7.6 INTERACTION DIAGRAMS

When a user invokes one of the functions supported by a system, the required behaviour is realized through the interaction of several objects in the system. Interaction diagrams, as their name itself implies, are models that describe how groups of objects interact among themselves through message passing to realize some behaviour.

Typically, each interaction diagram realizes the behaviour of a single use case.

Sometimes, especially for complex use cases, more than one interaction diagrams may be necessary to capture the behaviour. An interaction diagram shows a number of example objects and the messages that are passed between the objects within the use case.

There are two kinds of interaction diagrams: sequence diagrams and collaboration diagrams. These two diagrams are equivalent in the sense that any one diagram can be derived automatically from the other. However, they are both useful. These two actually portray different perspectives of behaviour of a system and different types of inferences can be drawn from them. The interaction diagrams play a major role in any effective object-oriented design process. We discuss this issue in Chapter 8.

Sequence diagram

A sequence diagram shows interaction among objects as a two-dimensional chart. The chart is read from top to bottom. The objects participating in the interaction are shown at the top of the chart as boxes attached to a vertical dashed line. Inside the box the name of the object is written with a colon separating it from the name of the class and both the name of the object and the class are underlined. This signifies that we are referring any arbitrary instance of the class. For example, in Figure 7.29 :Book represents any arbitrary instance of the Book class.

An object appearing at the top of the sequence diagram signifies that the object existed even before the time the use case execution was initiated. However, if some object is created during the execution of the use case and participates in the interaction (e.g. a method call), then the object should be shown at the appropriate place on the diagram where it is created.

The vertical dashed line is called the object's lifeline. Any point on the lifeline implies that the object exists at that point. Absence of lifeline after some point indicates that the object ceases to exist after that point in time. particular point of time. Normally, at the point if an object is destroyed, the lifeline of the object is crossed at that point and the lifeline for the object is not drawn beyond that point. A rectangle called the activation symbol is drawn on the lifeline of an object to indicate the points of time at which the object is active. Thus, an activation symbol indicates that an object is active as long as the symbol (rectangle) exists on the lifeline. Each message is indicated as an arrow between the lifelines of two objects. The messages are shown in chronological order from the top to the bottom. That is, reading the diagram from the top to the bottom would show the sequence in which the messages occur.

Each message is labelled with the message name. Some control information can also be included. Two types of control information are particularly valuable.

- A condition (e.g. [invalid]) indicates that a message is sent, only if the condition is true.
- An iteration marker shows that the message is sent many times to multiple receiver

objects as would happen when you are iterating over a collection or the elements of an array. You can also indicate the basis of the iteration, e.g. [for every book object].

The sequence diagram for the book renewal use case for the Library Automation Software is shown in Figure 7.29. Observe that the exact objects which participate to realize the renewbook behaviour and the order in which they interact can be clearly inferred from the sequence diagram. The development of the sequence diagram in the development methodology (discussed in Chapter 8) would help us to determine the responsibilities that must be assigned to the different classes; i.e. what methods should be supported by each class.

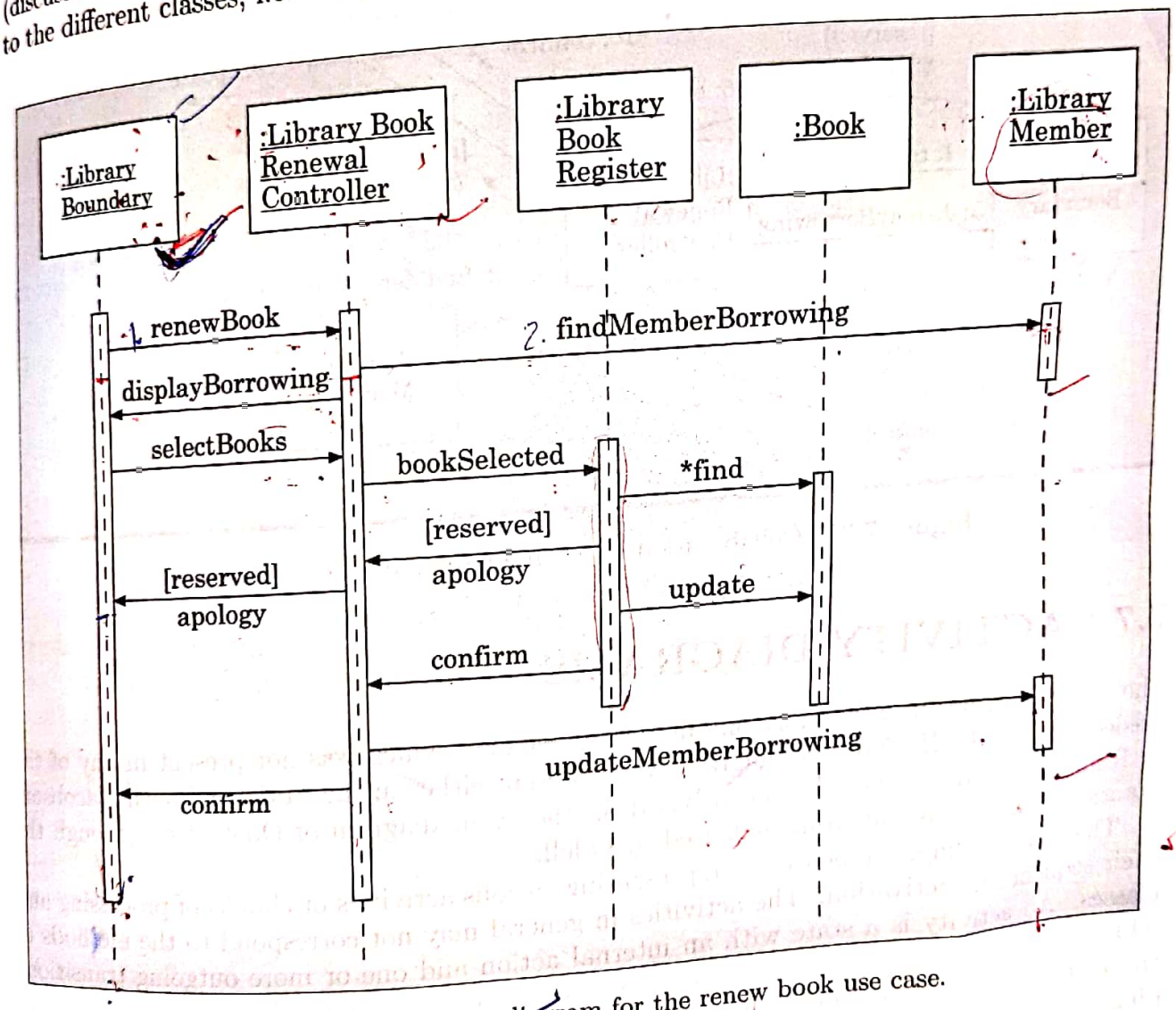

Figure 7.29: Sequence diagram for the renew book use case.

Collaboration diagram

A collaboration diagram shows both structural and behavioural aspects explicitly. This is unlike a sequence diagram which shows only the behavioural aspects. The structural aspect of a collaboration diagram consists of objects and links among them indicating association. In this diagram, each object is also called a collaborator. The behavioural aspect is described by the set of messages exchanged among the different collaborators.

The link between objects is shown as a solid line and can be used to send messages between two objects. The message is shown as a labelled arrow placed near the link. Messages

are prefixed with sequence numbers because they are the only way to describe the relative sequencing of the messages in this diagram.

The collaboration diagram for the example of Figure 7.29 is shown in Figure 7.30. An use of the collaboration diagrams in our development process would be to help us to determine which classes are associated with which other classes.

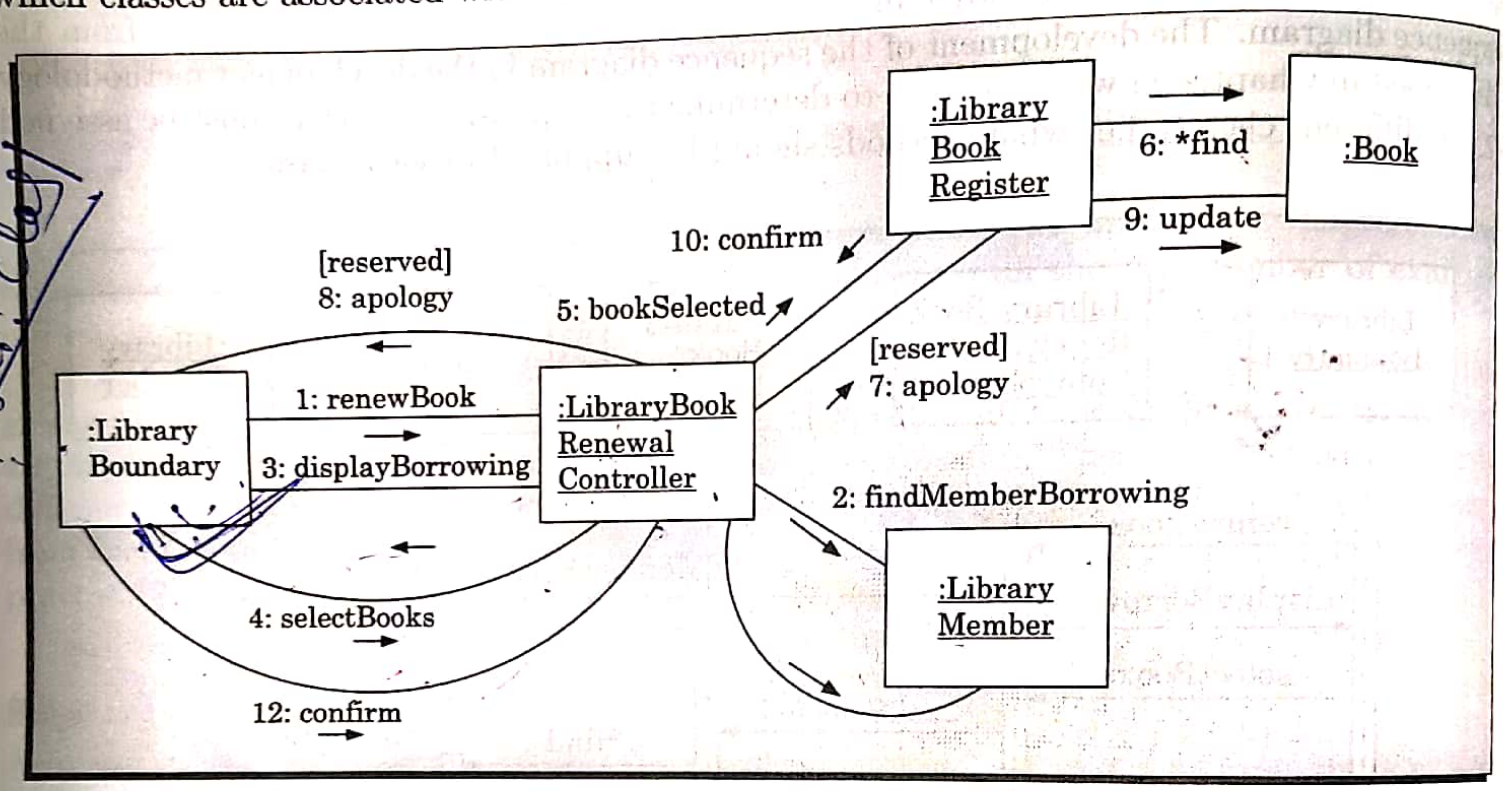

Figure 7.30: Collaboration diagram for the renew book use case.