Figure 4.10 *Object diagram showing a customer hiring a bike*

Figure 4.10 is an object diagram showing a :Customer[3] linked to a :Bike. It models the real-life relationship of a customer hiring a bike. By contrast, in a class diagram (like the one in Figure 4.11) association relationships simply build into the model the possibility for objects to be linked; so that a :Customer can be linked to a :Bike as required. The association does not tell us which :Customers are actually linked to which :Bikes, as the object diagram in Figure 4.7 does, just that they can be linked. An association represents a group of links between objects in much the same way as a class represents a group of objects.

In the early stages of modelling, we don't yet know in detail how objects will need to communicate. At this stage, when we model an association between classes of objects, we are not saying much more than that a real-life connection exists between these objects, and can be used by the objects when needed.

Figure 4.11 shows the association relationship between the Customer and Bike classes.

We can see from Figure 4.11 that an association can be named: a customer *hires* a bike. An association has two ends, each attached to a class. Each association end can have a role name: on the association between Customer and Bike the association end next to Customer is *hirer*, next to Bike is *hired*. The default name for an association end is the name of the class it is attached to; for example, in Figure 4.11 the default name for the association end attached to Customer would be *customer*. In practice, both association names and role names are omitted, unless they significantly aid understanding.

Association ends also have multiplicity. Multiplicity is indicated by the numbers and asterisks on the line. The multiplicity of an association indicates limits on the number of objects allowed to participate in the

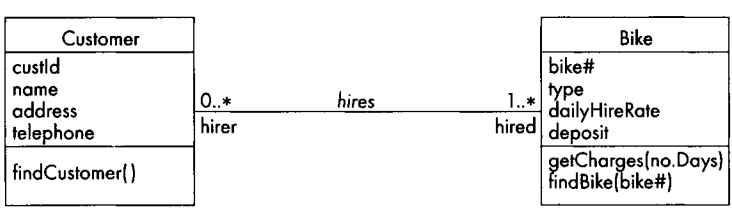

Figure 4.11 *Association between the Customer and Bike classes in the Wheels system*

3. *Remember – this notation means an object of the Customer class.*