use the system, the employees of the company. The modelling we describe in this book is aimed at the design of computer systems and the boundaries we draw are automation boundaries; our actors are therefore the people who use the system. In the Wheels system this would be the Receptionist and the Administrator.

The important thing to remember is that what we really want to know about, at the stage where we are designing the use case model, are the use cases themselves. In the early stages of software development the identification of actors is done principally as a means of finding out about the use cases. Later on, when considering security aspects and when designing the user interface, we might like to think some more about the actors involved so that we can specify user privileges and design interfaces that are appropriate for each user.

We find that it really is not worth losing sleep over who the actors are – it's a bit of a red herring. The actors we show in this book are the ones who physically use the system, e.g. put their hands on the keyboard, read the screen or receive a report. This is because, when we move on to doing the interaction diagrams, these actors are the only ones providing input directly to the system and reading the output from the screen. Our preferred use case model for the Wheels system, therefore, is that shown in Figures 3.2 and 3.9, not that shown in Figure 3.14.

Essential and real use cases. It is worth understanding the difference between essential and real use cases (again without getting lost in the jargon). An essential use case is one that is completely free of implementation or detailed design decisions. A use case will be in essential form in the early stages of software development before these decisions have been made. It is important to keep use cases free of implementation detail in the early stages so as not to constrain subsequent design decisions. Real use cases, by contrast, do show detail of design and implementation decisions insofar as they affect the user. A real use case will show, for example, detail about the user interface. Implementation decisions that do not directly affect the user's view of the system such as choice of programming language, data storage structures or programming algorithms are not specified in a use case.

The use case descriptions in Figures 3.5 and 3.6 are free of implementation information and are therefore essential use case descriptions. The use case description in Figure 3.15 has some implementation decisions added (e.g. that the bike details will be displayed in report format) and is therefore a real use case description.