as object, but in UML they are called things. Structural things are most common. Structural things are classes, interfaces, use cases, and many other elements that provide a way to create models. Structural things allow the user to describe relationships. Behavioral things describe how things work. Examples of behavioral things are interactions and state machines. Group things are used to define boundaries. An example of a group thing is a package. Finally, we have annotational things, so that we can add notes to the diagrams.

Relationships are the glue that holds the things together. It is useful to think of relationships in two ways. Structural relationships are used to tie the things together in the structural diagrams. Structural relationships include dependencies, aggregations, associations, and generalizations. Structural relationships show inheritance, for example. Behavioral relationships are used in the behavioral diagrams. The four basic types of behavioral relationships are communicates, includes, extends, and generalizes.

There are two main types of diagrams in UML: structural diagrams and behavioral diagrams. Structural diagrams are used, for example, to describe the relationships between classes. They include class diagrams, object diagrams, component diagrams, and deployment diagrams. Behavioral diagrams, on the other hand, can be used to describe the interaction between people (called actors in UML) and the thing we refer to as a use case, or how the actors use the system. Behavioral diagrams include use case diagrams, sequence diagrams, communication diagrams, statechart diagrams, and activity diagrams.

In the remainder of this chapter, we first discuss use case modeling, the basis for all UML techniques. Next, we look at how a use case is used to derive activities, sequences, and classes—the most commonly used UML diagrams. Because entire books are dedicated to the syntax and usage of UML (the actual UML specification document is over 800 pages long), we provide only a brief summary of the most valuable and commonly used aspects of UML.

The six most commonly used UML diagrams are:

1. A use case diagram, describing how the system is used. Analysts start with a use case diagram.
2. A use case scenario (although technically it is not a diagram). This scenario is a verbal articulation of exceptions to the main behavior described by the primary use case.
3. An activity diagram, illustrating the overall flow of activities. Each use case may create one activity diagram.
4. Sequence diagrams, showing the sequence of activities and class relationships. Each use case may create one or more sequence diagrams. An alternative to a sequence diagram is a communication diagram, which contains the same information but emphasizes communication instead of timing.
5. Class diagrams, showing the classes and relationships. Sequence diagrams are used (along with CRC cards) to determine classes. An offshoot of a class diagram is a gen/spec diagram (which stands for generalization/specialization).
6. Statechart diagrams, showing the state transitions. Each class may create a statechart diagram, which is useful for determining class methods.

How these diagrams relate to one another is illustrated in Figure 10.5. We will discuss each of these diagrams in the following sections.

USE CASE MODELING

UML is fundamentally based on an object-oriented analysis technique known as use case modeling, which was introduced in Chapter 2. A use case model shows a view of the system from the user perspective, thus describing *what* a system does without describing *how* the system does it. UML can be used to analyze the use case model, and to derive system objects and their interactions with each other and with the users of the system. Using UML techniques, you further analyze the objects and their interactions to derive object behavior, attributes, and relationships.

A use case provides developers with a view of what the users want. It is free of technical or implementation details. We can think of a use case as a sequence of transactions in a system. The use case model is based on the interactions and relationships of individual use cases.

A use case always describes three things: an actor that initiates an event; the event that triggers a use case; and the use case that performs the actions triggered by the event. In a use case,

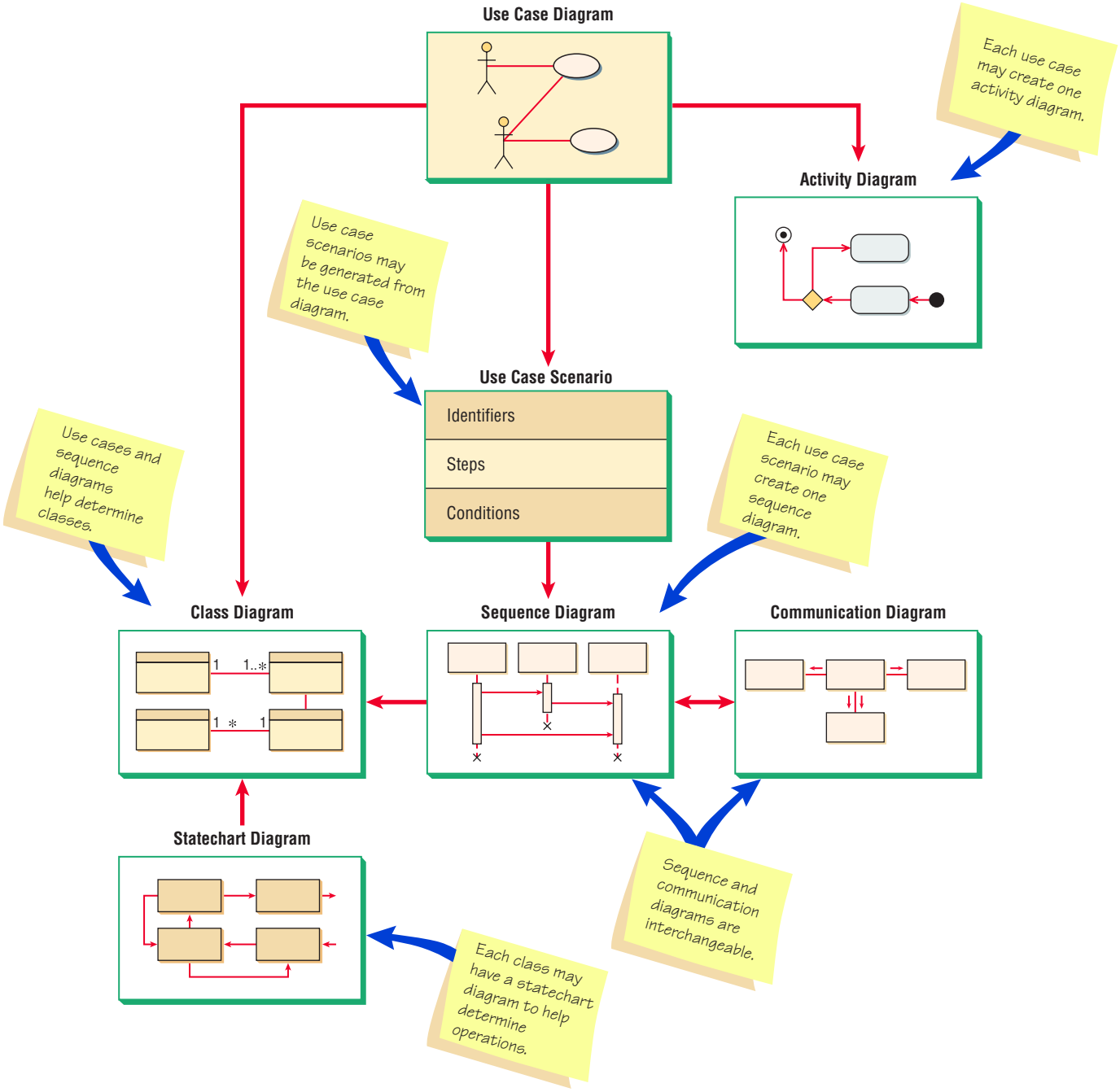

FIGURE 10.5

An overall view of UML diagrams showing how each diagram leads to the development of other UML diagrams.

an actor using the system initiates an event that begins a related series of interactions in the system. Use cases are used to document a single transaction or event. An event is an input to the system that happens at a specific time and place and causes the system to do something. For more information about use case symbols and how to draw use case diagrams, see Chapter 2.

Figure 10.6 is a use case example of student enrollment at a university. Notice that only the most important functions are represented. The **Add Student** use case does not indicate how to add students, the method of implementation. Students could be added in person, using the Web, using a touch-tone telephone, or any combination of these methods. The **Add Student** use case includes the **Verify Identity** use case to verify the identity of the student. The **Purchase Textbook** use case extends the **Enroll in Class** use case, and may be part of a system to enroll students in an online course.

It may seem as if the **Change Student Information** use case is a minor system feature and should not be included on the use case diagram, but because this information changes fre-

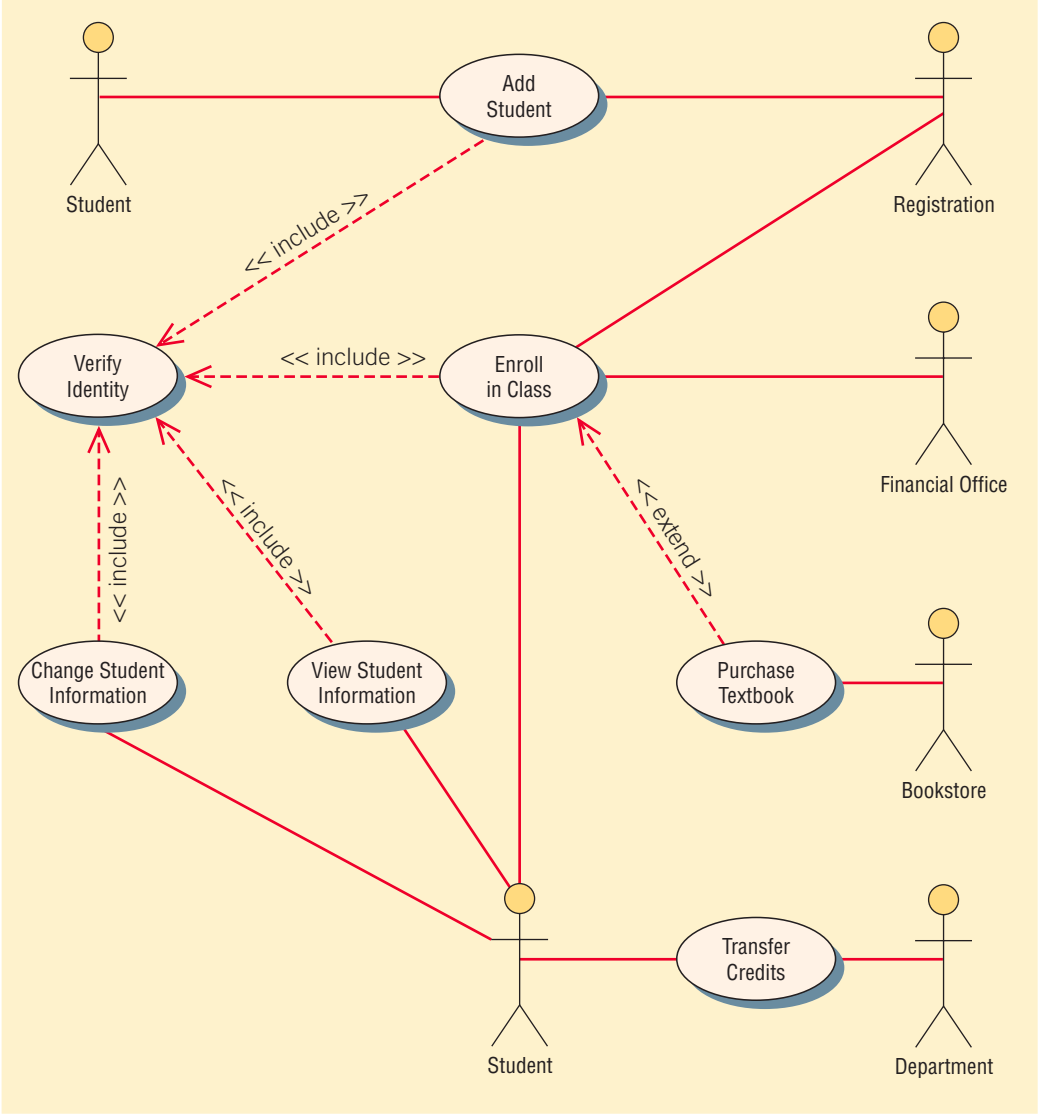

FIGURE 10.6

A use case example of student enrollment.

quently, administration has a keen interest in allowing students to change their own personal information. The fact that the administrators deem this to be important not only justifies, but calls for, the use case to be written up.

Students would not be allowed to change grade point average, outstanding fees, and other information. This use case also includes the **Verify Identity** use case, and in this situation, it means having the student enter a user ID and password before gaining access to the system. **View Student Information** allows students to view their personal information, as well as courses and grades.

A use case scenario example is shown in Figure 10.7. Some of the areas included are optional, and may not be used by all organizations. The three main areas are:

1. A header area containing case identifiers and initiators.
2. Steps performed.
3. A footer area containing preconditions, assumptions, questions, and other information.

In the first area the use case is identified by its name, **Change Student Information;** the actor is identified as a **Student;** and the Use Case and Triggering Event are described. The second area contains a series of steps that are performed as long as no errors are encountered. Finally, in the third area, all of the pre- and postconditions and assumptions are identified. Some of these are obvious, such as the precondition that the student is on the correct Web page and the assumption that the student has a valid student ID and password. Others are not so obvious, such as the outstanding issue regarding how many times the student is allowed to log on to the system.

Use case name:	Change Student Information	**UniqueID:** Student UC 005
Area:	Student System	
Actor(s):	Student	
Description:	Allow student to change his or her own information, such as name, home address, home telephone, campus address, campus telephone, cell phone, and other information using a secure Web site.	
Triggering Event:	Student uses Change Student Information Web site, enters student ID and password, and clicks the **Submit** button.	
Trigger type:	☒ External ☐ Temporal	
Steps Performed (Main Path)		**Information for Steps**
1. Student logs on to the secure Web server.		Student ID, Password
2. Student record is read and password is verified.		Student Record, Student ID, Password
3. Current student personal information is displayed on the Change Student Web page.		Student Record
4. Student enters changes on the Change Student Web form and clicks **Submit** button.		Change Student Web Form
5. Changes are validated on the Web server.		Change Student Web Form
6. Change Student Journal record is written.		Change Student Web Form
7. Student record is updated on the Student Master.		Change Student Web Form, Student Record
8. Confirmation Web page is sent to the student.		Confirmation Page
Preconditions:	Student is on the Change Student Information Web page.	
Postconditions:	Student has successfully changed personal information.	
Assumptions:	Student has a browser and a valid user ID and password.	
Requirements Met:	Allow students to be able to change personal information using a secure Web site.	
Outstanding Issues:	Should the number of times a student is allowed to logon be controlled?	
Priority:	Medium	
Risk:	Medium	

FIGURE 10.7

A use case scenario is divided into three sections: identification and initiation, steps performed, and conditions, assumptions, and questions.

Use case diagrams provide the basis for creating other types of diagrams, such as class diagrams and activity diagrams. Use case scenarios are helpful in drawing sequence diagrams. Both use case diagrams and use case scenarios are powerful tools to help us understand how a system works in general.

ACTIVITY DIAGRAMS

Activity diagrams show the sequence of activities in a process, including sequential and parallel activities, and decisions that are made. An activity diagram is usually created for one use case and may show the different possible scenarios.

The symbols on an activity diagram are illustrated in Figure 10.8. A rectangle with rounded ends represents an activity, either a manual one, such as signing a legal document, or an automated one, such as a method or program.