Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS
（use case/activity/swimlane/）

9.1 REQUIREMENTS ANALYSIS

The requirements modeling action results in one or more of the following types of models:

- **Scenario-based models of requirements from the point of view of various system "actors."**
- **Class-oriented models or Data models that depict the information domain for the problem and represent object-oriented classes (attributes and operations) and the manner in which classes collaborate to achieve system requirements.**
- **Behavioral models that depict how the software behaves as a consequence of external "events."**
- **Function models or Flow-oriented models that represent the functional elements of the system and how they transform data as they move through the system.**

Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS
（use case/activity/swimlane）

In this chapter, we focus on scenario-based modeling —a technique that is growing increasingly popular throughout the software engineering community. In Chapters 10 and 11 we consider class-based models and behavioral models.Over the past decade, flow and data modeling have become less commonly used, while scenario and class-based methods, supplemented with behavioral approaches and pattern-based techniques have grown in popularity.

9.1.1 Overall Objectives and Philosophy

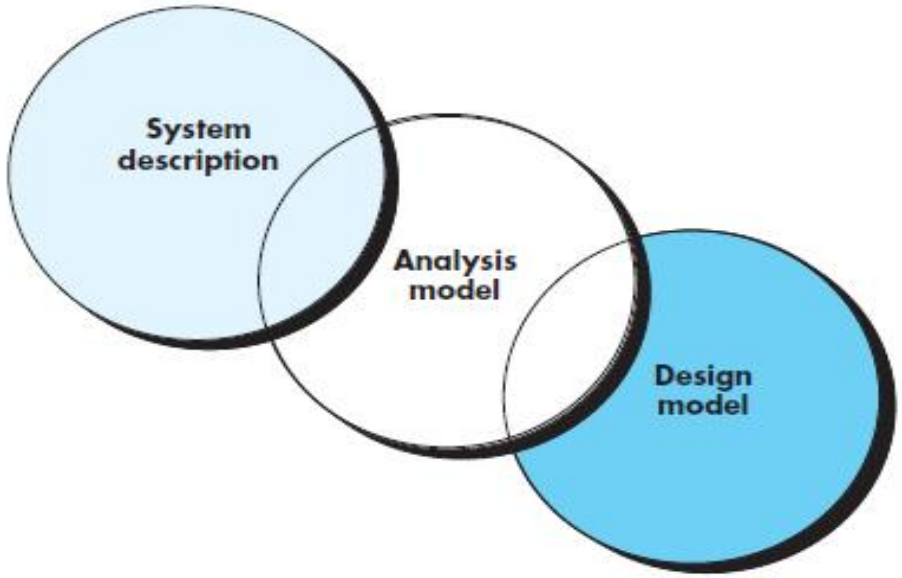
The requirements model must achieve three primary objectives: (1) to describe what the customer requires, (2) to establish a basis for the creation of a software design, and (3) to define a set of requirements that can be validated once the software is built.

Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS （use case/activity/swimlane）

It is important to note that all elements of the requirements model will be directly traceable to parts of the design model. A clear division of analysis and design tasks between these two important modeling activities is not always possible. Some design invariably occurs as part of analysis, and some analysis will be conducted during design.

FIGURE 9.1

The require-ments model as a bridge between the system description and the design model



Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS （use case/activity/swimlane）

9.1.2 Analysis Rules of Thumb

Arlow and Neustadt suggest a number of worthwhile rules of thumb that should be followed when creating the analysis model:

- **The model should focus on requirements that are visible within the problem or business domain. The level of abstraction should be relatively high.**
- **Each element of the requirements model should add to an overall understanding of software requirements and provide insight into the information domain, function, and behavior of the system.**
- **Delay consideration of infrastructure and other nonfunctional models until design.**
- **Minimize coupling throughout the system.**

Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS （use case/activity/swimlane）

- Be certain that the requirements model provides value to all stakeholders.
- Keep the model as simple as it can be.

9.1.3 Domain Analysis

The goal of domain analysis is straightforward: to find or create those analysis classes and/or analysis patterns that are broadly applicable so that they may be reused.

domain analysis may be viewed as an umbrella activity for the software process. By this we mean that domain analysis is an ongoing software engineering activity that is not connected to any one software project.

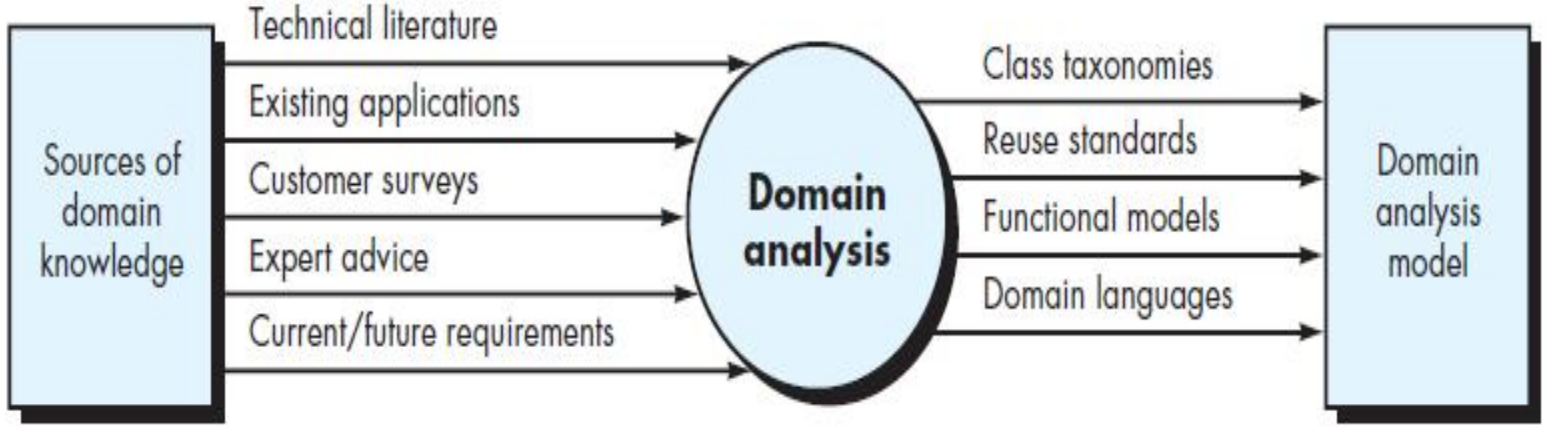
The role of the domain analyst is to discover and define analysis patterns, analysis classes, and related information that may be

Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS （use case/activity/swimlane）

used by many people working on similar but not necessarily the same applications.

Please refer to Appendix1.

FIGURE 9.2 Input and output for domain analysis



Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS （use case/activity/swimlane）

9.1.4 Requirements Modeling Approaches

- One view of requirements modeling, called structured analysis, considers data and the processes that transform the data as separate entities(new less used).
- A second approach to analysis modeling, called object-oriented analysis,focuses on the definition of classes and the manner in which they collaborate with one another to effect customer requirements. UML and the Unified Process are predominantly object oriented.
- In this edition of the book, we have chosen to emphasize elements of object oriented analysis as it is modeled using UML. Our goal is to suggest a combination of representations will

Chapter 9 REQUIREMENTS MODELING:SCENARIOBASED METHODS （use case/activity/swimlane）

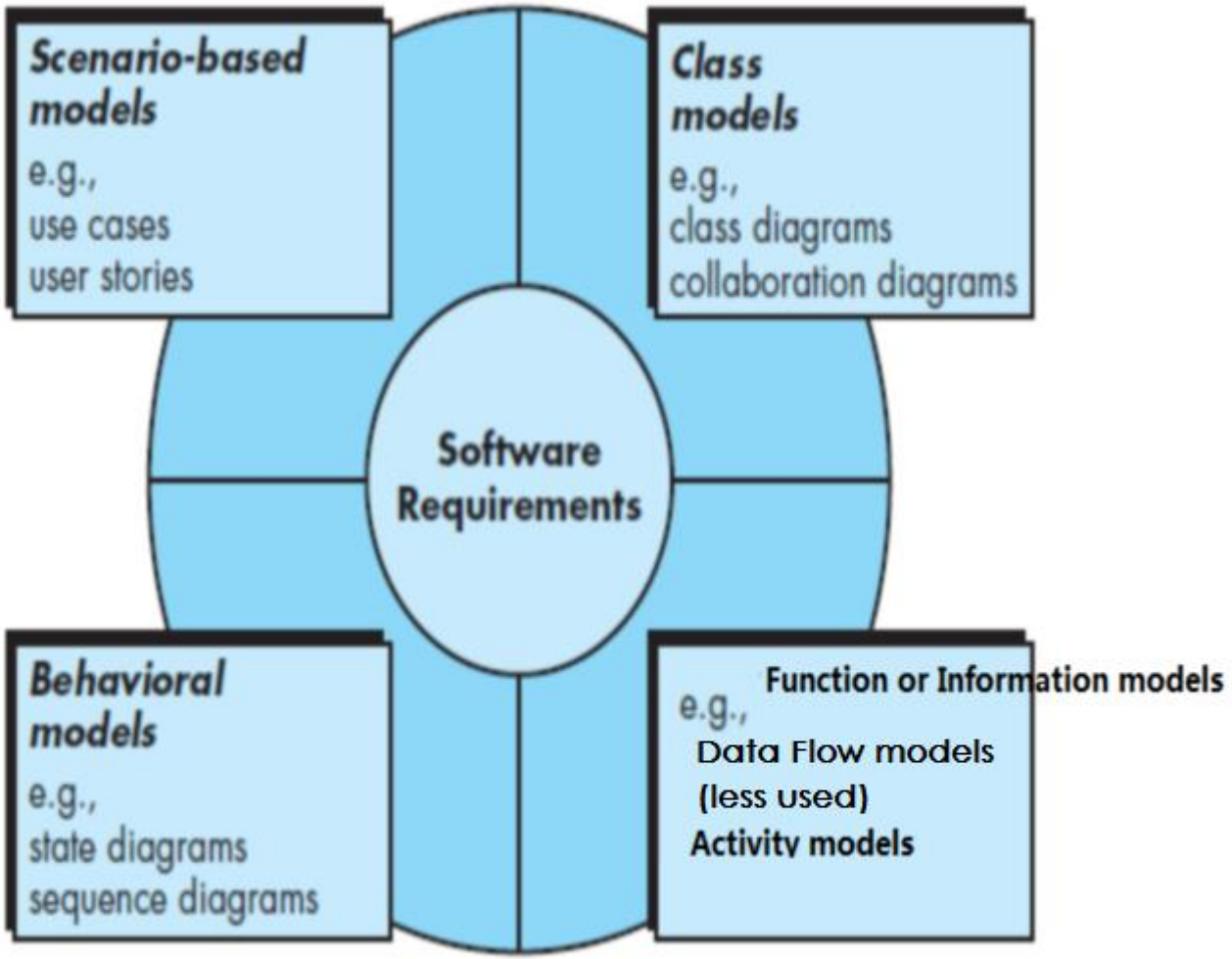
provide stakeholders with the best model of software requirements and the most effective bridge to software design.

Each element of the requirements model (Figure 9.3) presents the problem from a different point of view. Scenario-based elements depict how the user interacts with the system and the specific sequence of activities that occur as the software is used. Class-based elements model the objects that the system will manipulate, the operations that will be applied to the objects to effect the manipulation, relationships (some hierarchical) between the objects, and the collaborations that occur between the classes that are defined. Behavioral elements depict how external events change the state of the system or the classes that reside within it. Finally, flow-oriented elements represent the system as an information transform, depicting how data objects are transformed as they flow through various system functions.

Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS （use case/activity/swimlane）

FIGURE 9.3

Elements of the analysis model



Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS （use case/activity/swimlane）

9.2 SCENARIO-BASED MODELING

Requirements modeling with UML begins with the creation of scenarios in the form of use cases, activity diagrams, and swimlane diagrams.

9.2.1 Creating a Preliminary Use Case

9.2.2 Refining a Preliminary Use Case

9.2.3 Writing a Formal Use Case

Combine section 8.3.3 and the example of Library Management Systemin in courseware of object-oriented analysis and design to analyse 9.2.1,9.2.2 and 9.2.3.

9.3 UML MODELS THAT SUPPLEMENT THE USE CASE

9.3.1 Developing an Activity Diagram

Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS （use case/activity/swimlane）

The UML activity diagram supplements the use case by providing a graphical representation of the flow of interaction within a specific scenario.

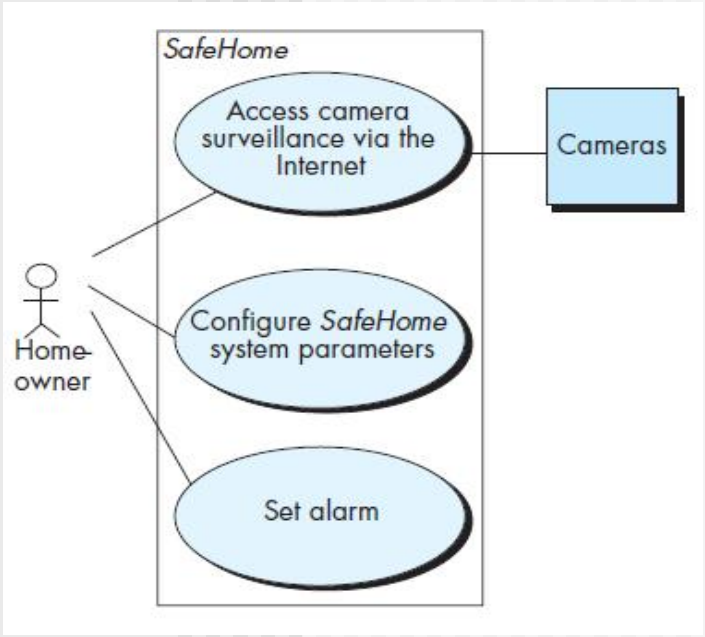
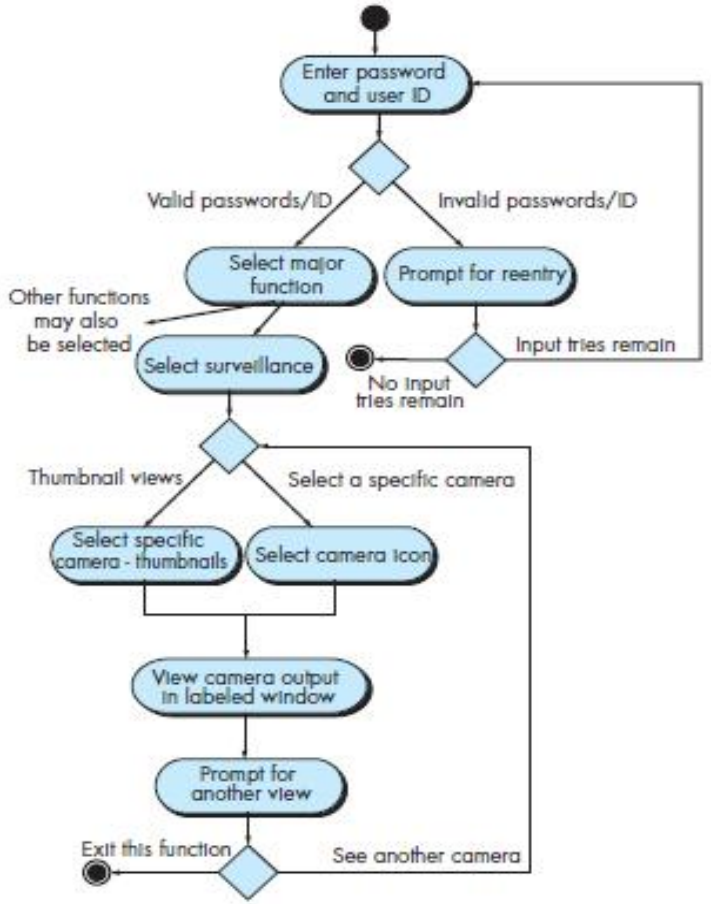


FIGURE 9.5
Activity diagram for Access camera surveillance via the Internet—display camera views function.



Chapter 9 REQUIREMENTS MODELING:SCENARIO-BASED METHODS （use case/activity/swimlane）

9.3.2 Swimlane Diagrams

The UML swimlane diagram is a useful variation of the activity diagram and allows you to represent the flow of activities described by the use case and at the same time indicate which actor (if there are multiple actors involved in a specific use case) or analysis class (Chapter 10) has responsibility for the action described by an activity rectangle.

Referring to Figure 9.6 , the activity diagram is rearranged so that activities associated with a particular analysis class fall inside the swimlane for that class. For example, the Interface class represents the user interface as seen by the homeowner. The activity diagram notes two prompts that are the responsibility of the interface—“prompt for reentry” and “prompt for another view.”

Figure 9.6 Swimlane diagram for Access camera surveillance via the Internet—display camera views function.