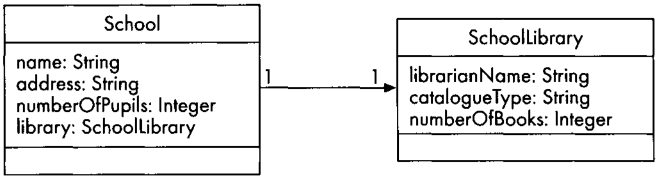

Figure 10.3 *Unidirectional one to one association*

messages to SchoolLibrary objects, but not vice-versa. The direction in which messages are sent is called the *navigability* of the association. One-way associations are referred to as *unidirectional.* The direction of the navigability is indicated by the arrowhead. Two-way navigability can be shown either by an arrowhead on both ends of the association, or by having no arrows. However, no arrowheads can also mean that the navigability has not been specified, as is the case on the analysis model. It is clearer to show the arrowheads.

To implement the one to one association, School has an attribute, library, which is used to hold the object identifier (otherwise referred to as a pointer or reference) of the SchoolLibrary object with which it needs to communicate. In a two-way association each class needs to hold a reference to the other. If the association between School and SchoolLibrary were two way, School would need to hold a reference to SchoolLibrary and SchoolLibrary would need to hold a reference to School.

One to many associations. Figure 10.4 shows a unidirectional, one to many association between the interface class MaintainBikeUI and the Bike class. A MaintainBikeUI object has to communicate with objects of the Bike class but not vice versa.

A MaintainBikeUI object has to be able to send messages to more than one :Bike, sometimes it will send a message to a specific :Bike identified by bike#, sometimes to a selection with the same make,

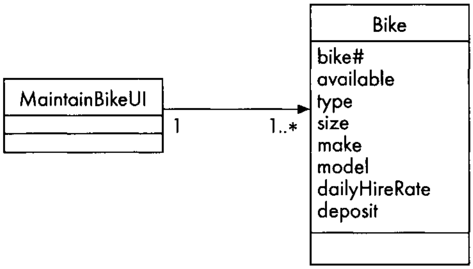

Figure 10.4 *Unidirectional one to many association*