Chapter 11

Building Information Systems and Managing Projects

LEARNING TRACK 3: UNIFIED MODELING LANGUAGE (UML)

Object-oriented development can be used to improve system quality and flexibility. A number of techniques for the analysis and design of object-oriented systems have been developed, but the Unified Modeling Language (UML) has become the industry standard. UML allows system builders to represent different views of an object-oriented system using various types of graphical diagrams, and the underlying model integrates these views to promote consistency during analysis, design, and implementation.

Table 11-3 provides an overview of UML and its components. "Things" are objects and "structural things" allow system builders to describe objects and their relationships. UML uses two principal types of diagrams: structural diagrams and behavioral diagrams.

TABLE 11-3 An Overall View of UML and Its Components: Things, Relationships, and Diagrams

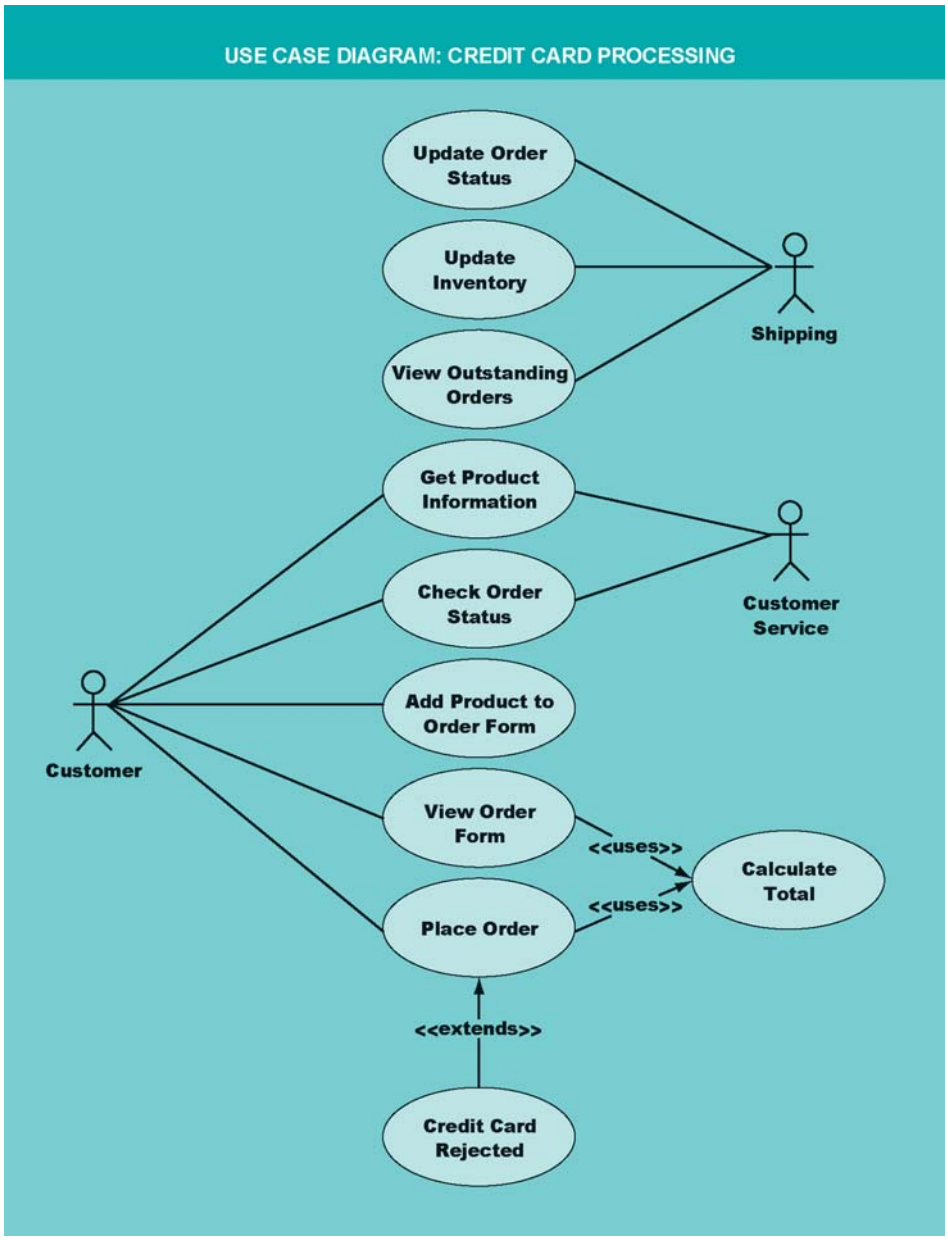
UML Category	UML Elements	Specific UML Details
Things	Structural Things	Classes Interfaces Collaborations Use Cases Active Classes Components Nodes
	Behavioral Things	Interactions State Machines
	Grouping Things	Packages
	Annotational Things	Notes
Relationships	Structural Relationships	Dependencies Aggregations Associations Generalizations
	Behavioral Relationships	Communicates Includes Extends Generalizes
Diagrams	Structural Diagrams	Class Diagrams Object Diagrams Component Diagrams Deployment Diagrams
	Behavioral Diagrams	Use Case Diagrams Sequence Diagrams Collaboration Diagrams Statechart Diagrams Activity Diagrams

Structural diagrams are used to describe the relationships between classes. One type of structural diagram is called a class diagram. It shows classes of employees and the relationships between them. The terminators at the end of the relationship lines in this type of diagram indicate the nature of the relationship. Generalization is a relationship between a general kind of thing and a more specific kind of thing. This type of relationship is sometimes described as a "is a relationship." Generalization relationships are used for modeling class inheritance.

Behavioral diagrams are used to describe interactions in an object-oriented system. Figures 11-5 and 11-6 illustrate two types of behavioral diagrams: a use case diagram and a sequence diagram. A use case diagram shows the relationship between an actor and a system. The actor (represented in the diagram by a stick figure) is an external entity that interacts with the system, and the use case represents a series of related actions ini-

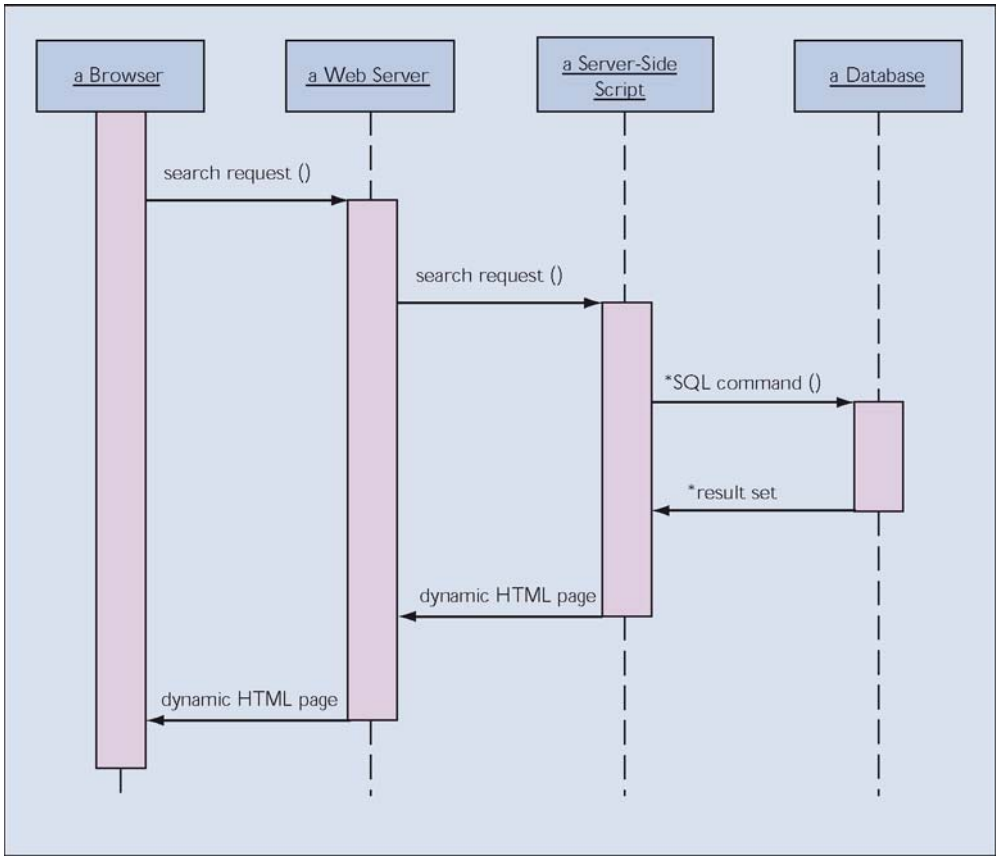
FIGURE 11-5 A UML use-case diagram.

Use case diagrams model the functions of a system, showing how objects interact with each other and with the users of the system. Illustrated here is a use case diagram for credit card processing that was created with SmartDraw software.



tiated by the actor to accomplish a specific goal. Several interrelated use cases are represented as ovals within a box. Use case modeling specifies the functional requirements of a system, focusing on what the system does rather than how it does it. The system's objects and their interactions with each other, and with the users of the system, are derived from the use case model. A sequence diagram describes the interactions among objects during a certain period of time. The vertical axis represents time, whereas the horizontal axis represents the participating objects and actors. Boxes along the top of the diagram represent actors and instances of objects. Lateral bars drop down from each box to the bottom of each diagram, with interactions between objects represented by arrows drawn from bar to bar. The sequence of events is displayed from top to bottom, with the first interaction at the top and the last at the bottom of the diagram. Sequence diagrams are used in system design to derive the interactions, relationships, and operations of the objects in the system.

FIGURE 11-6 A UML sequence diagram.



Sequence diagrams describe interactions among classes in terms of the communication between objects (messages) during a specified period of time. Illustrated here is a sequence diagram for a Web page interacting with a database that was created with SmartDraw software.