

This depends on how the data dictionary will be used. If it is to be used simply to document the system developer's understanding of the current system and to support discussions with the client, labels such as 'name' or 'date' will be self-explanatory; no more detail is required. During design, however, the developer is thinking ahead to the detail required in an automated system. Decisions have to be made that were unnecessary earlier, for example, about input and output formats for dates and how they are going to be represented internally.

Chapter summary

The class diagram defines the software architecture and the internal structure of the objects in an object-oriented system; the classes we model in the class diagram form the basis of the classes in the code. In this chapter we limit the discussion to the analysis class diagram; this only models classes which have meaning in the application domain. Classes relating to the design and implementation of the solution will be discussed in Chapters 9 and 10. The association relationships we identify in the analysis diagram model real-life relationships between objects and evolve through various iterations of the model to define the navigable paths between the objects. The inheritance relationships we identify clarify and simplify the model and allow us to build in some future-proofing of our system.

The stages in the construction of a class diagram are identifying objects and deriving classes, identifying attributes, identifying relationships, writing a data dictionary, identifying operations and writing operation specifications. In this chapter we cover the first four of these stages, relating to classes, attributes, relationships and the data dictionary. We also show how managing the model in larger systems can be facilitated by using a package to split classes into logical groups.

Bibliography

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

Britton, C. and Doake, J. (2000) *Object-Oriented Systems Development: A Gentle Introduction*, McGraw-Hill, London.

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