

and for clarifying the developer's understanding of the users' requirements. Actors identified in the use case diagram represent users who interact with the system in some way, who use the system to achieve a particular task. Each use case represents a task or major chunk of functionality. The use case description describes the task in more detail. The description views the task from the user's perspective, it does not attempt to provide a program specification.

There are several different ways of describing alternative paths through a use case. All routes through the use case – what normally happens, minor variations and significant alternatives – can be illustrated in detail using scenarios. Minor variations can be documented as alternative courses in the use case description. More significant differences can be modelled as separate use cases linked to the original by an «extend» relationship. Alternatively, a specialized use case can be used to model different behaviour within a use case. Common functionality can be extracted into a separate use case linked by an «include» relationship to any use case that needs this functionality.

The use case model is a useful starting point for identifying classes and forms the basis of the interaction diagrams. It is also useful as a guide for project management, for checking requirements and for system testing.

Bibliography

Bennett, S., McRobb, S. and Farmer, R. (2002) *Object-Oriented Systems Analysis and Design Using UML* (2nd edition), McGraw-Hill, London.

Brown, D. (1997) *Object-Oriented Analysis: objects in plain English*, John Wiley, NY.

Fowler, M. (2000) *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (2nd edition), Addison-Wesley, Reading, MA.

Jacobson, I. (1992) *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, Wokingham, England.

Larman, C. (1998) *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*, Prentice Hall, NJ.

Stevens, P., with Pooley, R. (2000) *Using UML. Software Engineering with Objects and Components* (updated edition), Addison-Wesley, Harlow.