Coming to grips with the required functionality by studying typical user interactions with the system is a technique that has been used informally by software developers since long before the arrival of object-orientation. Use cases are not essentially object-oriented; they are a useful tool whatever method of developing software is being used. Use case modelling formalizes and documents a long-standing practice. Jacobson (1992) was responsible for raising the profile of the use case to the extent that it is now almost universally adopted by the object-oriented community, is formally included in the UML notation and is part of almost all object-oriented methods of developing software.

It would be an inefficient use of time for a developer to write scenarios for every possible sequence of events in a use case. A software tester writes test cases to test an intelligent selection of cases; he tests to agreed coverage criteria. In much the same way a developer should write a representative set of scenarios. The scenarios should document:

- A typical sequence of events leading to the achievement of the use case goal – e.g. a customer hires one bike
- Obvious variations on the norm – e.g. a customer hires several bikes for a fixed period; a customer hires several bikes for different periods; a customer hires a specialist bike etc.
- Sequences of events where the use case goal is not achieved – e.g. the customer cannot find a bike he likes; the customer thinks the cost is too great, etc.

The developer needs to be sure he understands and documents how the system should respond in every eventuality. He will probably write detailed scenarios to document large or complicated use cases, but small, simple use cases can be adequately described by use case descriptions (see below).

To summarize, the purpose of the use case model is to describe, from the user's perspective, what the system does. A use case describes a cohesive piece of the system's functionality as the user perceives it. The user may see it as a task that he uses the system to achieve, one of the jobs that make up his daily workload, or it may produce a list or a report that he gets from the computer. A use case is a complete end-to-end use of the computer, a complete path through the system. A use case must deliver some benefit to the actor associated with it – it must have a goal. Each use case will have several scenarios associated with it. Some will be successful, i.e. achieve the use case goal, some will not. The software developer needs to be aware of all possible scenarios because the system must be able to cope with them all and respond appropriately. However,