with any diagram that is bigger than an A4 sheet. This is especially true if we are using a CASE tool (it's hard to follow a diagram that we can't see on one screen). To control complexity and manage our models, we partition the system into packages.

A *package* is a UML mechanism for grouping modelling elements. A package itself does not represent anything in the system, but is used to group modelling elements that do represent things in the system. In Chapter 3 we discussed grouping use cases into a package. Packages can also be used to group classes, collaborations, subsystems or a complete view of the system such as a use case model. We can also use packages to nest models; we might have a high-level class diagram that contains a package which itself contains a class diagram, and so on. In fact we can have high-level class diagrams that show only packages and dependencies – these are sometimes referred to as package diagrams, although this is not a UML term. For example, we might divide the Wheels system into two subsystems Hire Bike (which deals with issuing bikes and handling bike returns) and Manage Data (which maintains the Wheels lists of customers and bikes). We could represent each subsystem with a package. Each package would contain a class diagram. Figure 9.1 shows, on level 1, a package diagram representing the two subsystems; on level 2 the class diagram for the Manage Data subsystem. In reality, the Wheels system is so small that we would be unlikely to split it into subsystems.

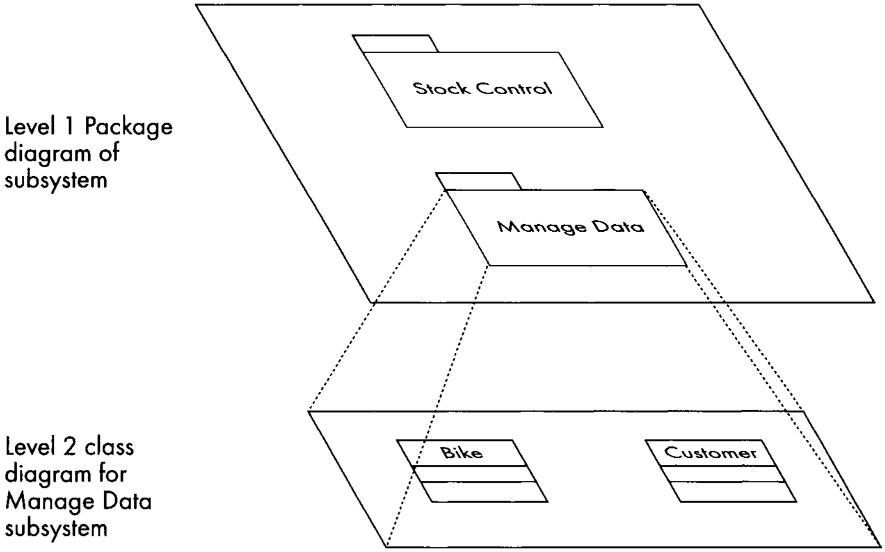
Level 1 Package diagram of subsystem
Stock Control
Manage Data
Level 2 class diagram for Manage Data subsystem
Bike
Customer

Figure 9.1 Nested models