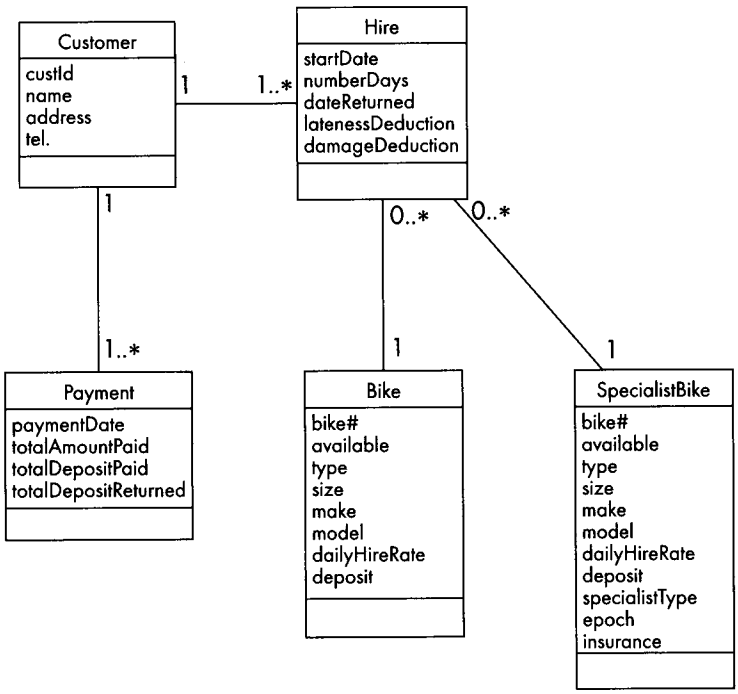


*Figure 5.13* *Wheels class diagram with initial associations*

attributes: bike#, available, type, size, make, model, dailyHireRate and deposit. If we make SpecialistBike a subclass of Bike, it will inherit all of these attributes and add its distinctive attributes: specialistType, epoch and insurance. Examination of the possible values of type (mountain bike, racer, tourer, etc.) and specialistType (penny farthing, tandem, unicycle, etc.) show that they are just different sets of values for the same attribute, so we can keep type and get rid of specialistType, see Figure 5.14.

If we incorporate this inheritance relationship into the Wheels model, we get the class diagram in Figure 5.15. Notice that SpecialistBike inherits not only Bike's attributes and operations, but also its relationships to other classes. SpecialistBike therefore has a one to many relationship with Hire.

### *Write a data dictionary to support the class diagram*

The UML does not provide a specific notation for constructing a data dictionary, but it is nonetheless important in any development project to have some agreed way of documenting the data and operations in the system. In the main UML models (such as the class, interaction and state diagrams) labels have to be short and detail about the data kept to a minimum, so that the model as a whole is uncluttered and readable. Detailed information about classes and their attributes is recorded in the data dictionary,