numerical sequence. This is to make it clear which object is calling which.

- calcTotalPayment() will scoop up the total hire fees and total deposits paid and record them in totalAmountPaid and totalDepositPaid. How it does this is left to the designer.
- As it executes, the calcTotalPayment() operation calls issueReceipt() which is another operation on the Payment class; this is known as a *reflexive message*. The Payment object is already active, so this second activation is shown as a new activation box on top of the existing one.
- As the issueReceipt() operation is executing, it sends a message to :Customer to retrieve the customer name and address so that it can print them on the receipt.
- In Figure 6.12 all returns are shown so that you can clearly see the nesting of operations and how control returns eventually to the sending objects. In fact return arrows are usually omitted unless they add meaning to the diagram. Control is always returned to the sending object as soon as the receiving object ceases to be active. The activation boxes tell us how long an object is active, so we can assume the returns. Figure 6.13 shows us the same diagram without the return arrows.
- Notice that :Customer remains active while it sends a message to :Payment. It ceases to be active only when it returns control to the interface.

Collaboration diagrams

Collaboration diagrams show pretty much the same information as sequence diagrams. In fact most CASE tools will automatically generate a collaboration diagram from a sequence diagram or vice versa.

The collaboration diagram version of Figure 6.13 is shown in Figure 6.14.

Although this diagram contains much the same information as the diagram in Figure 6.13, it is obviously different in a number of ways.

- Messages are not shown in time sequence, so numbering becomes essential to indicate the order of the messages
- Links between objects are explicitly modelled, which is not the case with the sequence diagram
- Messages are grouped together on the object links
- Messages are shown as text labels with an arrow that points from the client (the object sending the message) to the server (the object providing the response)