L

UML 2: Additional Diagram Types

L.1 Introduction

If you read the optional "Software Engineering Case Study" sections in Chapters 2—8*** and 10, you should now have a comfortable grasp on 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 2.9 summarizes the six diagram types that we use in the case study. This appendix lists and briefly defines the seven remaining diagram types.

L.2 Additional Diagram Types

The following are the seven diagram types that we have 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 instance of a class from a class diagram, and there may be several objects 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 runtime requirements of the system (such as the computer or computers on which the system will reside), memory requirements for the system, 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. Composite structure diagrams are new in UML 2 and allow system designers to hierarchically decompose a complex object into smaller parts. Composite structure diagrams are beyond the scope of our case study. Composite structure diagrams are more appropriate for larger industrial applications, which exhibit complex groupings of objects at execution time.
- **Interaction overview diagrams**, which are 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.

If you are interested in learning more about these diagrams and advanced UML topics, please visit `www.uml.org` and the Web resources listed at the ends of Section 1.16 and Section 2.9.