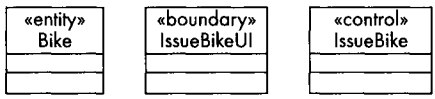

Figure 10.2 *Entity, boundary and control classes*

The same interaction is shown as a sequence diagram in Figure 10.7.

Entity, boundary and control classes are modelled as stereotypes. We have already come across stereotypes in Chapter 3 when discussing use case modelling: an actor is stereotyped class, «include» and «extend» are stereotyped associations. A stereotype is a specialized use of a modelling element which allows us to extend the basic UML set of modelling elements. Guillemets «» are used to indicate stereotypes, see Figure 10.2. Notice that it is not always useful to display the stereotype of a class on a diagram – we have managed quite well up to now without knowing that Bike, Customer, Hire, etc. were entity classes. As with many of the advanced modelling features, it is only worth showing the stereotype when it adds meaning to the diagram.

In reality, the Wheels system is so small that, when we implemented it, we combined the functions of a control and boundary class in a single class, see Figure 10.11. For the purposes of discussing the use of boundary and control classes, however, we have modelled them separately.

Designing associations

When we are designing, we need to make decisions about how the relationships between classes will be implemented. On early analysis class diagrams, associations between the classes Customer, Payment, Hire and Bike simply reflected the fact that in real life customers make payments and hire transactions, and hire transactions are for bikes, etc. Identification of class responsibilities and collaborators using CRC cards gave us more idea of what the associations should be. By the time we get to design, we know, from doing the interaction diagrams, exactly which objects need to communicate and the nature and direction of the messages. Associations between classes specify a requirement to implement navigable paths between objects of those classes. How we implement an association depends on the multiplicity of the association and whether communication is one way or two way.

One to one associations. There are no 1:1 associations in the Wheels system. Figure 10.3 shows a 1:1 association between a School class and a SchoolLibrary class. School objects need to be able to send