classes in the problem domain. Features of the real-life objects that are of relevance to the system are modelled as attributes, and real-life links are modelled as relationships. During design activities we begin to think in terms of a software solution rather than just modelling the real-world situation. We add classes required by the logic of the solution that are not part of the real world, for example interface and control classes. The implementation model will add classes required to implement it, for example classes from the programming language's library such as mouse listeners, buttons and windows. However, the classes identified during analysis will still be in the code that implements the system.

The class diagram is the basis of the overall analysis model. Interaction diagrams (see Chapter 6) cannot be attempted until the class diagram has been constructed. Interaction diagrams identify the classes involved in producing the functionality of each use case. They specify, scenario by scenario, the messaging between objects that is required in the execution of the use cases. State diagrams (see Chapter 7) are also based on the class diagram. The state diagram models, class by class, all the things that can happen to an object and all its possible reactions. It focuses on one class of objects at a time and shows how all the scenarios affect it.

Common problems

1 Do the stages of building a class diagram have to be carried out in the same order as in the list?

As we said at the beginning of the chapter, there are many ways of building a class diagram. The list of stages is only a guideline, but it helps beginners to the subject to get started. Once you have constructed a number of class diagrams, your own experience will be the best guide as to how to proceed.

2 The Wheels class diagram does not contain any aggregation – why is that?

It is not compulsory to use aggregation, or any of the relationship constructs, in a class diagram. Aggregation is really only useful when there is an obvious 'consists-of' or 'is-made-up-of' relationship and it is important to document this. For example, in a restaurant system it might be useful to document that a set meal consists of three food courses and a drink. In the Wheels system there are no obvious aggregation relationships that it is important to record, so we have used association only.

3 How do I know what level of detail to go down to in a data dictionary?