is that information technology and architecture begin to influence the decisions we must make. In some cases, we must work within standards. In other cases,

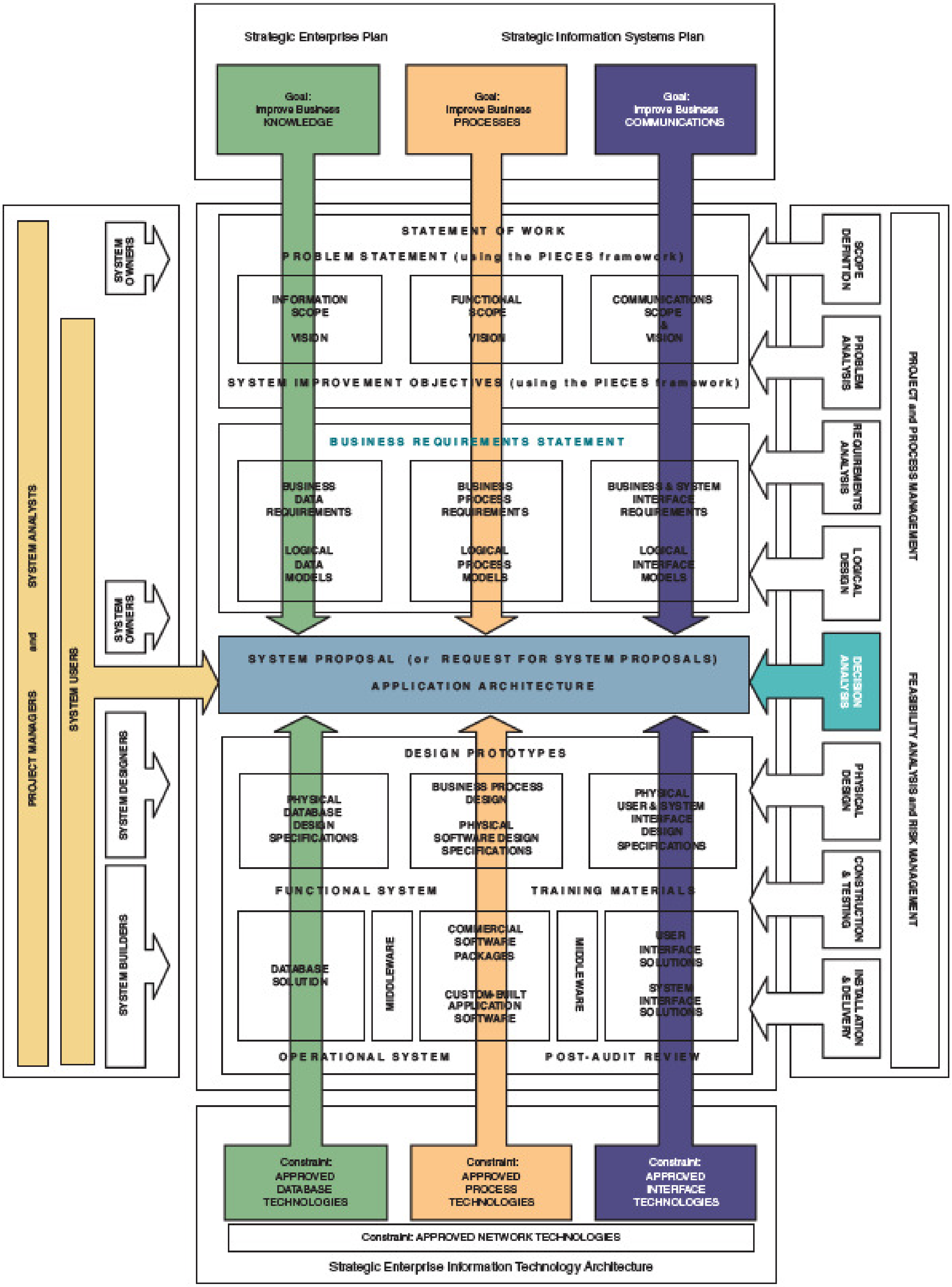
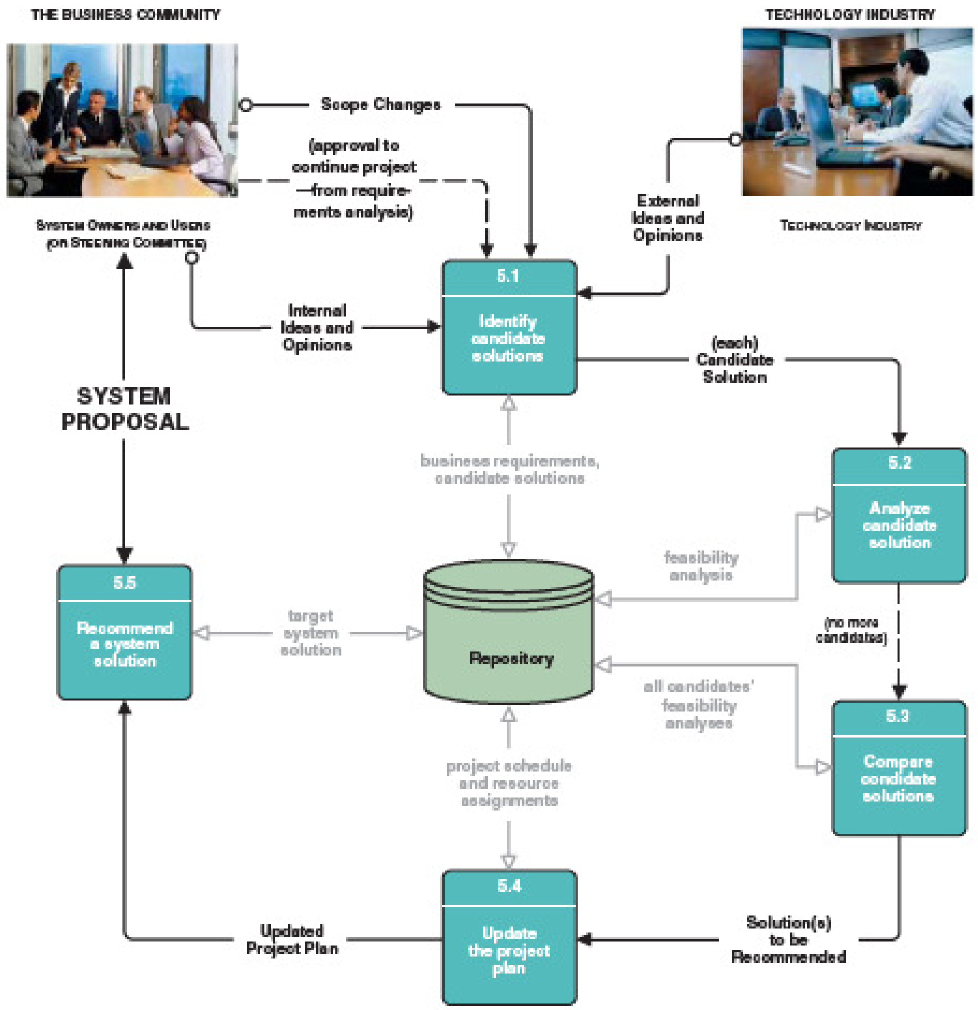


FIGURE 5-18 The Context of the Decision Analysis Phase of Systems Analysis

FIGURE 5-19
Tasks for the Decision Analysis Phase of Systems Analysis



we can look to apply different or emerging technology. You should also notice that the perspectives are in transition—from those of the SYSTEM USERS to those of the SYSTEM DESIGNERS. Again, this reflects our transition from pure business concerns or technology. But we are not yet designing. The building blocks indicate our goal as developing a proposal that will fulfill requirements.

Figure 5-19 illustrates the typical tasks of the decision analysis phase. The final phase deliverable and milestone is producing a SYSTEM PROPOSAL that will fulfill the business requirements identified in the previous phase. The decision analysis phase typically includes the following tasks:

5.1 Identify candidate solutions.
5.2 Analyze candidate solutions.
5.3 Compare candidate solutions.
5.4 Update the project plan.
5.5 Recommend a system solution.

Let's now examine each of these tasks in greater detail.

> Task 5.1—Identify Candidate Solutions

Given the business requirements established in the definition phase of systems analysis, we must first identify alternative candidate solutions. Some candidate solutions will be posed by design ideas and opinions from SYSTEM OWNERS and USERS. Others may come from various sources including SYSTEMS ANALYSTS, SYSTEMS DESIGNERS, technical consultants, and other IS professionals. And some technical choices may be limited by a predefined, approved technology architecture. It is the intent of this task not to evaluate the candidates but, rather, simply to define possible candidate solutions to be considered.

The SYSTEMS ANALYSTS facilitate this task. SYSTEM OWNERS and USERS are not normally directly involved in this task, but they may contribute ideas and opinions that start the task. For example, an owner or user may have read an article about, heard about, or learned how some competitor's or acquaintance's similar system was implemented. In any case, it is politically sound to consider the ideas. SYSTEM DESIGNERS and BUILDERS such as database administrators, network administrators, technology architects, and programmers are also a source of ideas and opinions.

As shown in Figure 5-19, this task is formally triggered by the APPROVAL TO CONTINUE THE PROJECT FROM THE REQUIREMENTS ANALYSIS phase. In reality, ideas and opinions have been generated and captured since the preliminary investigation phase—it is human nature to suggest solutions throughout any problem-solving process. Notice that, in addition to coming from the project team itself, IDEAS AND OPINIONS can be generated from both internal and external sources. Each idea generated is considered to be a CANDIDATE SOLUTION to the BUSINESS REQUIREMENTS.

The amount of information describing the characteristics of any one candidate solution may become overwhelming. A candidate matrix, such as Figure 5-20, is a useful tool for effectively capturing, organizing, and comparing the characteristics of different candidate solutions.

As has been the case throughout this chapter, fact-finding and group facilitation techniques like JRP are the principle techniques used to research candidate system solutions. Fact-finding and group facilitation techniques are taught in the next chapter. Also, Chapter 10, "Feasibility Analysis and the System Proposal," will teach you how to actually generate candidate system solutions and document them in the matrix.

> Task 5.2—Analyze Candidate Solutions

Each candidate system solution must be analyzed for feasibility. This can occur as each candidate is identified or after all candidates have been identified. Feasibility analysis should not be limited to costs and benefits. Most analysts evaluate solutions against at least four sets of criteria:

- *Technical feasibility*—Is the solution technically practical? Does our staff have the technical expertise to design and build this solution?
- *Operational feasibility*—Will the solution fulfill the user's requirements? To what degree? How will the solution change the user's work environment? How do users feel about such a solution?
- *Economic feasibility*—Is the solution cost-effective?
- *Schedule feasibility*—Can the solution be designed and implemented within an acceptable time period?

When completing this task, the analysts and users must take care not to make comparisons between the candidates. The feasibility analysis is performed on each individual candidate without regard to the feasibility of other candidates. This approach discourages the analyst and users from prematurely making a decision concerning which candidate is the best.

Again, the SYSTEMS ANALYSTS facilitate the task. Usually SYSTEMS OWNERS and USERS analyze operational, economic, and schedule feasibility. SYSTEMS DESIGNERS and BUILDERS usually contribute to the analyses and play the critical role in analyzing technical feasibility.

Figure 5-19 shows that the task is triggered by the completion of each candidate solution; however, it is acceptable to delay the task until all candidate solutions have been identified. Input to the actual feasibility analyses comes from the various team

FIGURE 5-20 A Candidate Systems Matrix

Characteristics	Candidate 1	Candidate 2	Candidate 3	Candidate . . .
Portion of System Computerized Brief description of that portion of the system that would be computerized in this candidate.	COTS package Platinum Plus from Entertainment Software Solutions would be purchased and customized to satisfy Member Services required functionality.	Member Services and warehouse operations in relation to order fulfillment.	Same as candidate 2.	
Benefits Brief description of the business benefits that would be realized for this candidate.	This solution can be implemented quickly because it's a purchased solution.	Fully supports user's required business processes for SoundStage Inc. Plus more efficient interaction with member accounts.	Same as candidate 2.	
Servers and Workstations A description of the servers and workstations needed to support this candidate.	Technically, architecture dictates Pentium Pro, MS Windows NT class servers and Pentium, MS Windows NT 4.0 workstations (clients).	Same as candidate 1.	Same as candidate 1.	
Software Tools Needed Software tools needed to design and build the candidate (e.g., database management system, emulators, operating systems, languages, etc.). Not generally applicable if applications software packages are to be purchased.	MS Visual C++ and MS Access for customization of package to provide report writing and integration.	MS Visual Basic 5.0 System Architect 3.1 Internet Explorer	MS Visual Basic 5.0 System Architect 3.1 Internet Explorer	
Application Software A description of the software to be purchased, built, accessed, or some combination of these techniques.	Package solution.	Custom solution	Same as candidate 2.	
Method of Data Processing Generally some combination of online, batch, deferred batch, remote batch, and real time.	Client/server.	Same as candidate 1.	Same as candidate 1.	
Output Devices and Implications A description of output devices that would be used, special output requirements (e.g., network, preprinted forms, etc.), and output considerations (e.g., timing constraints).	(2) HP4MV department laser printers. (2) HP5SI LAN laser printers.	(2) HP4MV department laser printers. (2) HP5SI LAN laser printers. (1) PRINTRONIX bar code printer (includes software & drivers). Web pages must be designed to VGA resolution. All internal screens will be designed for SVGA resolution.	Same as candidate 2.	
Input Devices and Implications A description of input methods to be used, input devices (keyboard, mouse, etc.), special input requirements (e.g., new or revised forms from which data would be input), and input considerations (e.g., timing of actual inputs).	Keyboard & mouse.	Apple "Quick Take" digital camera and software. (15) PSC Quickscan laser bar code scanners. (1) HP Scanjet 4C flatbed scanner. Keyboard & mouse.	Same as candidate 2.	
Storage Devices and Implications Brief descriptions of what data would be stored, what data would be accessed from existing stores, what storage media would be used, how much storage capacity would be needed, and how data would be organized.	MS SQL Server DBMS with 100GB arrayed capability.	Same as candidate 1.	Same as candidate 1.	

participants; however, it is not uncommon for external experts (and influences) to also provide data. The feasibility analysis for each candidate is saved in the repository for later comparison to other candidates.

Fact-finding techniques, again, play a role in this systems analysis task. But the ability to perform a feasibility analysis on a candidate system solution is essential. That technique is taught in Chapter 10, "Feasibility Analysis and the System Proposal."

> Task 5.3—Compare Candidate Solutions

Once the feasibility analysis has been completed for each candidate solution, we can compare the candidates and select one or more solutions to recommend to the SYSTEM OWNERS and USERS. At this point, any infeasible candidates are usually eliminated from further consideration. Since we are looking for the most feasible solution of those remaining, we will identify and recommend the candidate that offers the best overall combination of technical, operational, economic, and schedule feasibilities. It should be noted that in selecting such a candidate, it is rare that a given candidate is found to be the most operational, technical, economic, and schedule feasible.

Once again, the SYSTEMS ANALYSTS facilitate the task. SYSTEM DESIGNERS and BUILDERS should be available to answer any technical feasibility questions. But ultimately, the SYSTEMS OWNERS and USERS should be empowered to drive the final analysis and recommendation.

In Figure 5-19, this task is triggered by the completion of the feasibility analysis of all candidate solutions (NO MORE CANDIDATE SOLUTIONS). The input is ALL OF THE CANDIDATES' FEASIBILITY ANALYSES. Once again, a matrix can be used to communicate the large volume of information about candidate solutions. The feasibility matrix in Figure 5-21 allows a side-by-side comparison of the different feasibility analyses for a number of candidates.

The deliverable of this task is the SOLUTION(S) TO BE RECOMMENDED. If more than one solution is recommended, priorities should be established.

Again, feasibility analysis techniques (and the matrix) will be taught in Chapter 10, "Feasibility Analysis and the System Proposal."

> Task 5.4—Update the Project Plan

Hopefully, you noticed a recurring theme throughout this chapter. We are continually updating our project plan as we learn more about a system, its problems, its requirements, and its solutions. We are adjusting scope accordingly. Thus, based on our recommended solution(s), we should once again reevaluate project scope and *update the project plan* accordingly.

The project manager, in conjunction with SYSTEM OWNERS and the entire project team, facilitates this task. The SYSTEMS ANALYSTS and SYSTEM OWNERS are the key individuals in this task. But because we are transitioning into technical system design, we need to begin involving the SYSTEM DESIGNERS and BUILDERS in the project plan updates.

As shown in Figure 5-19, this task is triggered by completion of the SOLUTION(S) TO BE RECOMMENDED. The latest PROJECT SCHEDULE AND RESOURCE ASSIGNMENTS must be reviewed and updated. The UPDATED PROJECT PLAN is the key output. The updated plan should now include a detailed plan for the system design phase that will follow. The techniques and steps for updating the project plan were taught in Chapter 4, "Project Management."

> Task 5.5—Recommend a System Solution

As with the preliminary investigation and problem analysis phases, the decision analysis phase concludes with a communication task. We must *recommend a system solution* to the business community.

FIGURE 5-21 A Feasibility Analysis Matrix

Feasibility Criteria	Weight	Candidate 1	Candidate 2	Candidate 3	Candidate . . .
Operational Feasibility **Functionality.** A description of to what degree the candidate would benefit the organization and how well the system would work. **Political.** A description of how well received this solution would be from user management, user, and organization perspectives.	30%	Only supports Member Services requirements, and current business processes would have to be modified to take advantage of software functionality. **Score: 60**	Fully supports user's required functionality. **Score: 100**	Same as candidate 2. **Score: 100**	
Technical Feasibility **Technology.** An assessment of the maturity, availability (or ability to acquire), and desirability of the computer technology needed to support this candidate. **Expertise.** An assessment of the technical expertise needed to develop, operate, and maintain the candidate system.	30%	Current production release of Platinum Plus package is version 1.0 and has been on the market for only 6 weeks. Maturity of product is a risk, and company charges an additional monthly fee for technical support. Required to hire or train C++ expertise to perform modifications for integration requirements. **Score: 50**	Although current technical staff has only Powerbuilder experience, the senior analysts who saw the MS Visual Basic demonstration and presentation have agreed the transition will be simple and finding experienced VB programmers will be easier than finding Powerbuilder programmers and at a much cheaper cost. MS Visual Basic 5.0 is a mature technology based on version number. **Score: 95**	Although current technical staff is comfortable with Powerbuilder, management is concerned with recent acquisition of Powerbuilder by Sybase Inc. MS SQL Server is a current company standard and competes with SYBASE in the client/server DBMS market. Because of this we have no guarantee future versions of Power-builder will "play well" with our current version SQL Server. **Score: 60**	
Economic Feasibility **Cost to develop:** **Payback period (discounted):** **Net present value:** **Detailed calculations:**	30%	Approximately $350,000. Approximately 4.5 years. Approximately $210,000. See Attachment A. **Score: 60**	Approximately $418,040. Approximately 3.5 years. Approximately $306,748. See Attachment A. **Score: 85**	Approximately $400,000. Approximately 3.3 years. Approximately $325,500. See Attachment A. **Score: 90**	
Schedule Feasibility An assessment of how long the solution will take to design and implement.	10%	Less than 3 months. **Score: 95**	9–12 months. **Score: 80**	9 months. **Score: 85**	
Ranking	**100%**	**60.5**	**92**	**83.5**	

The project manager and executive sponsor should jointly facilitate this task. Other meeting participants should include the entire project team, including assigned SYSTEM OWNERS, USERS, ANALYSTS, DESIGNERS, and BUILDERS. As usual, the meeting should be open to any and all interested staff from the business community. Also, if an intranet Web site was established for the project, it should have been maintained throughout the problem analysis phases to ensure continuous communication of project progress.

This task is triggered by the completion of the UPDATED PROJECT PLAN. The TARGET SYSTEM SOLUTION (from Task 4.3) is reformatted for presentation as a SYSTEM PROPOSAL. The format may be a report, a verbal presentation, or an inspection by an auditor or peer group (called a *walkthrough*). An outline for a written report is shown in Figure 5-22.

The Next Generations: Systems Analysis

Predicting the future of systems analysis is not easy, but we'll make an attempt. CASE technology will continue to improve, making it easier to model system requirements. First, CASE tools will include object modeling to support emerging object-oriented analysis techniques. While some CASE tools will be purely object-oriented, we believe that the demand for other types of modeling support (e.g., data modeling for databases, process modeling for BPR) will place a premium on comprehensive CASE tools that can support many types of models. Second, the reverse-engineering technology in CASE tools will continue to improve our ability to more quickly generate first-draft system models from existing databases and application programs.

In the meantime, RAD technology will continue to enable accelerated analysis approaches such as prototyping. We also expect the trend for RAD and CASE tools to interoperate through reverse and forward engineering to further simplify both system modeling and discovery prototyping.

Object-oriented analysis will eventually replace structured analysis and information engineering as the best practice for systems analysis. This change may not occur as rapidly as object purists would like, but it will occur all too rapidly for a generation of systems analysts who are skilled in the older methods. There is a grand opportunity for talented young analysts to lead the transition; however, career opportunities will remain strong for analysts who know data modeling that will continue to be used for database design. Also, the process modeling renaissance will continue as BPR projects continue to proliferate.

We also predict our systems proposals will continue to get more interesting. As the Internet, e-commerce, and e-business become increasingly pervasive in our economy, systems analysts will be proposing new alternatives to old problems. There will be a fundamental change in business and information systems to use these new technologies.

One thing will *not* change! We will continue to need systems analysts who understand how to fundamentally investigate and analyze business problems and define the logical business requirements as a preface to system design. But we will all have to do that with increased speed and accuracy to meet the accelerated systems development schedules required in tomorrow's faster-paced economy.

Interpersonal and communications skills are essential to this task. Soft skills such as salesmanship and persuasion become important. (Many schools offer speech and communications courses on these subjects.) Systems analysts should be able to write a formal business report and make a business presentation without getting into technical issues or alternatives.

This concludes the decision analysis phase. And it also concludes our coverage of systems analysis.

FIGURE 5-22 An Outline for a Typical System Proposal

I. Introduction
   A. Purpose of the report
   B. Background of the project leading to this report
   C. Scope of the project
   D. Structure of the report
II. Tools and techniques used
   A. Solution generated
   B. Feasibility analysis (cost-benefit)
III. Information systems requirements
IV. Alternative solutions and feasibility analysis
V. Recommendations
VI. Appendixes

Learning Roadmap

This chapter provided a detailed overview of the systems analysis phases of a project. You are now ready to learn some of the systems skills introduced in this chapter. For most students, this would be the ideal time to study the fact-finding techniques that were identified as critical to almost every phase and task that was described in this chapter. Chapter 6 teaches these skills. It is recommended that you read Chapter 7, "Modeling System Requirements with Use Cases," before proceeding to any of the modeling chapters since use cases are commonly used to facilitate the activity of modeling.

The sequencing of the system modeling chapters is flexible; however, we personally prefer and recommend that Chapter 8, "Data Modeling and Analysis," be studied first. All information systems include databases, and data modeling is an essential skill for database development. Also, it is easier to synchronize early data models with later process models than vice versa. Your instructor may prefer that you first study Chapter 9, "Process Modeling." Advanced courses may elect to jump straight to Chapter 10 to learn about object-oriented analysis and modeling with UML.

If you do jump straight to a system modeling chapter from this chapter, make a commitment to return to Chapter 6 to study the fact-finding techniques. Regardless of how well you master system modeling, that modeling skill is entirely dependent on your ability to discover and collect the business facts that underlie the models.

For those of you who have already completed a systems analysis course, this chapter was probably scheduled only as a review or context for systems design. We suggest that you merely review the system modeling chapters and proceed directly to Chapter 12, "Systems Design." That chapter will pick up where this chapter left off.

Summary

1. Formally, systems analysis is the dissection of a system into its component pieces. As a problem-solving phase, it precedes systems design. With respect to information systems development, systems analysis is the preliminary investigation of a proposed project, the study and problem analysis of the existing system, the requirements analysis of business requirements for the new system, and the decision analysis for alternative solutions to fulfill the requirements.
2. The results of systems analysis are stored in a repository for use in later phases and projects.
3. There are several popular or emerging strategies for systems analysis. These techniques can be used in combination with one another.
   a. Model-driven analysis techniques emphasize the drawing of pictorial system models that represent either a current reality or a target vision of the system.
      i) Structured analysis is a technique that focuses on modeling processes.
      ii) Information engineering is a technique that focuses on modeling data.
      iii) Object-oriented analysis is a technique that focuses on modeling objects that encapsulate the concerns of data and processes that act on that data.
   b. Accelerated analysis approaches emphasize the construction of working models of a system in an effort to accelerate systems analysis.
      i) Discovery prototyping is a technique that focuses on building small-scale, functional subsystems to discover requirements.
      ii) Rapid architected analysis attempts to automatically generate system models from either prototypes or existing systems. The automatic generation of models requires reverse engineering technology.

c. Both model-driven and accelerated system analysis approaches are dependent on requirements discovery techniques to identify or extract problems and requirements from system owners and users.

   i) Fact-finding is the formal process of using research, interviews, questionnaires, sampling, and other techniques to collect information.
   ii) Joint requirements planning (JRP) techniques use facilitated workshops to bring together all interested parties and accelerate the fact-finding process.

d. Business process redesign is a technique that focuses on simplifying and streamlining fundamental business processes before applying information technology to those processes.

4. Each phase of systems analysis (preliminary investigation, problem analysis, requirements analysis, and decision analysis) can be understood in the context of the information system building blocks: KNOWLEDGE, PROCESSES, and COMMUNICATIONS.
5. The purpose of the preliminary investigation phase is to determine the worthiness of the project and to create a plan to complete those projects deemed worthy of a detailed study and analysis. To accomplish the preliminary investigation phase, the systems analyst will work with the system owners and users to: *(a)* list problems, opportunities, and directives; *(b)* negotiate preliminary scope; *(c)* assess project worth; *(d)* plan the project, and *(e)* present the project to the business community. The deliverable for the preliminary investigation phase is a project charter that must be approved by system owners and/or a decision-making body, commonly referred to as the steering committee.
6. The purpose of the problem analysis phase is to answer the questions, Are the problems really worth solving, and is a new system really worth building? To answer these questions, the problem analysis phase thoroughly analyzes the alleged problems and opportunities first identified in the preliminary investigation phase. To complete the problem analysis phase, the analyst will continue to work with the system owner, system users, and other IS management and staff. The systems analyst and appropriate participants will *(a)* study the problem domain; *(b)* thoroughly analyze problems and opportunities; *(c)* optionally, analyze business processes; *(d)* establish system improvement objectives and constraints; *(e)* update the project plan; and *(f)* present the findings and recommendations. The deliverable for the problem analysis phase is the system improvement objectives.
7. The purpose of the requirements analysis phase is to identify what the new system is to do without the consideration of technology—in other words, to define the business requirements for a new system. As in the preliminary investigation and problem analysis phases, the analyst actively works with system users and owners as well as other IS professionals. To complete the requirements analysis phase, the analyst and appropriate participants will *(a)* define requirements, *(b)* analyze functional requirements using system modeling and/or discovery prototyping, *(c)* trace and complete the requirements statement, *(d)* prioritize the requirements, and *(e)* update the project plan and scope. The deliverable of the requirements analysis phase is the business requirements statement. Because requirements are a moving target with no finalization, requirements analysis also includes the ongoing task of managing changes to the requirements.
8. The purpose of the logical design phase is to document business requirements using system models for the proposed system. These system models can, depending on the methodology, be any combination of process models, data models, and object models. The models depict various aspects of our building blocks. Alternatively, prototypes could be built to "discover requirements." Some discovery prototypes can be reverse engineered into system models. The systems analyst and appropriate participants will *(a)* structure or prototype functional requirements, *(b)* validate functional requirements, and *(c)* define acceptance test cases. These tasks are not necessarily sequential; they can occur in parallel. The deliverable for the logical design phase is the business requirements statement.
9. The purpose of the decision analysis phase is to transition the project from business concerns to technical solutions by identifying, analyzing, and recommending a technical system solution. To complete the decision analysis phase, the analyst and appropriate participants will *(a)* define candidate solutions; *(b)* analyze candidate solutions for feasibility (technical, operational, economic, and schedule feasibility); *(c)* compare feasible candidate solutions to select one or more recommended solutions; *(d)* update the project plan based on the recommended solution; and *(e)* present and defend the target solution. The deliverable of the decision analysis phase is the system proposal.

Review Questions

1. What are the business factors that are driving systems analysis? Based on these factors, what should systems analysis address?
2. What is model-driven analysis? Why is it used? Give several examples.
3. What is the major focus of structured analysis?
4. What is the major focus of information engineering?
5. Why has object-oriented analysis become popular? What problems does it solve?
6. What are the five phases of systems analysis?
7. What is the goal of the scope definition phase?
8. What are the five tasks that you do in the scope definition phase?
9. What is the trigger for communicating the project plan, and who is the audience? Why is communicating the project plan important?
10. Why do many new systems analysts fail to effectively analyze problems? What can they do to become more effective?
11. What is a popular tool used to identify and express the functional requirements of a system?
12. What is a commonly used technique for prioritizing system requirements?
13. When could prototyping be used instead of system modeling for determining functional requirements?
14. Why is the decision analysis phase needed?
15. What are some ways to identify candidate solutions?

Problems and Exercises

1. There are many different approaches to systems analysis. Despite these different approaches, what is the universally accepted definition of systems analysis? What is the general consensus as to when systems analysis begins and when it ends? As a project manager, what is important to know regarding the definition of systems analysis, and what is important to ensure in your organization regarding the definition?
2. As a systems analyst, you will be exposed to and use many different approaches to systems analysis throughout your career. It is important that you understand the conceptual basis of each type of approach, and their essential differences, strengths and weaknesses. Consider the differences in structured analysis, information engineering and data modeling, and object-oriented analysis, all of which represent model-driven analysis, and fill in the matrix shown below.

	CENTRICITY (data, process, etc.)	TYPE OF MODELS USED	ESSENTIAL DIFFERENCES
STRUCTURED ANALYSIS			
INFORMATION ENGINEERING AND DATA MODELING			
OBJECT-ORIENTED ANALYSIS			

3. Accelerated systems analysis approaches are based on the premise that prototypes can help reveal the most important business requirements faster than other methods. Describe the two most commonly used approaches to accelerated analysis. What do they do and how do they do it? What is one of the criticisms of prototyping? Do the accelerated systems analysis approaches completely replace more formal approaches, such as structured analysis?
4. During the scope definition phase, what is one question that you should never lose sight of? And how do you answer this question? What five tasks should occur during the scope definition phase?
5. You are a new systems analyst and eager to prove your abilities on your first project. You are at a problem analysis meeting with the system owners and users and find yourself saying, "We need to do this to solve the problem." Into what common trap are you in danger of falling? What technique could you use to avoid this trap?
6. Your project team has completed the scope definition phase, and is now at the point in the problem analysis phase for establishing system improvement objectives. As the systems analyst on the project team, you are the facilitator of a brainstorming session to define the system improvement objectives. Since several of the project owners and users have never done this before, describe the characteristics of good system improvement objectives and provide some examples. Members of the project team suggest the following objectives:

   a. Reduce the time required to process the order.
   b. The new system must use Oracle to store data.
   c. The data input screens must be redesigned so they are more user-friendly.
   d. The customer satisfaction rate with the online ordering process must be increased by 10 percent.

   Are these examples of good system improvement objectives? Why or why not? If not, how could they be reworded? Also, objectives frequently have constraints that are tied to them; what, if any, do you think the matching constraint might be for each of these objectives?
7. You've made it through the problem analysis phase of the project, and are now beginning the requirements analysis phase. During the first meeting on the business requirements, one of the other analysts on the project team asks the system users, "How should the new system meet your needs?" What common mistake is the analyst making? What are often the consequences of making this mistake?
8. What is the difference between functional and nonfunctional requirements, and what is the purpose of categorizing them into these categories? What are two formats that an analyst can use to document the functional system requirements?
9. Is it important to prioritize system requirements, and if so, when should the requirements be prioritized? What is one technique that can be used, and what is the difference between mandatory and desirable requirements? What is one way to test whether a mandatory requirement is truly a mandatory requirement?
10. Once the system requirements are identified and prioritized, shouldn't everything be frozen to prevent scope or feature creep? Doesn't updating the project plan or allowing stakeholders to continue to request changes just delay system design and construction, and maybe even project completion itself?
11. Why should acceptance test cases be defined during the logical design phase? After all, the technical design hasn't been done yet, let alone building the system. Shouldn't testing activities at least wait until construction is actually underway?
12. How is the logical design phase different from the requirements analysis phase?
13. Let's say you are on the project team of a project that had a great deal of difficulty during the requirements analysis phase, and fell several weeks behind schedule. The project manager wants to try to catch up by either skipping or abbreviating some of the tasks in the logical design phase. After all, the project manager reasons, we really have a clear idea of the requirements now, the designers and builders are really experienced, and they don't really need the logical design in order to do the technical design. Is this a legitimate method to get back on schedule? What are the possible consequences?
14. In identifying and defining possible candidate solutions, what are the typical roles of the various stakeholders who are involved in the project?
15. You are a systems analyst and have been asked to facilitate the analysis and evaluation of several candidate system solutions for their feasibility. What sets of criteria would you typically use? Who do you involve in this task? Should you compare the candidate solutions against each other at this point? Why or why not? What is the typical deliverable coming out of this task?

Projects and Research

1. Select an information system with which you are familiar, and which you feel needs to be improved, based upon your experiences as an employee, customer, other system user, or system owner. Switch roles and perspectives as necessary to perform or answer the following:
   a. Describe the nature of the information system you have selected.
   b. Describe the organization that owns and maintains the information system.
   c. Identify the baseline problems and opportunities, per Task 1.1.
   d. Develop a preliminary problem statement, using the format shown in Figure 5-8.
2. Assume you are the now a systems analyst on the project described in the preceding question. Executive management was extremely impressed by your work on the problem statement. As a result, they have given the project the go-ahead, the baseline schedule and budget have been developed, and the project plan has been approved by the executive steering committee. As the systems analyst, you now have been tasked to do the following:
   a. Develop and document your understanding of the problem domain and business vocabulary, using the textbook's information system building blocks framework as described in Task 2.1.
   b. Analyze problems and opportunities using cause-and-effect analysis (Task 2.2).
   c. Analyze business processes and develop process models (Task 2.3).
   d. Establish system improvement objectives (Task 2.4).
   e. Prepare a Problems, Opportunities, Objectives, and Constraints Matrix, using Figure 5-12 as an example.
3. Communicating findings and recommendations is the final task in the problem analysis phase. As a systems analyst on the project, you have been tasked with preparing the System Improvement Objectives and Recommendations Report. For this exercise, prepare only the Executive Summary portion of the report, using the format shown in Figure 5-13. The executive steering committee will use this summary to make its decisions regarding the recommendations.
4. Your strong work on the project to date has continued to impress executive management. You have received a pay increase and have been tasked with conducting the requirements analysis phase. Specifically:
   a. Identify the system requirements, and prepare an outline of functional and nonfunctional requirements, per Task 3.1. Since your organization uses structured analysis and does not employ use case modeling, list each system improvement objective, and the inputs, processes, outputs, and stored data needed to meet each objective.
   b. Assume that the requirements you identified in the preceding step have been validated. Prioritize the requirements according to their relative importance, using the method described in Task 3.2.
5. Your work has helped keep the project well ahead of schedule, so executive management gives you a couple of weeks of paid vacation. When you return, the project is moving into the decision analysis phase. Your next task is to identify candidate solutions.
   a. Describe the process for identifying candidate solutions. What should you be careful *not* to do at this point?
   b. Develop a candidate systems matrix, using the format in Figure 5-20 as an example, and include three possible solutions.
6. After identifying candidate solutions, the next step is to analyze these solutions.
   a. Describe the process for analyzing candidate solutions. What should the project team *not* do in completing this task?
   b. Develop a Feasibility Analysis Matrix, based upon the candidate solutions identified in the preceding question, and using the format shown in Figure 5-21 as an example. Determine what your weighting factors should be.

Minicases

1. You are the CIO of a major retailer. Recently, you read "Spying on the Sales Floor" in the *Wall Street Journal* on December 21, 2004. You see that your competitors are using video mining to analyze consumer behavior. Should your company also adopt this tool (video mining)? What are the strategic

implications to your company of your competitors' move? What opportunities have been created? Threats?

2. Read "Human Reengineering," by Cooper and Markus, in the *Sloan Management Review*, Summer 1995. In this article, Okuno works on instituting a positive attitude toward change. How does he do this? Discuss the importance of change acceptance by employees to the success of a technology implementation.
3. Refer to Minicase 1. You, as the CIO, believe that the business gains for implementing video mining in your retail stores will outweigh any negative customer perceptions. Your company is Baby's R Us, a child company of Toys R Us. Do an economic feasibility study for this investment. Be sure to include a listing of intangible costs and benefits, as well as an argument for your chosen discount rate. What is the ROI of the video mining? Try to keep your analysis to under 15 pages.
4. Develop a project plan and schedule feasibility study for the video-mining investment into Baby's R Us. Be sure to include a Gantt and PERT/CPM chart, as well as a clear discussion of all tasks that need to be completed.

Team and Individual Exercises

1. How often do you think legal issues play a role in project success? Think of an example of a potentially good information system or program that was constrained or not feasible due to legal requirements.
2. As a team, brainstorm some ways to enhance employee change acceptance of new information systems or business processes.
3. Think of an example when business process improvement is more appropriate than business process reengineering. Share with the class.

Suggested Readings

Application Development Trends (monthly periodical). Natick, MA: Software Productivity Group, a ULLO International company. This is our favorite systems development periodical. It follows systems analysis and design strategies, methodologies, CASE, and other relevant trends. Visit its Web site at www.adtmag.com.

Gause, Donald C., and Gerald M. Weinberg. *Are Your Lights On? How to Figure Out What the Problem REALLY Is.* New York: Dorset House Publishing, 1990. Here's a title that should really get you thinking, and the entire book addresses one of the least published aspects of systems analysis: problem solving.

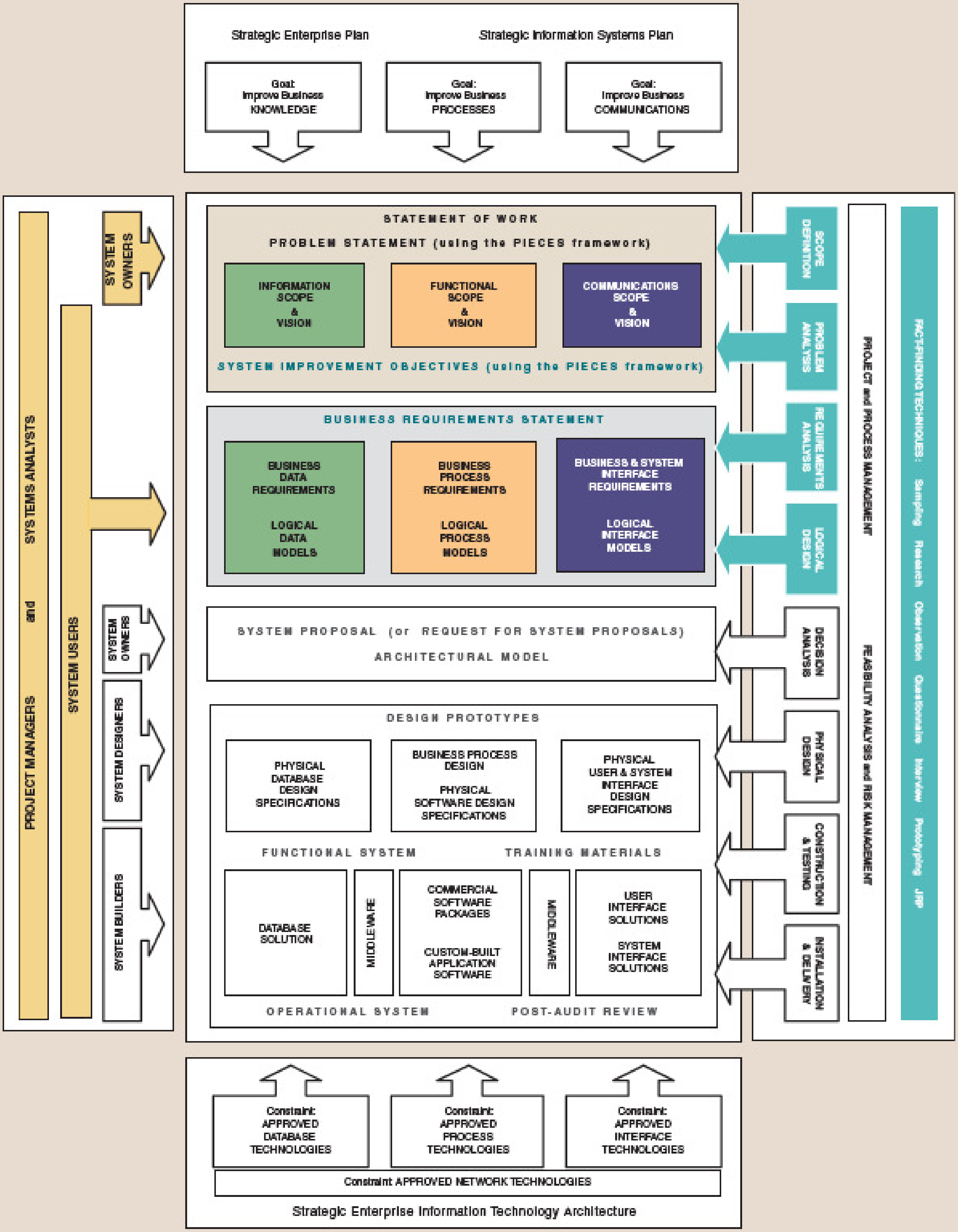
Hammer, Mike. "Reengineering Work: Don't Automate, Obliterate." *Harvard Business Review*, July–August 1990, pp. 104–11. Dr. Hammer is a noted expert on business process redesign. This seminal paper examines some classic cases where the technique dramatically added value to businesses.

Wetherbe, James. *Systems Analysis and Design: Best Practices*, 4th ed. St. Paul, MN: West Publishing, 1994. We are indebted to Dr. Wetherbe for the PIECES framework.

Wood, Jane, and Denise Silver. *Joint Application Design: How to Design Quality Systems in 40% Less Time.* New York: John Wiley & Sons, 1989. This book provides an excellent in-depth presentation of joint application development (JAD).

Yourdon, Edward. *Modern Structured Analysis.* Englewood Cliffs, NJ: Yourdon Press, 1989. This update to the classic DeMarco text on the same subject defines the current state of the practice for the structured analysis approach.

Zachman, John A. "A Framework for Information System Architecture." *IBM Systems Journal* 26, no. 3 (1987). This article presents a popular conceptual framework for information systems surveys and the development of an information architecture.



Strategic Enterprise Plan
Strategic Information Systems Plan
Goal: Improve Business KNOWLEDGE
Goal: Improve Business PROCESSES
Goal: Improve Business COMMUNICATIONS
PROJECT MANAGERS and SYSTEMS ANALYSTS
SYSTEM USERS
SYSTEM OWNERS
SYSTEM DESIGNERS
SYSTEM BUILDERS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
BUSINESS DATA REQUIREMENTS
LOGICAL DATA MODELS
BUSINESS PROCESS REQUIREMENTS
LOGICAL PROCESS MODELS
BUSINESS & SYSTEM INTERFACE REQUIREMENTS
LOGICAL INTERFACE MODELS
SYSTEM PROPOSAL (or REQUEST FOR SYSTEM PROPOSALS)
ARCHITECTURAL MODEL
DESIGN PROTOTYPES
PHYSICAL DATABASE DESIGN SPECIFICATIONS
BUSINESS PROCESS DESIGN
PHYSICAL SOFTWARE DESIGN SPECIFICATIONS
PHYSICAL USER & SYSTEM INTERFACE DESIGN SPECIFICATIONS
FUNCTIONAL SYSTEM
TRAINING MATERIALS
DATABASE SOLUTION
MIDDLEWARE
COMMERCIAL SOFTWARE PACKAGES
CUSTOM-BUILT APPLICATION SOFTWARE
MIDDLEWARE
USER INTERFACE SOLUTIONS
SYSTEM INTERFACE SOLUTIONS
OPERATIONAL SYSTEM
POST-AUDIT REVIEW
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
DECISION ANALYSIS
PHYSICAL DESIGN
CONSTRUCTION & TESTING
INSTALLATION & DELIVERY
PROJECT and PROCESS MANAGEMENT
FEASIBILITY ANALYSIS and RISK MANAGEMENT
FACT-FINDING TECHNIQUES: Sampling Research Observation Questionnaire Interview Prototyping JRP
Constraint: APPROVED DATABASE TECHNOLOGIES
Constraint: APPROVED PROCESS TECHNOLOGIES
Constraint: APPROVED INTERFACE TECHNOLOGIES
Constraint: APPROVED NETWORK TECHNOLOGIES
Strategic Enterprise Information Technology Architecture