Introduction

In Chapter 3 we discussed how use case analysis can help identify and document, from the user's perspective, what the system has to do. In Chapters 4 and 5 we discussed how to identify objects and classes and their attributes using noun analysis and how to model relationships between classes. We made a list of the stages in the development of a class diagram and applied the first four stages to the Wheels system. The remaining three stages are: to identify class responsibilities using CRC cards, to separate responsibilities into operations and attributes, and to write process specifications to describe the operations.

In this chapter we discuss how to use the CRC technique to allocate responsibilities to classes and work out the interaction between classes that is required to implement the use case scenarios. We then discuss how to turn these high-level responsibilities into operations on classes and how to describe the functionality of the operations using process specifications. We introduce interaction diagrams and discuss how they are used to document the details of the interactions we identified using the CRC technique. Interaction diagrams also give us a much more precise idea of the associations between classes that are necessary to allow the requisite message passing between objects. In this chapter we also explain the difference between the two types of interaction diagram, sequence and collaboration, and discuss where to use each.

Identifying operations using the CRC card technique

As we saw in Chapter 5, objects can be identified from nouns in a description of the problem domain. In the same way it is possible to identify operations on classes by picking out verbs and verb phrases from the problem description. Phrases that occur in the requirements for the Wheels system (see Figure 5.2), such as 'keep a record of all customers' or 'work out automatically how much it will cost to hire a given bike', tell us that operations will be needed to fulfil these functions. However, analysing verb phrases in this way turns out to be an inefficient method of uncovering operations and allocating them to classes – a more effective and popular approach is to use CRC cards.

CRC (class-responsibility-collaboration) cards are not officially part of the UML, but are regarded as a valuable technique that works extremely well with it. CRC was popularized by a development method called Responsibility Driven Design. The method and the book that describes it (Wirfs-Brock *et al.*, 1990) are both quite old now, but the idea of thinking about a class in terms of the responsibilities it has to fulfil and the technique of CRC cards