7

Modeling System Requirements with Use Cases

Overview

Chapter 7 provides a comprehensive look at use case modeling as a technique for documenting system requirements. This is a very important chapter partly because, as Fred Brooks is quoted in this chapter, "No other part of the conceptual work is a difficult as establishing the detailed technical requirements." This chapter is also important because of the growing popularity of use case modeling, a tool that aids the communication between analysts and users in documenting system requirements.

The chapter introduces use-case modeling with definitions of actors, associations, and extends, uses, depends on, and inheritance relationships. Concrete steps are provided for identifying business actors and business use cases.

Both Use-Case Model Diagrams and Use-Case Narratives are discussed and constructed within the context of the SoundStage case study. A Use-Case Ranking and Priority Matrix is also explained and demonstrated. Finally, Chapter 7 provides a discussion of how use cases fit in with project management.

Chapter to Course Sequencing

This chapter should follow Chapter 6 in all but exceptional course scenarios and even when object-oriented analysis and design is not being taught. This chapter is the first "modeling" chapter and a bridge from the earlier general chapters into the techniques and tools of systems analysis. Adopters wanting to focus on traditional structured analysis would then move on to Chapter 8, Data Modeling and Analysis, and Chapter 9, Process Modeling. Adopters desiring to focus on object-oriented analysis and design would move on to Chapter 10, Object-Oriented Analysis and Modeling Using the UML. Or you could cover all approaches.

What's Different Here and Why?

The following changes have been made to this chapter in the seventh edition:
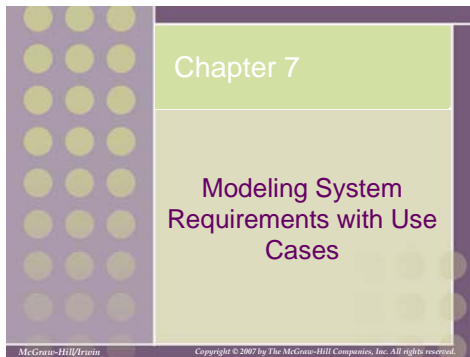
1. As with all chapters, we have streamlined the SoundStage episode into a quick narrative introduction to the concepts presented the chapter.

2. We have added a key term for "functional decomposition" and explained how use cases are a functional decomposition of the system as a whole.

3. We have expanded the discussion of abstract and extension use cases.

4. We have added new key terms for "depends on" and "inheritance."

5.   We have provided advice for naming actors and use cases.

6.   We have added a key term for "business requirements use case."

7.   We have expanded the discussion of several of the items on the use case narrative.

Lesson Planning Notes for Slides

The following instructor notes, keyed to slide images from the PowerPoint repository, are intended to help instructors integrate the slides into their individual lesson plans for this chapter.

Slide 1

Chapter 7

Modeling System Requirements with Use Cases

McGraw-Hill/Irwin   Copyright © 2007 by The McGraw-Hill Companies, Inc. All rights reserved.

slide appearance after initial mouse click
in slide show mode

This repository of slides is intended to support the named chapter. The slide repository should be used as follows:
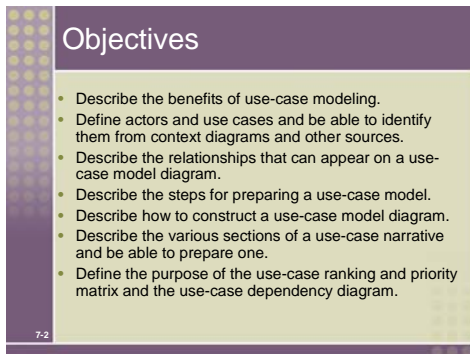Copy the file to a unique name for your course and unit.
Edit the file by deleting those slides you don't want to cover, editing other slides as appropriate to your course, and adding slides as desired.
Print the slides to produce transparency masters or print directly to film or present the slides using a computer image projector.
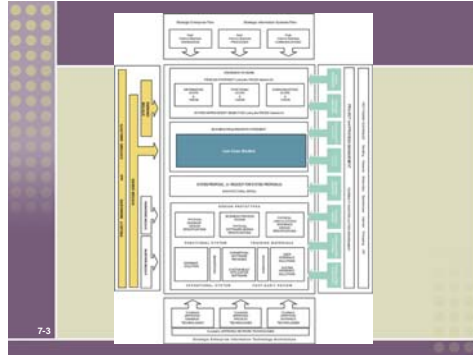
Most slides include instructor notes. In recent versions of PowerPoint, notes by default display in a window under the slide. The instructor notes are also reprinted below.

Slide 2

Objectives

- Describe the benefits of use-case modeling.
- Define actors and use cases and be able to identify them from context diagrams and other sources.
- Describe the relationships that can appear on a use-case model diagram.
- Describe the steps for preparing a use-case model.
- Describe how to construct a use-case model diagram.
- Describe the various sections of a use-case narrative and be able to prepare one.
- Define the purpose of the use-case ranking and priority matrix and the use-case dependency diagram.

7-2

Chapter 7 objectives.

Slide 3

Teaching Notes
This slide shows the how this chapter's content fits with the building blocks framework used throughout the textbook. Use Cases are first built in the Requirements Analysis phase and then used in the Logical Design phase. However, Use Cases should be refined and used again in the Physical Design phase and throughout the project. Use Case development involves system owners, system users, and systems analysts.
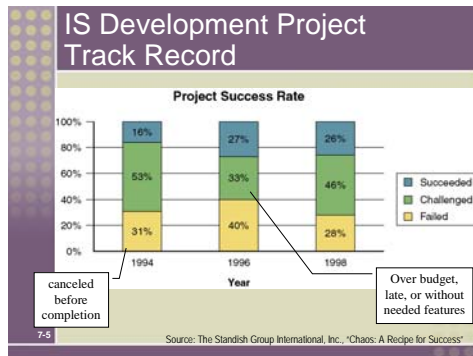
Slide 4

An Introduction to Use-Case Modeling

- One of the primary challenges is the ability to elicit the correct and necessary system requirements from the stakeholders and specify them in a manner understandable to them so those requirements can be verified and validated.

The hardest single part of building a software system is deciding precisely what to build. No other part of the conceptual work is a difficult as establishing the detailed technical requirements, including all the interfaces to people, to machines, and to other software systems. No other work so cripples the resulting system if done wrong. No other part is more difficult to rectify later.   Fred Brooks

7-4

Teaching Notes
This slide explains the industry problem that this material addresses.
If possible, the instructor should share real-life experiences in misunderstood or mis-specified system requirements.
To illustrate the inadequacy of data and process models, show the students some of the models from chapters 8 and 9. As them as novices if they can understand them.

Slide 5

IS Development Project Track Record

Teaching Notes
This slide illustrates both the spotty track record of information system development and the fact that the the track record has been showing some limited signs of improvement.

Slide 6

User-Centered Development and Use-Case Modeling

User-centered development – a process of systems development based on understanding the needs of the stakeholders and the reasons why the system should be developed.

Use-case modeling – the process of modeling a system's functions in terms of business events, who initiated the events, and how the system responds to those events.
- Use-case modeling has roots in object-oriented modeling.
- Gaining popularity in non-object development environments because of its usefulness in communicating with users.
- Compliments traditional modeling tools.

7-6

Teaching Notes
There is nothing about user-centered development that requires use cases. But employing use cases in requirements analysis is an excellent way to make sure your project begins as user-centered development.

Slide 7

Benefits of Use-Case Modeling

- Provides tool for capturing functional requirements.
- Assists in decomposing system into manageable pieces.
- Provides means of communicating with users/stakeholders concerning system functionality in language they understand.
- Provides means of identifying, assigning, tracking, controlling, and management system development activities.
- Provides aid in estimating project scope, effort, and schedule.

7-7

Teaching Notes
Using use-case modeling encourages user involvement.
By the same token, for use cases to be successful participation by the user is imperative.

Slide 8

Benefits of Use-Case Modeling (continued)

- Aids in defining test plans and test cases.
- Provides baseline for user documentation.
- Provides tool for requirements traceability.
- Provides starting point for identification of data objects or entities.
- Provides specifications for designing user and system interfaces.
- Provides means of defining database access requirements.
- Provides framework for driving the system development project.

7-8

Teaching Notes
Using use-case modeling encourages user involvement.
By the same token, for use cases to be successful participation by the user is imperative.

Slide 9

System Concepts for Use-Case Modeling

Use case – a behaviorally related sequence of steps (scenario), both automated and manual, for the purpose of completing a single business task.
- Description of system functions from the perspective of external users in terminology they understand.

Use-case diagram – a diagram that depicts the interactions between the system and external systems and users.
- graphically describes who will use the system and in what ways the user expects to interact with the system.

Use-case narrative – a textual description of the business event and how the user will interact with the system to accomplish the task.

7-9

Teaching Notes
use-case diagrams and use-case narratives are two views of the same sequence of steps that make up a conceptual use-case.
The use-case diagram communicates at a high level the scope of the business events that make up the Use-case.
The use-case narrative communicates at a more detailed level exactly how the user interacts with the system.
A use-case itself is not considered a functional requirement, but the use-case's story, or scenario, consists of one or more requirements.

Slide 10

Sample Use-Case Model Diagram

7-10
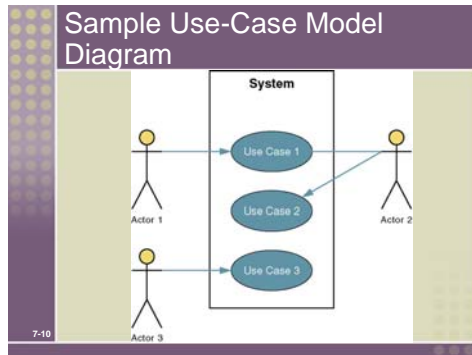
Teaching Notes
Definitions for these symbols are on the next slide.

Slide 11

Basic Use-Case Symbols

Use case – subset of the overall system functionality
- Represented by a horizontal ellipse with name of use case above, below, or inside the ellipse.

Use Case Symbol

Actor – anyone or anything that needs to interact with the system to exchange information.
- human, organization, another information system, external device, even time.

Actor Symbol

Temporal event – a system event triggered by time.
- The actor is time.

7-11

Teaching Notes
Use cases are the results of deconstructing the scope of system functionality into many smaller statements of system functionality.
Use cases describe the system functions from the perspective of external users and in the manner and terminology in which they understand.
An actor initiates system activity, a use case, for the purpose of completing some business task.
An actor represents a role fulfilled by a user interacting with the system and is not meant to portray a single individual or job title.
Have students provide examples of temporal events (nightly download, monthly billing, etc.).

Slide 12

Four Types of Actors

- **Primary business actor**
  - The stakeholder that primarily benefits from the execution of the use case.
  - e.g. the employee receiving the paycheck
- **Primary system actor**
  - The stakeholder that directly interfaces with the system to initiate or trigger the business or system event.
  - e.g. the bank teller entering deposit information
- **External server actor**
  - The stakeholder that responds to a request from the use case.
  - e.g. the credit bureau authorizing a credit card charge
- **External receiver actor**
  - The stakeholder that is not the primary actor but receives something of value from the use case.
  - e.g. the warehouse receiving a packing slip

7-12

Teaching Notes
Walk students through a couple of real-life information systems, identifying actors. For instance, paying for groceries with a cashier would have:
- Primary business actor - customer
- Primary system actor - cashier
- External server actor - if paying with a credit card, would be the credit card authorization company
- External receiver actor - the distributor who receives data about purchases and inventory

If the customer was checking himself/herself out at a U-Scan terminal, then the customer would also be the primary system actor.

Slide 13

Use Case Association Relationship

Association – a relationship between an actor and a use case in which an interaction occurs between them.
- Association modeled as a solid line connecting the actor and the use case.
- Association with an arrowhead touching the use case indicates that the use case was initiated by the actor. (1)
- Association lacking arrowhead indicates a receiver actor. (2)
- Associations may be bidirectional or unidirectional.

7-13 Club Member   Distribution Center

Teaching Notes
Most of the use case diagrams in this chapter shown only the primary system actor. This one also shows the external receiver actor. Use cases can be drawn to show some or all actors.

Slide 14

Use Case Extends Relationship

Extension use case –use case consisting of steps extracted from another use case to simplify the original.
- Extends the functionality of the original use case.
- Generally not identified in the requirements phase
- Extends relationship represented as arrow beginning at the extension use case and pointing to use case it is extending.
- Labeled <<extends>>.

7-14

Teaching Notes
An extension use case is employed to simplify another use case. An extension use case may have an extends relationship to one or more other use cases.

Slide 15

Teaching Notes
If your students have programming experience you can compare abstract use cases to programming code that is placed in public functions or classes. An abstract use case has a uses relationship to two or more other use cases.

Slide 16

Teaching Notes
Identifying use case dependencies is discussed at the end of the chapter. It is useful when using build cycles to build the system in successive versions.

Slide 17

Teaching Notes
Inheritance is depicted on the next slide.

Slide 18
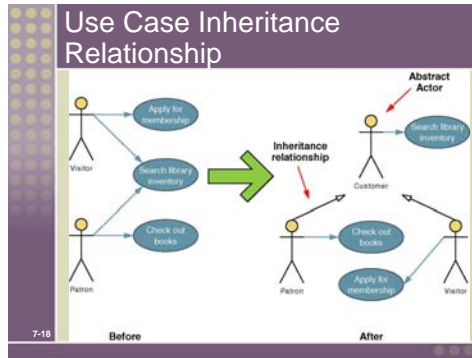
Teaching Notes
Walk through the Before and After of this figure.
Students should understand that though we have
added an actor, we have decreased the interac-
tions we have to model.

Slide 19

The Process of Requirements Use-Case Modeling

- Objective is to elicit and analyze enough requirements information to prepare a model that:
  – Communicates what is required from a user perspective.
  – Is free of specific details about how system will be implemented.
- To effectively estimate and schedule project, may need to include preliminary implementation assumptions.
- Steps
  1. Identify business actors.
  2. Identify business use cases.
  3. Construct use-case model diagram.
  4. Documents business requirements use-case narratives.

7-18

Teaching Notes
The individual steps will be discussed on the
following slides.

Slide 19

Step 1: identify Business Actors

- When looking for actors, ask the following questions:
  – Who or what provides inputs to the system?
  – Who or what receives outputs from the system?
  – Are interfaces required to other systems?
  – Are there events that are automatically triggered at a predetermined time?
  – Who will maintain information in the system?
- Actors should be named with a noun or noun phrase

7-19

Teaching Notes
By focusing first on actors, you concentrate on
how the the system will be used instead of how it
will be built.
Focusing on actors helps refine and further define
the scope and boundaries of the system.
Also, by first identifying actors you find candi-
dates to interview and observe so you can de-
velop and validate the use cases.

Slide 20

Sample List of Actors

Term	Synonym	Description
1. Potential member		An individual or corporation that submits a subscription order in order to join the club.
2. Club member	Member	An individual or corporation that has joined the club via an agreement.
3. Past member	Inactive member	A type of member that has fulfilled the agreement obligation but has not placed an order within the last six months but is still in good standing.
4. Marketing		Organization responsible for creating promotion and subscription programs and generating sales for the company.
5. Member services		Organization responsible for providing point of contact for SoundStage Entertainment customers in terms of agreements and orders.
6. Distribution center	Warehouse	Entity that houses and maintains SoundStage Entertainment product inventory and processes customer shipments and returns.
7. Accounts receivable		Organization responsible for processing customer payments and billing as well as maintaining customer account information.
8. Time		Actor concept responsible for triggering temporal events.

7-20

Teaching Notes
The can be called an actor glossary.

Slide 21

Step 2: Identify Business Requirements Use Cases

Business Requirements Use Case - a use case created during requirements analysis to capture the interactions between a user and the system free of technology and implementation details.

- During requirements analysis, strive to identify and document only the most critical, complex, and important use cases, often called *essential* use cases.

7-21

No additional notes.

Slide 22

Step 2: Identify Business Requirements Use Cases (cont.)

- When looking for use cases, ask the following questions:
  - What are the main tasks of the actor?
  - What information does the actor need form the system?
  - What information does the actor provide to the system?
  - Does the system need to inform the actor of any changes or events that have occurred?
  - Does the actor need to inform the system of any changes or events that have occurred?
- Use cases should be named with a verb phrase specifying the goal of the actor (i.e. Submit Subscription Order)

7-22

No additional notes.

Slide 23

Sample Context Diagram
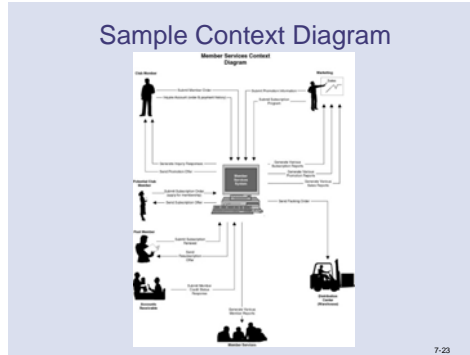
Conversion Notes
In the seventh edition we introduced context diagrams in chapter 5, so this may be review.
Teaching Notes
A context diagram is an excellent source for analyzing actors and finding potential use cases.
The primary inputs that trigger business events are considered use cases, and the external parties that provide those inputs are considered actors.
Individual reports are often not listed on a context diagram to reduce clutter. The systems analyst must research with the appropriate stakeholders the outputs they receive to uncover these "hidden use cases."
This is a different format than the context diagram used in chapter 5 or the context DFD from chapter 9, but the concepts are the same.

Slide 24

Sample Use-Case Glossary

No additional notes

Slide 25

Sample Use-Case Glossary (cont.)
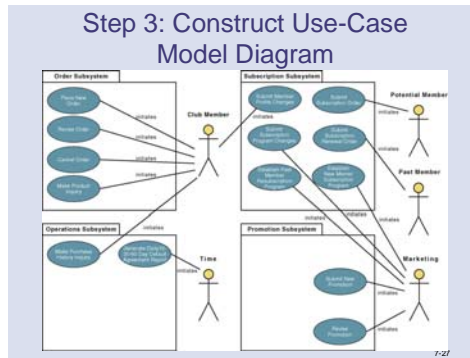

No additional notes

Slide 26

Sample Use-Case Glossary (cont.)

Establish New Member Subscription Program	This use case describes the event of the marketing department establishing a new membership subscription plan to entice new members.	• Marketing (primary business)
Submit Subscription Program Changes	This use case describes the event of the marketing department changing a subscription plan for club members (e.g., extending the fulfillment period).	• Marketing (primary business)
Establish Past Member Resubscription Program	This use case describes the event of the marketing department establishing a resubscription plan to lure back former members.	• Marketing (primary business)
Submit Member Profile Changes	This use case describes the event of the marketing department establishing a new promotion plan to entice active and inactive members to order the product. (Note: A promotion features specific titles, usually new, that the company is trying to sell at a special price. These promotions are interjected into a catalog sent (or communicated) to all members.)	• Marketing (primary business)
Revise Promotion	This use case describes the event of the marketing department revising a promotion.	• Marketing (primary business)
Generate Daily 15-30-60-Day Default Agreement Report	This use case describes the event of a report that is generated on a daily basis to list the members who have not fulfilled their agreement by purchasing the required number of products outlined when they subscribed. This report is sorted by members who are 15 days past due, 30 days past due, and 60 days past due.	• Time (initiating actor) • Member Services (primary* —external receiver)

concluded

7-26

Slide 27

Step 3: Construct Use-Case Model Diagram

7-27

Teaching Notes
Note that the use cases have been grouped into business sub-systems. This is key to defining your development strategy – which use cases will be developed first and by whom.

Slide 28

Step 4: Document Business Requirements Use-Case Narratives

• Document first at high level to quickly obtain an understanding of the events and magnitude of the system.
• Then expand to a fully-documented business requirement narrative.
   – Include the use case's typical course of events and its alternate courses.

7-28

No additional notes.

No additional notes

Slide 29

**Sample High-Level Version
of a Use-Case Narrative**

7-29

Teaching Notes
Author – the persons who wrote the use case and
provide a point of contact for anyone requiring
additional information.
Date – the date the use case was last modified.
Version – the current version of the use case.
Use-case name – the use-case name should
represent the goal that the use case is trying to
accomplish. Should begin with a verb.
Use-case type – Business requirements use
cases provide a general understanding of the
problem domain and scope but don't include
detail to communicate to developers what the
system should do.
Use-case ID – A unique identifier for the use
case.
Priority – The priority communicates the impor-
tance of the use case (high, medium, or low).
Source – The source defines the entity that trig-
gered the creation of the use case.
Primary business actor – The stakeholder that
primarily benefits from the execution of the use
case.
Other participating actors – Other actors that
participate in the use case.
Interested stakeholders – A person (other than
the actor) who has a vested interest in the goal of
the use case.
Description – A short summary description of the
purpose of the use case and its activities.

Slide 30

**Sample Expanded Version
of a Use-Case Narrative**

continued

7-30

Teaching Notes
Precondition – A constraint on the state of the
system before the use case can be executed.
Trigger – The event that initiates the use case.

Slide 31

Sample Expanded Version of a Use-Case Narrative (cont)

continued

7-31

Teaching Notes
Typical course of events – The normal sequence of activities performed by the actor(s) and the system to satisfy the goal of the use case.

Slide 32

Sample Expanded Version of a Use-Case Narrative (cont)

concluded

7-32

Teaching Notes
Alternate courses – The behaviors of the use case if an exception or variation to the typical course occurs.
Conclusion – When the use case successfully ends.
Postcondition – A constraint on the state of the system after the use case has successfully executed.
Business rules – Policies and procedures of the business that the system must abide by.
Implementation constraints and specifications – Any nonfunctional requirements that may impact the realization of the use case.
Assumptions – Assumptions made by the author.
Open issues – Issues that need to be resolved before the use case can be finalized.

Slide 33

Use Cases and Project Management

- Use-case model can drive entire development effort.
- Project manager or systems analyst uses business requirements use cases to estimate and schedule the build cycles of the project.
  - Build cycles are scoped on the basis of the importance of the use case and the time it takes to implement the use case.
- To determine importance of use cases, will create:
  - Use-case ranking and evaluation matrix
  - Use-case dependency diagram

7-33

Teaching Notes
You can relate this to the concept of timeboxing discussed in chapter 5.

Slide 34

Use-Case Ranking and Priority Matrix

- In most projects, the most important use cases are developed first.

  Use-case ranking and priority matrix – a tool used to evaluate use cases and determine their priority.

  – Evaluates use cases on 1-5 scale against six criteria.
    1. Significant impact on the architectural design.
    2. Easy to implement but contains significant functionality.
    3. Includes risky, time-critical, or complex functions.
    4. Involves significant research or new or risky technology.
    5. Includes primary business functions.
    6. Will increase revenue or decrease costs.

7-34

Teaching Notes
A sample matrix is shown on the next slide.

Slide 35

Sample Use-Case Ranking and Priority Matrix

Use-Case Name	Ranking Criteria, 1 to 5						Total Score	Priority	Build Cycle
	1	2	3	4	5	6			
Submit Subscription Order	5	5	5	4	5	5	29	High	1
Place New Order	4	4	5	4	5	5	27	High	2
Make Product Inquiry	1	1	1	1	1	1	6	Low	3
Establish New Member Subscription Program	4	5	5	3	5	5	27	High	1
Generate Daily 10-30-60-Day Default Agreement Report	1	1	1	1	1	1	6	Low	3
Revise Order	2	2	3	3	4	4	18	Medium	2

7-35

Teaching Notes
The ranking criteria are explained on the previous slide.

Slide 36

Use-Case Dependency Diagram

Use-case dependency diagram – graphical depiction of the dependencies among use cases.

- – Provides the following benefits:
  - Graphical depiction of the system's events and their states enhances understanding of system functionality.
  - Helps identify missing use cases.
  - Helps facilitate project management by depicting which use cases are more critical.

7-36

Teaching Notes
A sample dependency diagram is shown on the next slide.

Slide 36

Sample Use-Case
Dependency Diagram
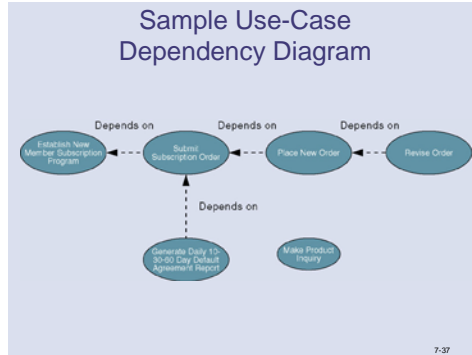
No additional notes.

Answers to End of Chapter Questions and Exercises

Review Questions

1. User-centered development is a process of systems development based on understanding the needs of the stakeholders and the reasons why the system should be developed. We need it because the process will help us to successfully plan, analyze, design, construct, and deploy an information system. The systems analyst must first understand the needs of the stakeholders and the reasons why the system should be developed.

2. Use-case modeling is the process of modeling a system's functions in terms of business events, who initiated the events, and how the system responds to those events. Use-case modeling is an approach that facilities usage-centered development by encouraging user involvement, which is a major factor in successful projects.

3. a. Provides a tool for capturing functional requirements
   b. Assist in decomposing system scope into more manageable pieces
   c. Provides a means of communicating with users and other stakeholders concerning system functionality, and use cases present a common language that is easily understood by various stakeholders
   d. Provides a means of identifying, assigning, tracking, controlling, and managing system development activities, especially incremental and iterative development
   e. Provides an aid in estimating project scope, effort, and schedule
   f. Provides a baseline for testing in terms of defining test plans and test cases
   g. Provides a baseline for user help systems manuals as well as system development documentation
   h. Provides a tool for requirements traceability
   i. Provides a starting point for the identification of data objects or entities
   j. Provides functional specifications for designing user and system interfaces
   k. Provides a means of defining database access requirements in terms of adds, changes, deletes and reads
   l. Provides a framework for driving the system development project

4. The use-case diagram depicts the interactions between the system and external systems and users. Its purpose is to communicate at a high level the business events that must be processed by the system by providing a graphic description of who will use the system and in what ways the user expects to interact with the system.

The use-case narrative is a textual description of each business event, the actors (users) involved in the business event, their roles and how they will interact with the system to accomplish the task.

5. 1) Use cases, which identify and describe the system functions from the perspective of external users.
   2) Actors, which can be human users, an organization, another information system, an external device or temporal event, who initiate system activity in order to complete a business task.
   3) Relationships, which depict the type of interactions between use cases and/or actors.

6. Use cases are initially defined during the requirements stages of the life cycle and will be additionally refined throughout the life cycle.
   During requirements discovery, systems analysts employ use cases to capture the essence of the business problems and to model the functionality of the proposed system at a high level. During requirements analysis, the use cases are refined to model usage of the system in more detail. During design, the use cases are refined to model how the users will actually use the system with regard to any interface and system constraints. During construction, use cases aid the system builders in programming and testing, as well serving as the source material for user and system documentation.

7. The primary system actor is the stakeholder who receives the major benefit upon execution of the use case. The primary actor can but doesn't have to be the one to directly initiate or trigger the business or event.

8. The five types of relationships are: Associations, Extends, Uses (or Includes), Depends on, and Inheritance.

   1) An <u>association</u> relationship shows the interaction relationship between an actor and a use case.
   2) An <u>extends relationship shows the relationship between a use case and the extension use case the original use case is invoking.</u>
   3) A <u>Uses relationship shows the relationship between a use case and the abstract use case in which the common steps used by multiple use cases has been extracted in order to reduce redundancy</u>
   4) A <u>Depends On relationship shows the relationship between a use case and its dependency relationship to another use case.</u>
   5) A <u>Inheritance relationship shows the relationship between two or more actors which initiate the same use and an abstract actor <u>created in order to reduce redundant communication with the system</u></u>

9. 1) Identify the business actors
   2) Identify business requirements use-cases.

3) Construct the use-case model diagram.
4) Document the business requirements use-case narratives.

10. When we identify actors first, we can concentrate on how the system will be used and not how it will be built. Also, it can help identify the candidates to later interview and/or observe to complete the use-case modeling.

11. The following questions need to be asked when identifying business requirements use-cases:
- What are the main tasks of the actor?
- What is the information the actor needs from the system?
- What is the information the actor provides to the system?
- Does the system need to inform the actor of any changes or events that have occurred?
- Does the actor need to inform the system of any changes or events that have occurred?

12. It is a step-by-step description of the normal sequence of activities performed by the actor(s) and the system in order to satisfy the goal of the use-case, starting with the actor initiating the use case and continuing until the end of the business event

13. A build cycle, which consists of the system analysis, design, and construction activities, is scoped and scheduled on the basis of the importance of the use case and the time it takes to implement the use case. In other words, one or more use cases will be developed in each build cycle. For a use case that is too large or complex to be completed in one build cycle, a simplified version will be implemented initially, followed by the full version in the next build cycle. In order to do that, it is essential to rank and evaluate the importance of use cases.

14. 1) Significant impact on the architectural design
2) Easy to implement but contains significant functionality
3) Includes risky, time-critical, or complex functions
4) Involves significant research or new or risky technology
5) Includes primary business functions
6) Will increase revenue or decrease costs

15. The use-case dependency diagram is a graphical depiction of the dependencies among use cases. Modeling these dependencies provides a number of benefits:

1) The graphical depiction of the system's events and their states enhances the understanding of system functionality.
2) It helps to identify missing use cases.

3) It helps facilitate project management by depicting which use cases are more critical because they have the most dependencies and thus need to have a higher priority.

Problems and Exercises

1. The most difficult task in systems development is determining what to build, i.e., what the requirements specifications are. User-center development addresses this challenge by having systems analysts focus on the needs of the stakeholders and the business issues driving the project. Use-case modeling is a methodology based upon user-centered development, and helps identify system requirements from the perspectives of users, owners and other stakeholders.

2. The main artifacts employed in use case modeling are the use case diagram and the use case narrative.

   The use case diagram illustrates the system as a related collection of use cases, users (who are called actors) and the interactions or relationships between them. The purpose of the use case diagram is to show at a high level the scope of the system and the business events the system processes.

   The use case narrative describes in non-technical language a business event and the interactions that must occur between the user(s) and the system for that business event to be processed.

3. The systems analyst should always remember that the purpose of a use case is to describe a system function using language and diagrams that non-technical users can understand.
   Although requirements discovery has been completed, the systems analyst needs to remember that identifying and developing use cases accurately and completely requires a significant amount of involvement from users and subject matter experts who understand the business processes.
   A use case depicts a single business task or goal, and the sequence of events and interactions needed to accomplish that task or goal.
   A use case is not a functional requirement, but one or more functional requirements are contained in the series of steps depicted in the use case.

4. Use cases are first identified and defined during the requirements phase and are generally used and refined throughout each phase of the development life cycle.

In the requirements discovery phase, use cases depict and document the business problems, as well as modeling at a high level the desired functionality of the new system. Later on in the requirements phase, they are refined in order to help to identify the system data entities and to model system usage at a lower more detailed level.

During the design phase, use cases are further refined to show the system user interfaces and system constraints, in order to assist the developers in writing the interface and code specifications. They also can be used to help develop the test plan and test scripts.

During the construct phase, use cases help the developers who are coding and unit testing the system, as well as helping in documenting the system.

During implementation, the use cases also help in preparing user guides and in training users.

5.

Stakeholder	Actor
• United States Postal Service	External receiver actor
• Computerized door lock with key pad	Primary system actor
• Rental car agent	Primary system actor
• Sales manager generating regional sales report	External receiver actor
• Sales manager receiving regional sales report	Primary business actor
• Automatic lawn sprinkler system	Time
• Driver purchasing gasoline with ATM card	Primary system actor <u>and</u> Primary business actor
• Bank loan authorization service	External server actor

6. • The relationship between the use case "Print Form" and several other use cases that involve printing different types of forms? **Includes relationship**
   • The relationship between a motorcycle officer and a handheld citation writing device? **Association relationship**
   • The relationship between a customer and a sales clerk who can each query the inventory system to see if an item is in stock, and an actor created specifically to minimize duplicative system communication? **Inheritance relationship**
   • The relationship between the use case "Calculate GPA" and the lengthy use case "Create Transcript?" **Extends relationship**
   • The relationship between the use case "Ship Order" and the use case "Submit Order?" **Depends on relationship**

7. 1. Online shoppers and browsers   4. Credit card authorization service
    2. Customers of Y&J Cookbooks   5. Shipping company
    3. Y&J Cookbook owner/operator

8. The systems analyst should define the actors from the perspective of system
    users and in non-technical language.

    A typical Actor Glossary for an online system such as Y&J Cookbooks is
    shown below:

Term	Synonym	Description
Visitor	Shopper	Someone who browses the website, but doesn't yet have an account to purchase books
Customer		An individual or business who has registered and has an account, shops on the website and purchase books from the site.
Owner/operator	Owner/operator	Interacts with buyers via the website, updates/replenishes inventory of books; and prepares orders for shipping
Credit Card Validation Service		Third-party entity responsible for validating and processing online credit card transactions and payments
Shipping		Shipping company which picks up the order from Y&J Cookbooks, and delivers it to the customer

9. The context diagram for an online system such as Y&J Cookbooks should look something like this:

Use Case Context Diagram
for
Y&J Cookbooks Website

Web Shopper/Customer

Credit Card
Validation
Service

Shipping

Y&J Cookbooks Website

Inventory

Y&J Cookbooks
Owner/Operator
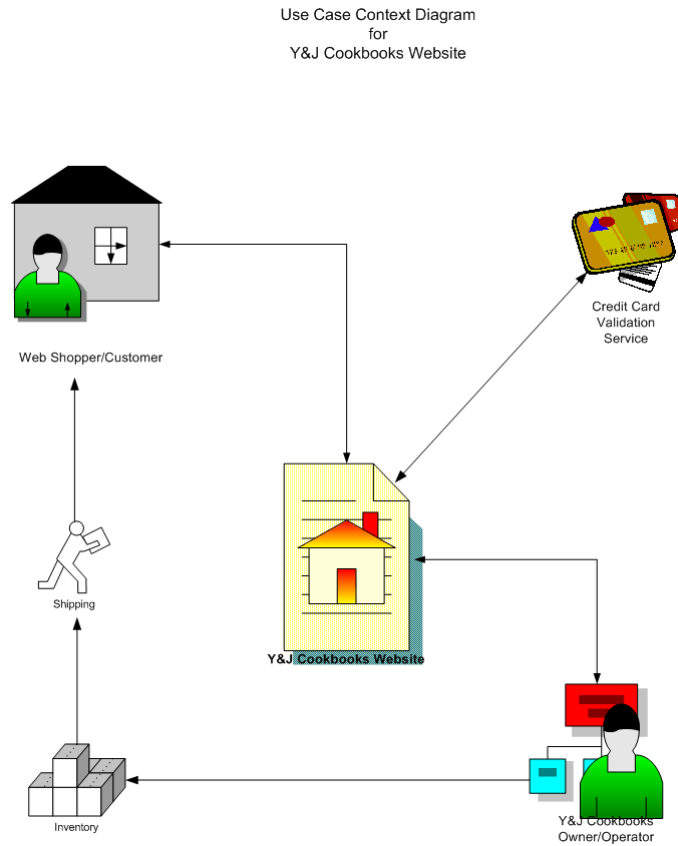
10. Each use case should capture the related series of steps (manual as well as automated) that are necessary to complete a single business task.

One technique that systems analysts can use to identify use cases is to look at how actors interact with the system.

Questions to ask when trying to identify the different use cases include:

- What does each actor do, i.e., what are their major tasks?
- What does each actor want or need from the system, i.e., what information, services or products?
- What information does each actor need to give to the system?
- What must the system tell or give to each actor regarding any events or changes that have taken place, and vice versa?

For most systems, there is not sufficient time or resources to identify and document every single possible use case. Thus, the difference between a use case and an essential case is that the essential use cases are those

determined to be the ones which are the most complex and/or of the great-
est criticality.

11. The use case diagram should look something like this:
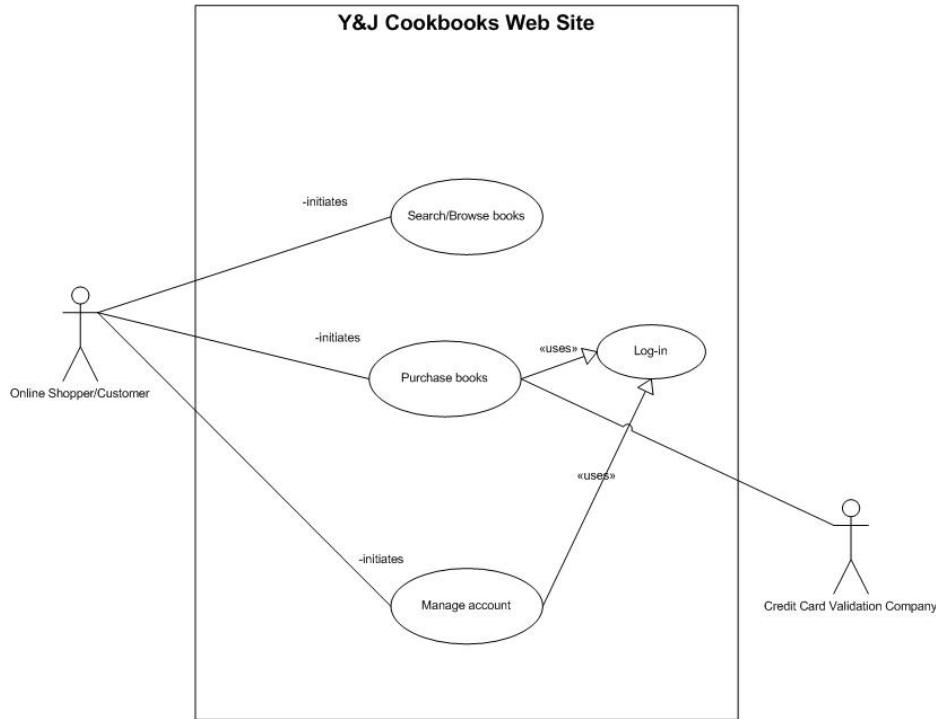
12. The systems analyst first creates high-level use case narratives in order to
provide a picture of the entire system, i.e., its scope, boundaries and the
events which define it. The analyst should then create a fully documented
expanded business requirement narrative to include each step in the typi-
cal course of events, as well as alternative courses of events. The expanded
narrative should look something like this:

USE CASE NAME:	High-Level	USE CASE TYPE
USE CASE ID:	1.0	Business Require-ments: ☐
PRIORITY:	High	System Analysis: ☐
SOURCE:	Online visitor to website	System Design: ☑
PRIMARY BUSI-NESS ACTOR	• Online customer	
OTHER ACTORS:	• Credit card verification service	
OTHER STAKE-HOLDERS:	• Website owner/operator (order fulfillment, ac-counts receivable)	

DESCRIPTION:	The purpose of this website is to provide a wide variety of cookbooks for online purchase. This use case is a high-level diagram describing each of the major actors and system functions that would be experienced in a typical visit to the site.	
PRE-CONDITION:	Website must be up and running	
TRIGGER:	Trigger is initiated when online shopper goes to home page	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
	Step 1: The customer chooses a sub page from selection panel: Search, Browse Categories, Account Info, About Us, FAQ	**Step 2**: System responds by taking customer to selected page.
	Step 3: In a typical scenario, shopper will first choose to browse the book categories	**Step 4:** System will take shopper to the requested category, and if selected, to book details
	Step 5: Shopper will add book to shopping cart	**Step 6:** System updates the shopping cart and displays contents to shopper
	Step 7: After completing book selections, shopper goes to checkout	**Step 8:** System goes to log-on page if shopper hasn't logged in yet.
		Step 9: System collects shipping information from shopper
		Step 10: System collects payment information from shopper and validates.
		Step 11: System displays order summary and requests confirmation from shopper
		Step 12: System displays final order summary and processes order
ALTERNATE COURSES:	Alternate courses may result from any of the following:	
	Attempting to checkout without logging in first will result in an error message, and will route the user to the log-in screen.	

	Entering a credit card and/or amount that is declined will result in a notification message
	Attempting to log in without the correct ID or password will result in an error message
	Attempting to purchase a book that is out of stock or in a quantity that exceeds the number in inventory will result in an error message
	Entering invalid address or personal information may result in an error message
CONCLUSION:	Successful conclusion is indicated by an "order confirmation" message
POST-CONDITION:	None
BUSINESS RULES	• See "Business Overview"
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• Supports Internet Explorer; may not fully support other browsers • Web hosting service and back-end database may not be handle high volumes of concurrent traffic and transactions
ASSUMPTIONS:	• Website owner/operator is manually keeping inventory current and accurate
OPEN ISSUES:	• See "Limitations and Constraints" in Business Overview

13. The use case model is used throughout the entire systems development life cycle. This means that once the business requirements use case model is completed, it can be used to estimate the resources and time that will be needed in the construction phases. If an iterative and/or incremental approach is being used, each build cycle can be scheduled based upon the criticality of each use case and the estimated time it will take to build.

In general, use cases are built in order of importance, based upon input from both the developers and the stakeholders. The tool commonly used is called a use case ranking and priority matrix, using the following criteria:

- Impact upon the technical architecture of the system
- Ratio of functionality to ease of implementation, i.e., use cases that are easy to build and which contain a high degree of functionality would be ranked higher.
- Has a high level of complexity or degree of risk
- Involves untried or unfamiliar technology
- Primary business functions are included in the use case
- Impact upon bottom line, i.e., impact upon revenue and costs

Identifying use case dependencies is a final step in determining the order of building. It is necessary because one use case may be the precondition for another use case to occur, e.g., the use case "Make Reservation" is a precondition for the use case "Cancel Reservation." A use case dependency diagram is used to model dependencies, and shows dependent relationships between different use cases.

Project and Research

1. a. The number will vary based upon the search engine and the search term used, but it will typically be in excess of 200,000 "hits."

   b. Response should indicate the student's understanding by drawing upon and integrating the information presented in this chapter and previous chapters regarding the importance and criticality of the requirements phase.

      Article: The name of the article was "No Silver Bullet: Essence and Accidents of Software Engineering." The theme of the article was that there are no magical solutions to the inherent difficulties of project management and software development, but "a disciplined, consistent effort..." will yield significant improvement.

   c. The book is The Mythical Man-Month, which was first published in 1975 and reprinted many times since. Its theme is focused around the human elements of software engineering, and their impact upon project management and systems development.

   d. Open-ended as to response, but should be logical, persuasive and indicate that the student is familiar with and understands Brooks' reoccurring major themes, and is able to synthesize these themes with changes in information technology.

2. a. A project succeeded if it was delivered on time, on budget, with required features and functions. A project was challenged if it was late, over budget and/or with less than the required features and functions. A project failed it was cancelled prior to completion, or delivered and never used.

   b. Twenty-nine percent of projects succeeded, 53% were challenged, and 18% failed. (2004 Third Quarter Research Report)

   c. Student should be able to accurately compare the latest rates to Figure 7.1., E.g., there has been a small increase in success rates since 1998, but given success rate variations since 1994, unable to determine if the

improvement represents a trend; project failure rates have dropped by about one third since 1998, and there appears to be a strong downward trend in failure rates since 1996.

   d./e. Responses to Questions 2d and 2e are open-ended, but should be thoughtful, logical, and indicate an understanding of the issues involved.

3. Response should be consistent with the use case modeling processes and techniques described in the textbook, and should demonstrate that student understands basic methods and principles.

4. Response should be consistent with the use case modeling processes and techniques described in the textbook, and should demonstrate that student understands basic methods and principles.

5. Bibliographies should be accurate, and the abstracts should reflect the articles on which they are based. Responses to Question 5c should indicate that student understands the underlying principles of use case modeling, and is able to compare different methodologies.

6. This research exercise is designed to help advance the student's interviewing, fact-finding skills and analytical skills. Responses to 6a and 6b are open-ended, but should indicate that a genuine attempt was made to find a range of developers. Likewise, responses to 6c – 6f should indicate that the student thought through the interview process in advance, and were able to elicit the needed information. Responses to 6g are intended to reinforce the knowledge regarding CMM, which were introduced in a prior chapter. The responses should be consistent with the maturity levels described in the book, and should focus on multiple characteristics of the organization rather than a single characteristic in determining the maturity level. Responses to 6h and 6i are open-ended, but should be logical, consistent and thoughtful.

Minicases

1. Note to Professor: expect some variation in findings. However, most class registration will have similar functionality (eg. Add, drop, and look up classes).

   Note to Professor: Students should determine that there is a need to login to the system, look up information, add a class to their schedule, and etc. There will be some variation across students, but there should not be much.

2. Note to Professor: If your class seems a little shaky with this, have them choose the specific case of logging in. Then walk them through the process:

   a. first sketch out the steps of what usually happens when they log into a system.
   b. who is the person handling the login process? – that is your actor
   c. are there any assumptions we have to make about the person logging in? It would probably help if they are already a student at the school….
   d. what actions are we going to have made this actor do before they login? (preconditions) How about 'create an account?'
   e. what is the state once the actor gets logged in? i.e. what is the post-condition? The actor is now 'logged in' and able to register…
   f. have the student look at their quick drafts of answers a-e and the example on pg. 284. Their answers (in a rough format) dump right into the table… then they just revise and add a little complexity.

3. The students may not see the connection, but they already interviewed potential actors and already determined the needed system functionality for each actor. This should correspond pretty closely with the actual actors and the use cases they initiate. Examples are: Student: add a class, log into system, drop a class, update information, etc.

4. Note to Professor: refer to pages 274-275 for a quick reference on <<uses>>, <<depends on>>, and <<extends>>. Your students may end up using one or all three, depending on how they structure their Use Cases. It is ok for the Use Cases to differ.

Team and Individual Exercises

There are no right or wrong answers to these exercises.