

Table 4.1: UML association notation

Meaning	Example	Notation
an exact number	exactly one exactly six	1 (or may be omitted) 6
many	zero or more one or more, lots of	0..* 1..*, *
a specific range	one to four, zero to six	1..4, 0..6,
a choice	two or four or five	2, 4, 5

relationship. We must read multiplicity separately for each association end. To interpret the association between Customer and Bike in Figure 4.11 we first look at how many Bikes a Customer may hire, then at how many Customers a Bike may be hired by. To work out how many Bikes one Customer may hire we look along the association from the Customer class end to the number specified at the Bike class end. In this case the number is 1..*. This means one customer may hire 1 or many bikes. In object terms the diagram specifies that one Customer may be linked to 1 or many Bikes. Interpreting the multiplicity in the other direction, the diagram specifies that any one Bike may be hired by 0, 1 or many (0..*) Customers. We assume this means over a period of time. The multiplicity is always specified from the point of view of a single object.

The UML multiplicity notation allows us to specify degrees of association as shown in Table 4.1

Aggregation is often thought of as a tighter form of association; it models a whole-part relationship between classes, e.g. wheels, doors and an engine are parts of a car. An aggregation relationship can be identified:

- If a phrase such as ‘consists of’, ‘has a’, or ‘is a part of’ is used to describe the relationship
- If one class in the relationship (the whole) is more important than the other (the part)
- If some operations apply to the whole and its parts.

Figure 4.12 shows the aggregation relationship between a car and its parts: wheel, door and engine.

In the UML notation, aggregation is shown as a line joining the two classes with a diamond next to the whole class, see Figure 4.12. Notice that multiplicity can be specified at the part end of the