G

UML: Additional Diagram Types

G.1 Introduction

If you've read the optional Software Engineering Case Study in Chapters 25–26, you should now have a comfortable grasp of the UML diagram types that we use to model our ATM system. The case study is intended for use in first- or second-semester courses, so we limit our discussion to a concise subset of the UML. The UML 2 provides a total of 13 diagram types. The end of Section 25.3 summarizes the six diagram types that we use in the case study. This appendix lists and briefly defines seven other diagram types.

G.2 Additional Diagram Types

The following are the seven diagram types that we've chosen not to use in our Software Engineering Case Study.

- **Object diagrams** model a "snapshot" of the system by modeling a system's objects and their relationships at a specific point in time. Each object represents an object of a class from a class diagram, and several objects may be created from one class. For our ATM system, an object diagram could show several distinct `Account` objects side by side, illustrating that they are all part of the bank's account database.
- **Component diagrams** model the **artifacts** and **components**—resources (which include source files)—that make up the system.
- **Deployment diagrams** model the system's runtime requirements (such as the computer or computers on which the system will reside), memory requirements, or other devices the system requires during execution.
- **Package diagrams** model the hierarchical structure of **packages** (which are groups of classes) in the system at compile time and the relationships that exist between the packages.
- **Composite structure diagrams** model the internal structure of a complex object at runtime. New in UML 2, they allow system designers to hierarchically decompose a complex object into smaller parts. Composite structure diagrams are beyond the scope of our case study. They are more appropriate for larger industrial applications, which exhibit complex groupings of objects at execution time.

- **Interaction overview diagrams**, new in UML 2, provide a summary of control flow in the system by combining elements of several types of behavioral diagrams (e.g., activity diagrams, sequence diagrams).
- **Timing diagrams**, also new in UML 2, model the timing constraints imposed on stage changes and interactions between objects in a system.

To learn more about these diagrams and advanced UML topics, please visit our UML Resource Center at `www.deitel.com/UML/`.